

Northumbria Research Link

Citation: Holt, Giles (2018) Characterising the impact on bacterial physiology of phage infection and phage as a tool to support microbiota studies. Doctoral thesis, Northumbria University.

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/39781/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

**Characterising the impact on bacterial physiology of
phage infection and phage as a tool to support
microbiota studies**

Giles Samuel Holt

PhD

2018

Characterising the impact on bacterial physiology of phage infection and phage as a tool to support microbiota studies

Giles Samuel Holt

BSc (Hons)

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of Doctor of Philosophy.

Research undertaken in the Faculty of Health and Life Sciences and in collaboration with Freeman hospital in Newcastle upon Tyne

March 2018

Abstract

Shiga-toxigenic encoding *Escherichia coli* are a global health concern. Carriage of the shigatoxin gene increases the pathogenicity of the bacteria as the toxin has downstream impact on clinical disease. Enterohaemorrhagic *E. coli* (EHEC) symptoms lead from mild to severe bloody diarrhoea, where the toxin targets protein synthesis in specific cells preceding cell death and clinical sequelae including; haemolytic ureamic syndrome (HUS), haemorrhagic colitis (HC) and thrombotic thrombocytopenic purpura (TTP). This toxin is carried by temperate lambdoid-like bacteriophages. How temperate bacteriophages play a role in microbial infection and disease progression is poorly understood but less is known of the extent in which they impact on microbial communities. Their role in bacterial adaptation and evolution is essential as they carry genes that promote positive evolutionary selection for the lysogen.

This study focuses on 4 key areas. The first compares previously studied Biolog phenotype microarrays to comparative metabolite profiling to study the impact of Shiga toxin-prophage $\phi 24_B$ on its *Escherichia coli* host MC1061. As a lysogen, this study determines that the prophage alters the bacterial physiology by increasing the rates of respiration and cell proliferation. This is the first reported study detailing phage-mediated control of the *E. coli* biotin and fatty acid synthesis that is rate limiting to cell growth. Through $\phi 24_B$ conversion the lysogen also gains increased antimicrobial tolerance to chloroxylenol and 8-hydroxyquinoline and other antimicrobials. When comparing the metabolic profiles between MC1061 and the $\phi 24_B$ lysogen in standard culture, and when treated with 2 antimicrobials, discrete differences are observed. This is also the first reported use of metabolite profiling to characterise the physiological impact of lysogeny under antimicrobial pressure. The study demonstrates that $\phi 24_B$ does not need to carry any distinguishable antimicrobial resistance genes to confer tolerance to antimicrobials.

The second focus further studies $\phi 24_B$ conversion of *E. coli* MC1061. It demonstrates that during growth increased resistance by the lysogen is acquired over time when challenged with increasing concentrations of 8-hydroxyquinoline and chloroxylenol. Using targeted GC-MS of the cell wall fatty acid structure the study shows that under optimal conditions the prophage alters the physiology of its host cell by decreasing the total fatty acid composition. Due to the biotin pathway

being intrinsically linked to the fatty acid synthesis pathway, it is probable that a similar mechanism is employed by $\phi 24_B$. Intriguingly distinct strategies in host cell wall fatty acids are noticed when treated with antimicrobials 8-hydroxyquinoline or chloroxylenol. When under challenge with either antimicrobial there is an increase in the total cell wall fatty acids in the lysogen, with significant increases observed in the presence of 8-hydroxyquinoline. From this study it can be hypothesised that when the host is not challenged by the antimicrobial, the phage manipulates the fatty acid synthesis pathway to redirect energy and resources from cell wall physiology. Further hypothesis can be made that under initial antimicrobial challenge, phage infection promotes broad antimicrobial tolerance by increasing total cell wall lipids, significantly increasing fatty acids that alter membrane fluidity. The observed tolerance increases exponentially over 24 hours compared to the naïve host, where the phage acquires true resistance by directing an alternative resistance mechanism to that of the cell wall fatty acid composition.

The third area focuses on design of a metabolomic program (CCRACD) that was completed to aid analysis downstream of discovery analysis software. Analysis of metabolite features over several conditions and file outputs is often difficult especially if chromatographic alignment offers error. Often steps in analysis between compound identification and plotting require laborious manual data mining to create profiles for compounds of interest that extend over several conditions. Metabolic profiling was achieved through construction of several bourne/bourne again (sh/bash) scripts and plots of the tabulated data were carried out using R scripts. The scripts were wrapped in a gui to make a user-friendly tool.

The fourth chapter illustrates the design of a genomic program (GGOSS) built to aid study in chapter 7. Genomic analysis using open source software (OSS) in a linux operating system has become standard practice for many genomic studies. However the increasing need to run analysis on a larger scale, to demonstrate the use of multiple tools for each step, and the amount of steps still required to be done by hand/eye, has been a challenge for researchers without a programming background. Over the years many programs have been built to broach this, which have high cost and/or strict pathways/tools and/or are web based (server/connection dependent). Recent initiatives like the MRC funded CLIMB have been developed to overcome this, but again this is driven at the

command line. This research informed construction and development of a bioinformatics tool that provides a free, installable GUI, with simple OSS installation, intuitive use, simple drag and drop for mass files, saveable tool setting menus and pipelines, and comparative OSS tools. GGOSS enables scientists to run genomic analysis without the need for prior computing skills, but with a working knowledge of the analysis to complete.

Finally, this study investigates the developing gut microbiota of very preterm infants. Necrotising enterocolitis (NEC) and late onset infection remain major causes of morbidity and mortality in those born preterm. The bacterial microbiota of preterm neonates has been widely studied: dysbiosis appears key to disease development, yet how this occurs is poorly understood. This study compares gut bacteria and bacteriophages over an 8 week period in 2 twin pair sets, plus one individual child. All children are extremely premature and were residing on the neonatal intensive care unit (NICU), Royal Victoria Infirmary, Newcastle upon Tyne. Massively parallel DNA sequencing was used to profile the bacterial, fungal, free virus and chemically induced lysogenic viruses from preterm infant stool to overlay analyses. Community structure and viral metagenomics were compared against clinical data to assess the impact of combining all techniques. Greater resolution in microbial dissimilarity between individual infants and twin pairs was observed with the inclusion of lysogenic bacteriophages, and even more so with free virus data. Lysogenic communities showed strongest similarities to the bacterial communities, reflecting bacterial viability. Bacterial taxonomic richness increased over time in all patients. Decrease in viral richness was seen in infants who remained healthy. This study demonstrates the potential importance of complementary viral community analysis in evaluating the role of microbiota stability and dysbiosis in disease states.

Table of Contents

Chapter 1.	Introduction.....	1
1.1	<i>Escherichia coli</i> , and shigatoxin (stx) encoding <i>Escherichia coli</i>	1
1.1.1	<i>Escherichia coli</i>	1
1.1.2	Stx toxin.....	1
1.1.3	Stx toxin compositions and binding properties.....	2
1.1.4	Shigatoxin encoding <i>Escherichia coli</i> (STEC).....	4
1.1.5	Shiga toxin encoding bacteriophages (Stx-phages).....	6
1.2	Types of viruses.....	7
1.2.1	DNA viruses.....	7
1.2.2	RNA viruses and reverse transcribing viruses.....	9
1.3	Phage life cycles.....	10
1.3.1	Adsorption and Infection.....	10
1.4	Bacteriophage structure and assembly.....	20
1.4.1	Capsid and assembly.....	21
1.4.2	DNA packaging: Mature procapsid to Nucleocapsid.....	22
1.4.3	Tail fibres.....	23
1.5	Phage/host co-evolution.....	23
1.5.1	Phage encoded protection.....	24
1.5.2	Toxin/anti-toxin mechanisms.....	27
1.5.3	Clustered regularly interspaced short palindromic repeats (CRISPR)/CRISPR-Associated-protein (CAS).....	31
1.6	Human gut microbiota.....	33
1.6.1	The gut flora and metabolic function.....	37

1.6.2	The gut flora and immune system.....	38
1.6.3	The gut flora and disease markers, progression and prevention	41
1.6.4	Gut microbiota and antibiotic treatment	45
1.6.5	Gut dysbiosis and therapy	45
1.6.6	The gut virome	47
1.7	Aims of the project.....	51
Chapter 2.	General Materials and Methods	52
2.1	Materials and growth media constituents.....	52
2.1.1	Sterilisation	52
2.1.2	Broth and Agar.....	52
2.1.3	Buffers and inducers	52
2.2	Bacterial and viral strains.....	53
2.3	Growth and Maintenance.....	53
2.3.1	Induction - Checking successfully integrated phage.....	53
2.3.2	Stocks, overnights and sub-cultures.....	54
2.4	Bacterial phenotypic microarray	54
2.5	Sub-inhibitory concentration (SIC) assay, 96 well plate assay set up	55
2.5.1	SIC assay.....	56
2.6	SIC LCMS analysis comparing metabolic compounds from Naïve host and Lysogens...56	
2.7	SIC assay bacterial strains and growth conditions - Buffer and Agar	57
2.8	Growth curve of single and double lysogens	57
2.9	Bacterial phenotypic microarray - Biolog.....	58
2.10	SIC assay: LCMS and MSMS preparation and run.	58
2.11	Biotin quantification assay.....	59

2.12	Cell harvesting of MC1061 and $\phi 24_B$ grown under sub-inhibitory antimicrobial challenge	59
2.13	Growth and cell harvesting of MC1061 and $\phi 24_B$ under increasing antimicrobial challenge	60
2.13.1	Preliminary growth curves	60
2.13.2	Growth and cell harvest	60
2.14	Cell wall fatty acid methyl ester isolation	61
2.15	Gas chromatographic analysis of fatty acid methyl esters	61
2.16	Stool sample collection, storage, preparation, and DNA extraction:	62
2.16.1	Patient cohort	62
2.16.2	Stool sample collection, storage,	62
2.16.3	Free viral particle (FVP) isolation	63
2.16.4	Induced viral particle isolation	63
2.16.5	Viral DNA extraction	63
2.16.6	Total DNA isolation for microbial community analysis	63
2.17	Genome sequencing	64
2.17.1	Viral metagenomic sequencing	64
2.17.2	Bacterial community amplicon sequencing analysis	64
2.17.3	Fungal community amplicon sequencing analysis	64
2.17.4	Data clean-up and read count/distribution	64
2.17.5	Data plotting	65
2.18	Statistical analysis	66
2.18.1	Determining p values	66
2.18.2	GC-MS Fatty acid methyl ester identification, data gathering and plotting	67

Chapter 3. Shigatoxin encoding Bacteriophage ϕ 24B modulates bacterial metabolism to raise antimicrobial tolerance.....	68
3.1 Introduction.....	68
3.1.1 STEC and Stx-phages	68
3.1.2 Phage-bacteria co-evolution.....	68
3.1.3 Aim	71
3.2 Results.....	72
3.2.1 Effect of ϕ 24 _B integration on naïve host growth and respiration	72
3.2.2 ϕ 24 _B integration effects naïve host metabolome	79
3.3 Discussion.....	88
Chapter 4. STX-phage ϕ 24 _B hi-jacks microbial cell wall fatty acid synthesis and increases the rate of acquired resistance.....	95
4.1 Introduction.....	95
4.1.1 Antimicrobial resistance	97
4.1.2 Roles of phage-host interactions in antimicrobial resistance.....	98
4.1.3 Aims.....	99
4.2 Results.....	100
4.2.1 Environmental impact on lysogen and naïve host growth	100
4.2.2 Cell wall fatty acid profiling of lysogen vs uninfected bacterium	106
4.3 Discussion.....	129
4.3.1 Influence of environmental factors on growth.....	129
4.3.2 Subversion of the Biotin pathway.....	131
4.3.3 Prophage carriage on fatty acids and cell wall fatty acids.	131
4.3.4 Conclusion	144

Chapter 5.	Comparative Metabolomics using: Cross Run Analysis for Comparable Compound Data profiling (CRACCD) and GUI interface	145
5.1	Introduction.....	145
5.1.1	Background of Metabolomics as a technique	145
5.2	Metabolomics and liquid chromatography mass spectrometry (LC-MS).....	146
5.2.1	LC-MS	146
5.2.2	LC-MS/MS	148
5.2.3	Data creation, quality and issues.....	148
5.2.4	Bioinformatics.....	150
5.2.5	Aim	151
5.3	Results.....	152
5.3.1	Languages and tools used	153
5.3.2	Mapping the program interface and data sorting	153
5.3.3	Example data run through, demonstrating program settings, calculations and output	
	161	
5.4	Discussion.....	178
Chapter 6.	Graphical User Interface (GUI) for Genomic analysis using Open Source Software (GGOSS)	180
6.1	Introduction.....	180
6.1.1	Genetic repositories and databases.....	180
6.1.2	Genomic bioinformatics.....	183
6.1.3	Difficulties in genomic analysis.....	184
6.1.4	Programming languages.....	185
6.1.5	Aim	185
6.2	Results.....	186

6.2.1	GGOSS installation.....	186
6.2.2	GUI map and file importation.....	187
6.2.3	Setting menus.....	189
6.2.4	Unique features to GGOSS.....	210
6.3	Discussion.....	217
6.3.1	OSS genomic tools.....	217
6.3.2	Future additions to GGOSS.....	227
Chapter 7.	Metagenomics approaches of bacterial and viral fraction of stool samples from low birth weight, preterm neonates.....	229
7.1	Introduction.....	229
7.1.1	Next generation sequencing technologies.....	230
7.1.2	Genome sequencing analysis.....	233
7.1.3	Viral metagenomics or viromics.....	236
7.1.4	Aim.....	240
7.2	Results.....	241
7.2.1	Fungal community analysis.....	241
7.2.2	Effect of frozen storage conditions on viral communities.....	241
7.2.3	Bacterial community analysis.....	242
7.2.4	Overlaying viral communities onto bacterial community analysis.....	243
7.2.5	Common comparisons between bacterial and viral communities between patients.....	249
7.3	Discussion.....	251
7.3.1	Conclusion.....	255
Chapter 8.	General Discussion.....	256
8.1	Discussion.....	256
8.2	Further work.....	264

Chapter 9.	References.....	266
Chapter 10.	Appendices.....	342
10.1	Appendix 1.....	342
10.1.1	Phage receptors localised in capsular and slime polysaccharides, pili and flagella	342
10.1.2	Cancer	344
10.1.3	Eating disorders and obesity	348
10.1.4	Prebiotics and Probiotics.....	350
10.2	Appendix 2.....	354
10.3	Appendix 3.....	364
10.3.1	Log10 normalisation	365
10.3.2	Reasoning for PLS-DA modelling.....	368
10.4	Appendix 4.....	369
10.5	Appendix 5.....	376
10.6	Appendix 6.....	377
10.6.1	dsDNA viruses, replication (Baltimore classification: Type I).....	377
10.6.2	ssDNA viruses, replication (Baltimore classification: type II)	378
10.6.3	dsRNA viruses (Baltimore classification: Type III).....	379
10.6.4	ssRNA viruses (Baltimore classification: Type IV and V).....	380
10.7	Appendix 7.....	384
10.8	Appendix 8.....	385
10.8.1	CRACCD installation GUI and script.....	385
10.8.2	GUI for metabolomics program CRACCD.....	390
10.8.3	Scripts	434
10.9	Appendix 9.....	513

10.9.1	GGOSS installation.....	513
10.9.2	GUI for genome sequencing program GGOSS.....	528
10.9.3	Scripts	618

Table of Figures

Figure 1.1	Stx retro-translocation and intracellular trafficking,.....	2
Figure 1.2	Displays structure and binding/cleavage sites of Stx.....	4
Figure 1.3	STEC major outbreaks time lined since 1982.....	5
Figure 1.4	The phage lytic and lysogenic cycle.	14
Figure 1.5	Map of the bacteriophage lambda chromosome that decides and carries out the lytic/lysogen decision.....	14
Figure 1.6	Regulation of PR and PL by CI repressor.....	15
Figure 1.7	Basic core genes driving lytic infection of phage lambda	17
Figure 1.8	Basic core genes driving lytic infection of phage lambda.	19
Figure 1.9	Lambda virion assembly.	20
Figure 1.10	Integrase region of Φ 24B, its Map, and its transcription	26
Figure 1.11	Types of TA systems. (A) Type I system regulated by interference of toxin mRNA translation, example; symR/symE module of E. coli.....	30
Figure 1.12	CRISP/Cas mechanism of action.	32
Figure 1.13	Development of the Microbiota	34
Figure 1.15	The taxonomic diversity and bacterial load through the gastrointestinal tract.....	36
Figure 1.16	Effect of Interactions of Bacteria, Viruses, and Eukaryotes in Health and Disease .42	
Figure 1.17	Potential strategies for phage therapy.	47
Figure 1.18	Potential consequences of a temperate phage lifecycle in the human gut.....	51
Figure 3.1	Clustered column graph representing percentage increase in cell proliferation of single (ϕ 24B:: Δ Kan, dark grey) and double (ϕ 24B:: Δ Kan, ϕ 24B:: Δ Cat, light grey) MC1061 lysogens. 73	
Figure 3.2	Respiration traces from raw Biolog data comparing naïve MC1061 respiration (light grey line) to lysogen (dark grey line), the hashed line represents (n=3) rates of respiration of both naïve MC1061 under standard growth conditions in the absence of challenge.	74
Figure 3.3	Respiration traces from raw Biolog data comparing naïve MC1061 respiration (light grey line) to lysogen (dark grey line), the hashed line represents (n=6) the combine respiration control data of both naïve MC1061 and Lysogen.....	75

Figure 3.4	A comparison of Area Under the Respiration Curve (AURC) data from the Biolog bacterial phenotypic microarray.....	76
Figure 3.5	Response in growth of both MC1061 (light grey) and the ϕ 24B lysogen (Dark grey) to an increasing concentration of (A) 8-hydroxyquinoline, (B) chloroxylenol, and (C) oxolinic acid	78
Figure 3.6	A, B and C: The metabolite profiles of MC1061 versus lysogen and multivariate analysis using partial least discriminant analysis (PLS-DA).	80
Figure 3.7	Biotin concentration, FAPy-Adenine and pimelic acid intensity showing significant biological differences between naïve host and lysogen during growth and antimicrobial challenge.	85
Figure 3.8	Heatmap generated by metabolic levels of 81 metabolites using HCA and DM.....	87
Figure 4.1	CGView-derived schematic of the Φ 24B genome; the concentric rings include the annotation, location and direction of expression.....	96
Figure 4.3	Clustered column graph representing percentage increase in CFU of single (ϕ 24B:: Δ Kan, dark grey) and double (ϕ 24B:: Δ Kan, ϕ 24B:: Δ Cat, light grey) MC1061 lysogens, under bile salt conditions.	102
Figure 4.4	Clustered column graph representing ϕ 24B:: Δ Kan MC1061 lysogens fold change in biotin gene expression compared to the naïve host MC1061.	104
Figure 4.5	Data gathered mapped to the biotin pathway.....	105
Figure 4.6	Fatty acid methyl ester profile at 6 hr under standard growing conditions.....	107
Figure 4.7	Fatty acid methyl ester profile at 6 hr grown in presence of 50 μ mol 8-Hydroxyquinoline (bactericidal drug).....	107
Figure 4.8	Fatty acid methyl ester profile at 6 hr grown in presence of 50 μ mol Chloroxylenol (bacteriostatic drug).	108
Figure 4.9	Clustered column graph representing ϕ 24B:: Δ Kan MC1061 lysogens fold change in fatty acid gene expression compared to the naïve host MC1061.....	109
Figure 4.10	Map of the fatty acid pathway.....	110
Figure 4.11	Clustered column graph representing ϕ 24B:: Δ Kan MC1061 lysogens fold change in lipid gene expression compared to the naïve host MC1061	111

Figure 4.12	Clustered column graph representing $\phi 24B::\Delta Kan$ MC1061 lysogens fold change in gene expression of peptidoglycan synthesis (A) and peptidoglycan associated factors (B) compared to the naïve host MC1061	112
Figure 4.13	Plot representing the fatty acid methyl esters and cell growth of the lysogen and naïve host over 18 hours under increasing concentrations of 8-hydroxyquinoline.....	115
Figure 4.14	Plot representing the fatty acid methyl esters and cell growth of the lysogen and naïve host over 18 hours under increasing concentrations of chloroxylenol.....	117
Figure 4.15	VIP plots representing the fatty acid methyl esters of the lysogen and naïve host over 18 hours under increasing concentrations of 8-hydroxyquinoline.....	119
Figure 4.16	Heatmap of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial.....	121
Figure 4.17	PLS-DA of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial.....	123
Figure 4.18	Biplot of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial.....	124
Figure 4.19	Heatmap of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline	126
Figure 4.20	PLS-DA of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline	127
Figure 4.21	Biplot (A) and VIP plot (B) of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline.....	128
Figure 4.24	Gram negative bacterial cell membrane structure, highlighting the target sites of antimicrobials 8-hydroxyquinoline and chloroxylenol.....	136
Figure 5.1	Q Exactive LCMS schematic.....	147
Figure 5.2	Map of the program ‘CRACCD’ GUI.....	154
Figure 5.3	Map of the program ‘CRACCD’ method for preliminary compound gathering. ...	155
Figure 5.4	Map of the program ‘CRACCD’ method for assessing and altering compound identities when separate identities have been given for the same compound.....	157

Figure 5.5	Map of the program ‘CRACCD’ method for identifying compounds of interest across datasets and conditions.....	159
Figure 5.6	Map of the program ‘CRACCD’ method for plotting compound table.....	160
Figure 5.7	Program windows associated to CRACCD identification of compounds of interest	164
Figure 5.8	Program windows associated to CRACCD identification of compounds of interest.	165
Figure 5.9	Program windows associated to CRACCD ‘compounds of interest’ clean-up.....	168
Figure 5.10	Program windows associated to CRACCD ‘compounds of interest’ profile tabulation of all datasets.....	171
Figure 5.11	Program windows associated to CRACCD plotting tabulated compound data	176
Figure 6.1	Cost per raw Megabase of DNA sequence compared to computer hardware related Moores law calculations	181
Figure 6.2	The Increase in the rate and amount of sequencing	182
Figure 6.3	‘GGOSS’ GUI main menu (tab 1, 2, 3, and 4).....	187
Figure 6.4	‘GGOSS’ GUI ‘Drag and Drop’	188
Figure 6.5	‘GGOSS’ GUI Cutadapt settings menu	190
Figure 6.6	‘GGOSS’ GUI Sickle settings menu.....	191
Figure 6.7	‘GGOSS’ GUI Khmer settings menu.....	192
Figure 6.8	‘GGOSS’ GUI FastQC menu.....	193
Figure 6.9	‘GGOSS’ GUI QUAST settings	193
Figure 6.10	‘GGOSS’ GUI SPAdes settings menu	195
Figure 6.11	‘GGOSS’ GUI PRICE settings menu	196
Figure 6.12	‘GGOSS’ GUI PRICE input/output settings.....	197
Figure 6.13	‘GGOSS’ GUI PRICE parameter settings.	198
Figure 6.14	‘GGOSS’ GUI PRICE filter read settings.....	199
Figure 6.15	‘GGOSS’ GUI PRICE filter contig settings.....	200
Figure 6.16	‘GGOSS’ GUI BLAST main menu and settings.	200
Figure 6.17	‘GGOSS’ GUI BWA settings	201

Figure 6.18	‘GGOSS’ GUI Ragout settings	202
Figure 6.19	‘GGOSS’ GUI MUMmer settings	203
Figure 6.20	‘GGOSS’ GUI mothur step selection.....	205
Figure 6.21	‘GGOSS’ GUI mothur run settings.....	206
Figure 6.22	‘GGOSS’ GUI PIPITS settings.....	207
Figure 6.23	‘GGOSS’ GUI MetaPhlAn settings	208
Figure 6.24	Artemis GUI.....	209
Figure 6.25	‘GGOSS’ GUI Prokka main menu.....	210
Figure 6.26	‘GGOSS’ GUI community distribution settings.....	213
Figure 6.27	‘GGOSS’ GUI Prokka settings menu	214
Figure 6.28	‘GGOSS’ GUI main menu (tab 5).	215
Figure 6.29	‘GGOSS’ GUI tool path settings.	216
Figure 7.1	Illumina SBS workflow. Core steps involved in the sequencing by synthesis technique employed in Illumina sequencing technology, from sample prep to nucleotide imaging and data collection	232
Figure 7.2	The number of currently known prokaryotic and eukaryotic viral families within each viral genomic type.	238
Figure 7.3	Structure-based viral lineages mapped onto current ICTV taxonomy with each lineage coloured separately.....	238
Figure 7.4	Seven classes of viruses distinguished by genome replication and encapsidation strategies	239
Figure 7.5	Stacked plots illustrating relative abundance of bacterial and viral communities. .	245
Figure 7.6	Illustrates fold change in bacterial (blue), FVP (red), and lysogenic (green) taxonomic richness between time points 1 and 2. Communities were normalised prior to comparison by rarefaction, and/or calculation of relative abundance. Taxonomic diversity was calculated by Reciprocal Simpson index and Bray-Curtis dissimilarity was used to compare communities.	246
Figure 7.7	Principle coordinates analysis of bacterial communities based on bray-curtis dissimilarity	247

Figure 7.8	Principle coordinates analysis of temperate (a, c) and lytic (b, d) viral communities based on bray-curtis dissimilarity	248
Figure 10.1	Triangulation between the Microbiome, the Immune System, and Cancer (Zitvogel, Ayyoub et al., 2016).....	345
Figure 10.2	Major microbial metabolites formed from dietary and environmental compounds that are involved in the initiation and/or progression of colorectal cancer	346
Figure 10.3	Potential mechanistic role of the gut microbiota in the aetiology and progression of eating disorders.	349
Figure 10.4	Proposed mechanism of action of prebiotics. IBD, inflammatory bowel disease; IBS, irritable bowel syndrome (Hamer, De Preter et al., 2012).	350
Figure 10.5	Percentage differences of metabolites present in MC1061 and ϕ 24B,	354
Figure 10.6	Map of the peptidoglycan pathway	364
Figure 10.7	Data normalisation of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial	366
Figure 10.8	Separate normalisation of 8-hydroxyquinoline and chloroxylenol fatty acid methyl ester data.	366
Figure 10.9	Dendrogram of CRACCD's example compound dataset	369
Figure 10.10	Heatmap of CRACCD's example compound dataset	370
Figure 10.11	Correlation plot of CRACCD's example compound dataset	371
Figure 10.12	Scree plot of CRACCD's example compound dataset.....	372
Figure 10.13	PCA Dot plot of CRACCD's example compound dataset.....	373
Figure 10.14	PCA biplot of CRACCD's example compound dataset.....	374
Figure 10.15	PCA of CRACCD's example compound dataset.....	375
Figure 10.16	shows relative abundance of phyla (a) and top twenty most abundant genera (b) observed between early and late time points	384

Table of Tables

Table 1.1	Commensal bacterial species that confer protection against pathogens.	40
Table 1.2	Changes in the Gut Microbiota Associated with Disease.	43
Table 1.3	Transit and abundance of ingested bacteria	46
Table 1.4	Known gut viro types according to culture techniques and metagenomics	48
Table 2.1	Antimicrobial stock solution content	56
Table 3.1	Statistics for compound ID's (sustained with reputable MSMS fragmentation) related to known bacterial pathways.....	83
Table 4.1	R2 and Q2 scores used for validating PLS-DA accuracy in group differentiation	118
Table 5.1	Example of an Input dataset.....	162
Table 5.2	Example of compounds of interest file	166
Table 5.3	Example of the cleaned up 'compounds of interest' file.....	169
Table 5.4	Example total data set file containing all of the conditions.	172
Table 5.5	Example of the final averaged tabulated data that represents the changes and appearances in the compounds of interest from all of the conditions.	173
Table 5.6	List of plots that can be graphed in CRACCD.....	177
Table 6.1	Range of some of the more prominent open source software tools available for genomic analysis, where tools incorporated into GGOSS are in highlighted in bold.....	218
Table 7.1	Describes clinical and demographic data for all 5 patients enrolled on this study.	241
Table 7.2	Shows differences in total counts (being the sum of all counts for all taxa in all samples), total taxa (being the counts of all taxa identified across all samples) and mean counts per taxa between frozen and non-frozen samples for both temperate and lytic viral taxa.	242
Table 7.3	Highlights similarites and unique features as a whole when using FVP, inducible viruses, and bacterial community analysis within our patient cohort	250
Table 10.1	Regulatory elements of bacteriophage Lambda.....	343
Table 10.2	The immunological effects of gut microbiota (Botticelli, Zizzari et al., 2017)	347
Table 10.3	Prebiotics: Randomised controlled trials	351
Table 10.4	Probiotics: Randomised controlled trials	352

Table 10.5	Synbiotics: Randomised controlled trials.....	353
Table 10.6	Putative metabolite identities and statistics.....	355
Table 10.7	T-test	360
Table 10.8	Statistically significant differences using area under the curve	361
Table 10.9	PLS-DA statistics.....	363
Table 10.10	Identified cell wall fatty acid methyl esters and their known functions.....	367
Table 10.11	The genomic related databases within Entrez (September 2015).	376
Table 10.12	dsDNA viral characteristics	378
Table 10.13	ssDNA viral characteristics.....	379
Table 10.14	dsRNA viral characteristics.	380
Table 10.15	ssRNA viral characteristics.....	382
Table 10.16	rtDNA and rtRNA viral characteristics.....	383

Acknowledgements

I would like to thank my supervisor Dr Darren Lee Smith for his continued support and encouragement and for giving me this opportunity, which has allowed me to improve and broaden my skills as a developing scientific researcher. I am grateful for his additional efforts in enabling me to participate in international conferences, which have been invaluable experiences both academically and personally. Overall, this opportunity has allowed me to pursue a true passion of mine, which I am excited to take forward and hopefully gain a fulfilling career within the scientific community.

I would also like to offer my gratitude to Dr Simon Bridge and Dr John Lodge, who have been a source of continued support and guidance throughout my studies. I would like to thank Nick Embleton, Janet Berrington and Andrew Sailes for providing me with samples from the Freeman Hospital and I am grateful for their support in my research.

I would like to express my appreciation to Paul Agnew, for his support and understanding throughout my PhD. I am especially grateful to Dr Adnan Tariq, Dr Andrew Nelson and Greg Young, whose support and friendship have been invaluable throughout.

I am indebted to my family for their foundation of support, both prior to and during my PhD, without which this would not have been possible. I cannot thank them enough for their patience and understanding, and for keeping me motivated throughout. I am also grateful to my partner Katy Taylor, who has been with me every step of the way, to celebrate the achievements and to motivate me through the more difficult times.

Finally, I would like to thank The Leverhulme Trust for supporting me financially from my undergraduate studies and through my PhD.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others. The work was in collaboration with Freeman Hospital in Newcastle upon Tyne.

Any ethical clearance for the research presented in this thesis has been approved. RE07 – 06 - 12940

Name: Giles Samuel Holt

Signature:

Date:

List of Abbreviations

%: Percent

°C: Degree Celsius

Abi: Abortive infection

APCI: Atmospheric pressure chemical ionization

APPI: Atmospheric pressure photo-ionization

ATP: Adenine triphosphate

ANOVA: One-way analysis of variance

Bacteriophage (phage)

Bash/sh: Bourne-again shell/Bourne shell

BLAST: Basic Local Alignment Search Tool

bp: Base pair

cAMP: Cyclic adenosine monophosphate

Cas: CRISPR-associated genes

cDNA: complementary DNA

CDS: Coding sequence

cdtB: Cytolethal distending toxin

CE-MS: Capillary electrophoresis mass spectrometry

CFU: Colony forming units

CNS: Central nervous system

contig: overlapping sequence

CRACCD: Cross run analysis for comparable compound data profiling

CRISPR: Clustered regularly interspaced short palindromic repeats

CV%: Coefficient of variation percentage

DDGE: Denaturation Gradient Gel Electrophoresis

dH₂O: distilled water

DNA: Deoxyribonucleic acid

ds: Double-stranded

dsDNA: Double stranded DNA

E. coli: Escherichia coli

EDTA: Ethylenediaminetetraacetic acid

EHEC: *Enterohemorrhagic Escherichia coli*

EM: Electron microscopy

ER: Endoplasmic reticulum

ERAD: ER-associated degradation

ESI: Electrospray ionisation

FA: Fatty acid

FT-MS: Fourier transform mass spectrometry

FVP: Free viral particle

g: Gram

GC/MS: Gas chromatography mass spectrometry

GGOSS: Genomic analysis using open source software

GRA: Genome relative abundance

GUI: Graphical user interface

HESI: Heated electrospray ionization

HGT: Horizontal gene transfer

HILIC: Hydrophilic interaction liquid chromatography

HK97: Hong kong phage 97

HMM: Hidden Markov model

hr: Hour

HUS: Haemolytic-uremic syndrome

HUVECs: Human umbilical vein endothelial cells

ICTV: International committee on taxonomy of viruses

IDBA-UD: Iterative De Bruijn Graph de novo Assembler - Uneven Sequencing Depth

IHF: Integration host factor

ITS: Internal transcribed spacer

kb: Kilobase

Kbp: Kilo base pair

kDa: KiloDalton

k-mer: Short DNA sequence of fixed (K) length

LB broth: Luria bertani broth

LCA: Lowest common ancestor

LC-MS: Liquid chromatography and mass spectrometry

LOS: Late onset sepsis

LPS: Lipopolysaccharide
m/z: Mass to charge ratio
M: Molar
MALDI: Matrix-assisted laser desorption ionization
Mauve: Multiple genome alignment (program)
Mb: Megabase
Mbp: Mega base pair
mg: Milligram
ml: Millilitre
MLE: Maximum likelihood estimation
mM: Millimolar
MOI: Multiplicity of infection
mRNA: Messenger RNA
MS/MS: Mass spectrometry/Mass spectrometry
MS: Mass spectrometry
mz: mass-to-charge ratio
NaCl: Sodium chloride
NaOH: Sodium hydroxide
NCBI: National Centre for Biotechnology Information
NEC: Necrotising enterocolitis
NFLX: Norfloxacin
NGS: Next-generation sequencing
NIH: National Institutes of Health
NMR: Nuclear magnetic resonance
NP: Normal-phase
OD 600nm: Optical density 600 nanometres
OD: Optical density
ORF: Open reading frames
OSS: Open source software
PCA: Principle component analysis
PERMANOVA: Permutational multivariate analysis of variance
Pfam: Protein families
PFU: Plaque forming units

Phage: Bacteriophage
PHAST: Phage search tool
PKA :Protein kinase A enzyme
PLS-DA: Partial least squares regression - discriminant analysis
PRICE: Paired-Read Iterative Contig Extension
RAST: Rapid annotation using subsystem technology
RFLP: Restriction fragment length polymorphism
RM: Restriction modification
RNA: Ribonucleic acid
ROS: Reactive oxygen species
RP: Reverse-phase
RPM: Revolutions per minute
rRNA: Ribosomal RNA
s: Seconds
SBS: Sequencing by synthesis
SD: Standard deviation
SEM: Standard error of mean
SNP: Single Nucleotide Polymorphism
SPAdes: St. Petersburg genome assembler (program)
SRA: Sequence Read Archive
ss: Single-stranded
STEC: Shigatoxigenic *Escherichia coli*
SVs: Structural variants
TA: Toxin antitoxin system
TGN: *Trans*-golgi network
tRNA: Transfer RNA
TTGE: Temporal temperature gradient electrophoresis
TTP: Thrombotic thrombocytopenic purpura
UK: United Kingdom
UV: ultra violet
VelvetOptimiser: automated velvet de novo assembler (program)
w/v: Weight/volume
WGS: Whole genome shotgun

WHO: World Health Organisation

YAD: Yet another dialog

μg : Microgram

μl : Microlitre

μm : Micrometer

Chapter 1. Introduction

1.1 *Escherichia coli*, and shigatoxin (stx) encoding *Escherichia coli*

1.1.1 *Escherichia coli*

E. coli is a Gram negative, rod shaped, coliform, and facultatively anaerobic bacteria. There are pathogenic and non-pathogenic strains of *E. coli*. While numerous strains of *E. coli* are commensal gut bacteria, there are many serotypes that commonly cause food poisoning, with the fecal-oral route being the main form of transmission.

1.1.2 Stx toxin

The toxic effect of Stx is caused through protein synthesis inhibition in eukaryotic cells, by depurination of adenine in 28S rRNA (N-Glycosidase) (O'Brien, 1998). The trafficking of Stx within the cell commences with the binding to the plasma membrane, leading to asymmetric reduction in the membrane area. The toxin moves from early endosome, to the *trans*-golgi network (TGN), and finally its retrotranslocated into the cell cytosol via the endoplasmic reticulum (ER) (Figure 1.1). Varying Stx toxin subtypes can have slightly altered characteristics, for example Stx2B (pentameric B) does not increase cyclic adenosine monophosphate (cAMP) levels but may activate protein kinase A enzyme (PKA) by a cAMP-independent mechanism (Liu, Huang et al., 2011). Compared to Stx1, Stx2 can activate different signalling pathways, possibly explaining its increased likelihood in causing diarrhoea associated haemolytic uremic syndrome (HUS) than *E. coli* expressing only Stx1 (Liu et al., 2011). Toxic characteristics can also be affected by *E. coli* types, for example; symptoms of HUS are expressed commonly by Stx encoding *Enterohemorrhagic Escherichia coli* (EHEC) (Kaper, 1998), this may be due to their enhanced ability to attach closely to intestinal epithelial cells and destroy micro villi (a phenomenon caused by LEE locus) (Agin, Zhu et al., 2005). Hence bacterial producers and Stx compositions are important to the binding and pathogenic capabilities.

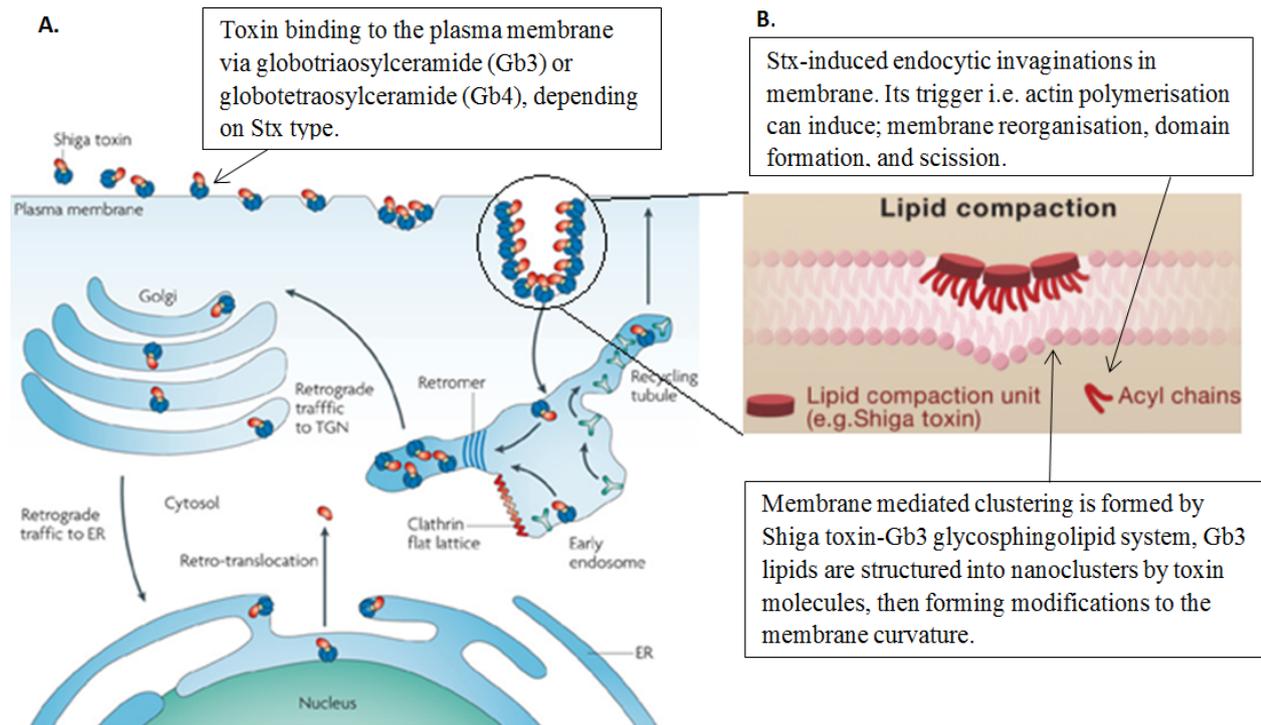


Figure 1.1 Stx retro-translocation and intracellular trafficking, A: (Johannes & Romer, 2010), B: (Johannes & Mayor, 2010). **A:** Depiction of intracellular trafficking of Stx. Toxin binds to the plasma membrane inducing membrane-mediated clustering, local spontaneous curvature, and the toxin-driven formation of endocytic invaginations. **B:** Membrane bending is caused through local compaction of glycosphingolipids in one leaflet, leading to an asymmetric reduction in membrane area. Retrograde sorting in early endosomes carried out on toxin, in which retrograde tubules form in a clathrin-dependent manner (Stx preferentially localize to the tubular environment). Retrograde tubules are processed by scission in a retromer-dependent manner. Stx bypass the late endocytic pathway and transfer straight from early endosome to the TGN, then to the ER. Stx use the ER-associated degradation (ERAD) machinery to enable retro-translocation into the host cell cytosol

1.1.3 Stx toxin compositions and binding properties

Shiga toxin is composed of both a cytotoxic A and a pentameric ring of B subunits, with the Stx receptor on the eukaryotic cells being glycolipid Gb3 if Stx1 or Stx2, but Gb4 if Stx2 E (Figure 1.2). Gb3 is in the sub-mucosal vasculature, and Gb4 is based on colonic epithelial cells and cell surface (Zumbrun, Hanson et al., 2010). The pentameric B subunits bind Gb3 on susceptible cells, Stx1B and Stx2B activate different signalling pathways in human umbilical vein endothelial cells (HUVECs) (Liu et al., 2011). Stx1 and Stx2 share ~55% amino acid similarity, possibly explaining their altered pathogenicity, as Stx2 is associated with increased cytotoxicity

(Nakao, Kiyokawa et al., 1999). Differences include the increase in Stx1 production under low-iron conditions, while Stx2 production is activated by phage-inducing agents, such as mitomycin C.

Additionally, alternative promoters are required for the expression of Stx1 and Stx2 in Stx-encoding phages (Shimizu, Ohta et al., 2009), and though both toxins are equally as harmful to Vero cells (O'Brien, 1998), Stx2 is more toxic to human renal cells (Kaper, 1998) (hence more likely to cause HUS). Stx1 and 2 bind to colonic epithelia and to colonic epithelial cell line (HCT-8) which have Gb3 present on them, acting as the receptors, helping bind and internalise Stx1 and Stx2 (Zumbrun et al., 2010). Human intestinal tissue and cultured colonic cells contain Gb3 synthase mRNA and the alternate Shiga toxin receptor Gb4 (Zumbrun et al., 2010).

There are many subtypes of Stx1 and Stx2, for example Stx1a, Stx1c, Stx1d, Stx2a, Stx2b, Stx2c, Stx2d, Stx2d-activatable, Stx2e, and Stx2f (Mora, Herrera et al., 2011). The AB holotoxin structures are identical in Stx1 and Stx2, but they can be genetically distinct enough at the protein level to require differing antibodies for detection, and have varying toxicities (Mohawk & O'Brien, 2011). Furthermore, various Stx producing bacteria other than *E. coli* have been found, with toxins isolated in bacteria such as *Citrobacter freundii* (Schmidt, 1993, Tschape, 1995), *Vibrio cholera* (O'Brien, Chen et al., 1984), *Enterobacter cloacae* (Paton, 1996), and *Aeromonas* spp (Haque, Sugiyama et al., 1996). This phenomenon is associated with the A subunit of the Shiga toxin positioned on the Stx phage or chromosome (O'Brien, 1998), advocating toxin dissemination as a result of temperate bacteriophages encoding to other bacterial species (Casas, Sobrepena et al., 2011).

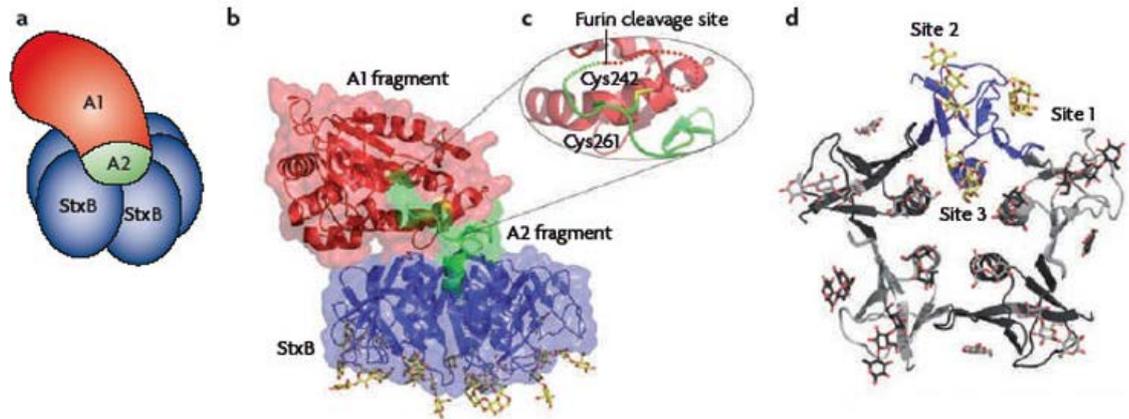


Figure 1.2 Displays structure and binding/cleavage sites of Stx (Johannes & Romer, 2010). A: Schematic of Shiga holotoxin; one A subunit (StxA), cleaved into fragments A1 and A2, and five B fragments that make up the homopentameric B subunit (StxB). B: Ribbon diagram of Stx, depicting Gb3-binding sites on StxB. Gb3 is shown in a ball-and-stick representation. C: Enlargement of StxA showing the disulphide bond (between Cys242 and Cys261) that links the A1 and A2 fragments at the site of furin cleavage (Arg25-Met252). D: Ribbon diagram of a StxB subunit from the membrane-oriented surface, displaying the three Gb3-binding sites. Gb3 represented by ball-and-stick (Johannes & Romer, 2010).

1.1.4 Shigatoxin encoding *Escherichia coli* (STEC)

Diarrhoea occurs world-wide and causes 4% of all deaths and 5% of health loss to disability. It is most commonly caused by gastrointestinal infections which kill around 2.2 million people globally each year (WHO, 2014). Children, the elderly, and people who have weak immune systems are most likely to contract intestinal infections. Gut infections are predominantly caused by a range of bacteria that include; *Salmonella*, *Shigella*, *Escherichia coli*, several species of *Campylobacter* genus, *Clostridium difficile*, *Listeria monocytogenes*, *Staphylococcus aureus*, and several species of the *Vibrio* genus. My research focuses on investigating the metabolic, genetic and growth profiles of *E. coli* (specifically Shigatoxin-encoding *E. coli*) and the interaction with their co-evolving bacteriophage.

Human infections of Shiga toxin encoding *Escherichia coli* (STEC) are prevalent worldwide, and potentially fatal. The most notorious STEC serotype being O157:H7, although there are now over 500 different serogroups detailed (Allison, 2007). Debilitating symptoms of Stx within humans include; diarrhoea, haemorrhagic colitis, thrombotic thrombocytopenic purpura (TTP) and haemolytic-uremic syndrome (HUS) (Kawano, Okada et al., 2008, O'Brien, 1998). The first reported STEC outbreak was in America 1982 (Eppinger, Mammel et al., 2011, Riley, Remis

et al., 1983), where the timeline detailed in Figure 1.3 describes incidence of a selection of large outbreaks over the last 29 years.

Stx is transmitted via food and water from animal reservoirs (Johansen, Wasteson et al., 2001) and human to human contact (Viazis & Diez-Gonzalez, 2011). As well as previously stated symptoms, STEC-infections can also exhibit neurological manifestations of disease (Cimolai, Morrison et al., 1992), which was further supported by Kurioka et al, 1998 who revealed that Stx can affect the central nervous system (CNS), occasionally resulting in death (Kurioka, Yunou et al., 1998). Stx is most commonly known for the toxin and its life threatening inference on HUS. In areas with poor medical facilities, such as sub-Saharan Africa, this syndrome can have a mortality rate of 17.3% (Bhimma, Rollins et al., 1997).

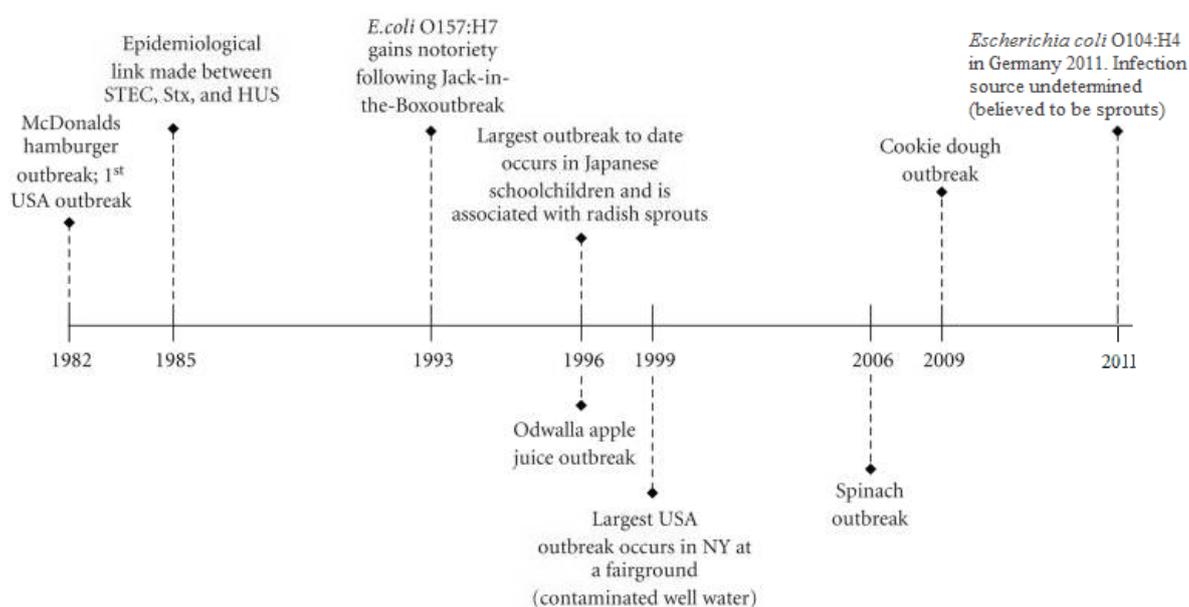


Figure 1.3 STEC major outbreaks time lined since 1982. Occurrences are listed by the year of incidence and the context of source contamination that linked Stx (modified from Mohawk & O'Brian, 2011).

1.1.5 Shiga toxin encoding bacteriophages (Stx-phages)

Stx-phages are lambdoid like phages, and are thus temperate phages, the life cycles of a temperate phage are 2-fold including lysogenic and lytic life cycles (see section 1.2). The spread of Shiga toxins by Stx-phages is directly linked to lysogen stability, which is lessened with the phenomenon of multiple lysogeny (Fogg, Saunders et al., 2012), due to the toxins being released upon host cell lysis. In natural selection only the most successful mutations or conveyed adaptations are retained within the phage gene pool, and the more generalised beneficial changes can ultimately become co-selected by the host bacteria.

Stx-phages not only have the ability to horizontally transfer the Stx toxin associated with the severe downstream infection of Shigatoxigenic *Escherichia coli* (STEC) infection, they also disseminate accessory genes that are well conserved throughout sequenced Stx-phages but have no assigned function. The functionality of these putative coding regions is difficult to determine as most work is intrinsically dedicated to pure culture and standard laboratory conditions and so gene function outside of the basic life cycle including infection, integration, genetic regulation of life cycle decision, replication propagation is difficult to ascertain. If we cannot associate function to the gene and through gene alignment no putative function of genetic motif can be identified then we miss the intricacies these viral entities disseminate and how they impact the host in either an advantageous or deleterious way.

Stx-phages are enterobacteria viruses in the caudovirales order, and their modulation of host virulence is not uncommon. It's thus important to understand the influence viruses have on their hosts, particularly in association to human health.

1.2 Types of viruses

The human gut contains a plethora of viruses that include giant viruses, plant-derived viruses, and an abundance of bacteriophages. These viruses cover a diverse range of genetic viral types, these being; dsDNA viruses, ssDNA viruses, dsRNA viruses, ssRNA viruses, and reverse transcribing viruses.

1.2.1 DNA viruses

dsDNA viruses are represented relatively evenly by both eukaryotic and prokaryotic alike (appendices section 10.6.1). dsDNA viruses make up nearly all bacterial viruses identified in the gut to date. dsDNA viruses consist of double stranded DNA and replicate with a DNA-dependent DNA polymerase. Details on their replication can be found in appendices section 10.6.1. Most dsDNA viruses contain a single genome of linear dsDNA, though there are still many circular genomes in the 'Type I' grouping. These viruses rarely infect plants, with their host range found almost entirely in the animal and bacterial population. Large dsDNA viral genomes have high genomic plasticity and often engage in horizontal gene exchange (Filee, 2013, Krupovic, Prangishvili et al., 2011, Yutin & Koonin, 2012). Most dsDNA eukaryotic viruses are nuclear-replicating, with the exceptions of the Poxviridae and Mimiviridae families. Unlike most other eukaryotic dsDNA viruses the Poxviruses and Mimiviruses replicate in the cytoplasm of their host, and encode all of the transcription and replication machinery themselves (Moss, 2013). Large eukaryotic dsDNA viruses originate from two distinct groups of bacteriophages; the Tectiviridae family and the Caudovirales order (Koonin, Krupovic et al., 2015).

Prokaryotic viruses are mostly dsDNA viruses, which consist of 14 families (see section 10.6.1). The majority of dsDNA viruses fall under the Caudovirales order, consisting of Siphoviridae, Myoviridae, and Podoviridae families (Lefkowitz, Dempsey et al., 2018). Commonalities of virions found within the Siphoviridae family are separate assembly of head and tail structures, where tails are long, thin, noncontractile, and often flexible (Lefkowitz et al., 2018). Commonalities of virions found within the Myoviridae family are long, thick, rigid-like contractile tails, with tail cores wrapped in a helical contractile sheath, which is separated from the capsid by a neck (Lefkowitz et al., 2018). When sheath subunits slide over one another, these helical sheaths

contract, shortening and thickening the sheath, thus bringing the tail core to contact the bacterial membrane (Lefkowitz et al., 2018). Of the tailed phages myoviruses tend to have greater particle weights, larger capsids, and higher DNA content (Lefkowitz et al., 2018), as well as a greater sensitivity to osmotic shock and freezing/thawing (Lefkowitz et al., 2018). Commonalities of virions found within the Podoviridae family are short, noncontractile tails, which are assembled after capsid assembly (Lefkowitz et al., 2018).

Viruses with ssDNA genomes have the lowest diversity of the major viral types (not including rt viruses), nonetheless they are widespread in the environment and are medically (Kekarainen & Segales, 2009, West, Bystrom et al., 1999), ecologically (Suttle, 2007a), and economically (Rybicki, 2015) important pathogens. All identified ssDNA viruses are non-enveloped and non-tailed, and the majority have a circular genome (with exception to the Parvoviridae family) (see appendices section 10.6.2). Only two bacteriophage families are categorised as ssDNA viruses, these are Inoviridae and Microviridae. While HGT does occur in rare cases (Diemer & Stedman, 2012, Krupovic, Ravantti et al., 2009, Roux, Enault et al., 2013), ssDNA viruses have difficulty acquiring new genetic material compared to the larger dsDNA viruses. Reasons for this may include their smaller capsid volumes, lower capsid pressure for genome compactness, and pervasive functional secondary structures within their genomes (Muhire, Golden et al., 2014). Similar to other eukaryotic viruses, eukaryotic ssDNA viral origins have been linked to prokaryotic and ssRNA viruses, however their prokaryotic similarities are most prominently associated to the prokaryotic rolling cycle-replicating plasmids. The evolution of ssDNA viruses may have occurred via a combination of genes from these plasmids and positive-sense RNA viruses (Krupovic, 2013, Stedman, 2013).

1.2.2 RNA viruses and reverse transcribing viruses

Nearly all currently identified dsRNA viruses are eukaryotic, with the only known family of dsRNA bacterial viruses being Cystoviridae (see appendices section 10.6.3), though RNA phage diversity is suggested to be far greater (Krishnamurthy, Janowski et al., 2016).

ssRNA viruses can be either positive (+) (type IV) or negative (-) (type V) sense, positive or negative depending on the polarity or sense of the RNA, i.e. 3'-5' (-) and 5'-3' (+). ssRNA viral hosts are almost entirely eukaryotic, the only known ssRNA viral family that infects and replicates in a bacterial host cell is the Leviviridae family, which infect primarily enterobacteria and some other proteobacteria (Bollback & Huelsenbeck, 2001). Enterobacteria and proteobacteria are among the most common gut microbes. ssRNA viral characteristics can be seen in appendices section 10.6.4.

There are no known prokaryotic reverse transcribing viruses, as their only currently known host are eukaryotic (see appendices Table 10.16). There are two types of reverse transcribing viruses, RNA and DNA. Reverse transcribing (rt) RNA (type VI) and DNA (type VII) viruses replicate by interconverting their genomes between RNA and DNA, with genome sizes ranging from 3-11 kb. RT-virus virions initially package their capsids with their genomes in RNA form, however not all RT viruses exit their host with their genome still RNA coded. For example virions of foamy retro-viruses and hepadnaviruses may reverse-transcribe their RNA genome into DNA before the virion exits its host cell

1.3 Phage life cycles

1.3.1 Adsorption and Infection

1.3.1.1 Adsorption

Phage adsorption is the mechanism in which a virion adheres to its host, which involves tail fiber interaction with cell wall structures. Phage have no structures responsible for virion motion, the adsorption process is the result of random phage-cell collisions. Phage tails have been shown to preferentially adhere to bacterial poles (Edgar, Rokney et al., 2008). This may be related to the likely structural consistency of the target region. Examples of pole preference includes *stxI* carrying *E. coli* phages such as P1, T4 and T7 (Hashemolhosseini, Holmes et al., 1994, Kaiser & Dworkin, 1975, Steven, Trus et al., 1988), as well as other phages like T7-like Yersinia phage A1122 (Garcia, Elliott et al., 2003) and T4 like vibrio phage KVP40 (Miller, Heidelberg et al., 2003).

There are 5 classes of protein receptors; structural proteins interacting with peptidoglycan layer, specific and non-specific porins forming membrane channels, enzymes, receptors with high substrate affinity, and transport proteins responsible for secretion (Silva, Storms et al., 2016). Porin examples include; transmembrane protein OmpA (Morona, Klose et al., 1984), OmpC (Ho & Slauch, 2001, Parent, Erb et al., 2014) and OmpF (Zhao, Cui et al., 2013). Enzyme examples include the localised proteases, OmpT and OmpX, in the outer membrane, which are receptors for T-like phages (Hashemolhosseini et al., 1994). Active transport system examples include lipopolysaccharide receptors, both smooth and rough lipopolysaccharides. Smooth lipopolysaccharide incorporates 3 parts; Lipid A, Core, and O-chain, rough lipopolysaccharide is made up of just the lipid A and core (Alexander & Rietschel, 2001). Receptor type can have a significant effect on host range, this can be observed in the lipopolysaccharide receptors (Szczyka, Szumala-Kakol et al., 2010). As there is large variability of the O-antigen in smooth lipopolysaccharide, whereas rough lipopolysaccharide has a more conserved lipopolysaccharide core (Erridge, Bennett-Guerrero et al., 2002).

Phage receptors can also be found localised in the cell wall of Gram positive bacteria, these systems include; components of transport systems such as GamR (Davison, Couture-Tosi et al., 2005), teichoic acids (Brown, Santa Maria et al., 2013) and lipoteichoic acids (Raisanen, Schubert et al., 2004), and extracellular polysaccharides in the cell wall. Interestingly phage receptors are also found localised in capsular and slime polysaccharides, pili and flagella (see appendices section 10.1.1).

The lambda model of phage adsorption has been extensively studied (Raisanen et al., 2004). Lambda tail J protein and tail tip protein 'I' binds to the porin outer membrane protein LamB (Wang, Hofnung et al., 2000). Interestingly the binding only becomes irreversible if the phage tail is attached to the phage head, preventing loss of receptor regions from non viable virion (Schwartz, 1975). The LamB porin is a maltoporin due to its necessity for growth on limiting concentrations of maltose (Ferenci, Schwentorat et al., 1980). LamB is a conserved structure concentrated irregularly on the bacterial poles (Gibbs, Isaac et al., 2004). The conserved nature of LamB may explain its use as a receptor for several other phages, including K10 (Roa, 1979) and TP1 (Moreno & Wandersman, 1980, Wandersman & Schwartz, 1978).

1.3.1.2 Infection

There are several mechanisms of phage nucleic acid injection after penetrating (or partially penetrating) the host cell, often no single mechanism is responsible for successful injection of nucleic acid. Some of the current known mechanisms include: diffusion, osmotic pressure, transcription, and translocation (Brownian ratchet mechanism) (Grayson & Molineux, 2007). Essential in any ejection method is the 'uncorking reaction' in which the DNA is released within the virion and becomes free to move (Molineux & Panja, 2013).

Random thermal agitation of the DNA causes it to move through the tail in an act of diffusion ejection (Grayson & Molineux, 2007). Pressure in the phage head derived from tightly packed nucleic acids can create an osmotic pressure, which is a mechanisms that can play a part in genome ejection in many phage (Grayson & Molineux, 2007, Sao-Jose, de Frutos et al., 2007).

Once a portion of the genome has made it into the cytoplasm of the host, transcription and/or translocation can occur to pull the entirety of the genome into the cell. It has been hypothesised that tight binding of DNA to intracellular proteins may cause translocation via a 'Brownian ratchet mechanism'.

In the case of Lambda, DNA is packaged into the capsid in a specific direction, one end known as the left end is inserted first. When the lambda DNA is released from the capsid, the right end exits first. An inner membrane protein called PstM is used to gain entry to the cytoplasm. However the exact mechanism in which the lambda genome reaches PstM through the peptidoglycan and periplasm is unknown.

1.3.1.3 Post infection and early gene expression

Once the linearised Lambda phage genome reaches the host cytoplasm it circularises at the two *cos* sites using DNA ligase and DNA gyrase. The decision toward a lytic or lysogenic life cycle (see Figure 1.4) then takes place, and is determined by early gene expression. A map of the genes that decide the lytic/lysogen decision is shown in Figure 1.5. Early gene expression is described in two steps, immediate early gene expression and delayed early gene expression. The immediate early genes are required for the transcription of the delayed early genes, which in turn are required for the transcription of the late genes. N and Q proteins are transcriptional antiterminators which are integral to the regulation of expression times. In the Lambda model immediate early gene expression is initiated by transcription at the *pR* and *pL* promoters via the subverted host RNA polymerase. This gene expression is limited by transcription terminators, the *pR* promoter terminators are *tR1*, *tR2*, *tR3*, *tR4* and the *PL* terminators are *tL1*, *tL2*, *tL3*.

N and *cro* genes are the first to be expressed and are thus the immediate early genes. Transcription of *N* and *cro* is initiated by host RNA polymerase transcription of the *P_R* and *P_L* promoters. The N protein binds to *nutL* and *nutR* located upstream of the transcription terminators, where the bound N and host cell proteins modify RNA polymerase upon it reaching the *nutL* and *nutR* sites. The modification of RNA polymerase allows it to transcribe through several

terminators, including the terminators t_{LI} and t_{RI}, thus N prevents the early termination of P_L and P_R promoters, allowing expression of delayed early genes (CII, CIII, and Q). Cro inhibits the P_{RE} promoter which has a mild impact on the lytic/lysogenic switch (inhibits cII production), however the operator effects play a larger role. Cro represses the promoters at O_R and O_L which inhibits the production of CI and CII (Miller et al., 2003).

Delayed early gene expression is the initial expression of CII, CIII and Q and the continued expression of N and cro. The Q protein modifies RNA polymerase allowing expression of the tR' terminator, preventing immediate lysis, and allowing late gene expression. CII increases gene transcription of cI via the P_{RE} promoter, which establishes but does not maintain, the lysogenic state. cIII controls the stability of cII by inhibiting the action of the hostencoded HflB(FtsH)-HflC-HflK protease complex. The CI and CRO competition is based around a double negative feedback loop, where CI protein represses *cro* gene and CRO protein represses *cI* gene. Lysis or lysogeny is dependent on which one gains control of the operator region, which is the genetic switch that ultimately determines whether a lytic or lysogenic cycle ensues.

The operator switch consists of O_{R1}, O_{R2}, O_{R3}, O_{L1}, O_{L2}, and O_{L3} (see Figure 1.6). These tandem CI binding sites (see Figure 1.16), can be bound by either cI or cro proteins (Johnson, Poteete et al., 1981, Ptashne, Jeffrey et al., 1980). These proteins have opposing effects when bound to the operator regions on phage gene expression. This is due to their differential order of binding to the operator sites.

Stimulation that promotes the eventual control of the operator switch comes from the physiological state of the host. The host factors determine if lysogeny takes place (presumably a phage evolved interaction to ensure suitable lysogeny), if interaction is insufficient, lysis occurs. Host factors influence; *cIII* mRNA stability, modulation of CII and CIII levels, direct inhibition of CII protein, and potentially even PL transcription. Host factor elements are; integration host factor (IHF), hostencoded HflB(FtsH)-HflC-HflK protease complex, cAMP and ppGpp levels, RNase III, HflD levels, DnaA and SeqA, and LexA (see appendices Table 10.1).

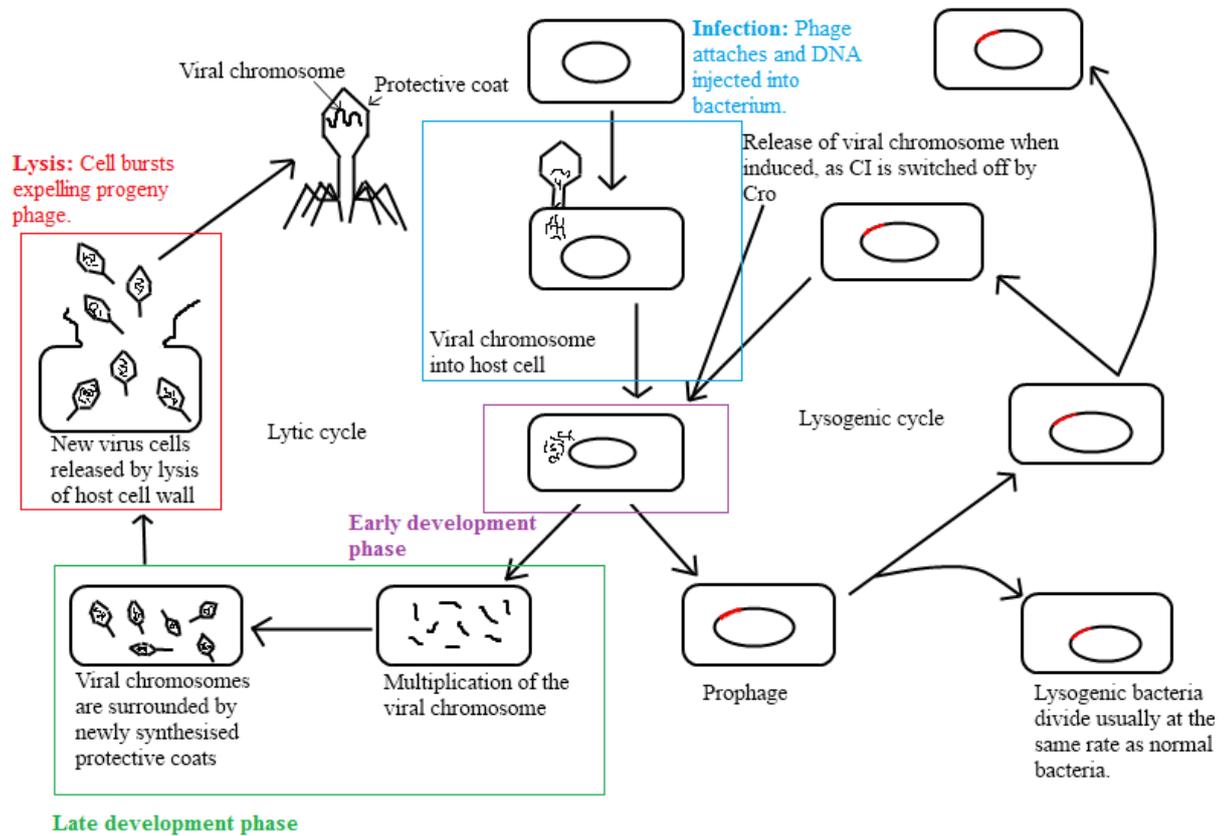


Figure 1.4 The phage lytic and lysogenic cycle. Lytic cycle: CI fails to overrule Cro in early development. Lysogenic cycle: CII and CIII cause an over powering synthesis of repressor and trigger inhibition of late gene transcription. Diagram designed around information from Cao et al, 2010 and Court et al, 2007.

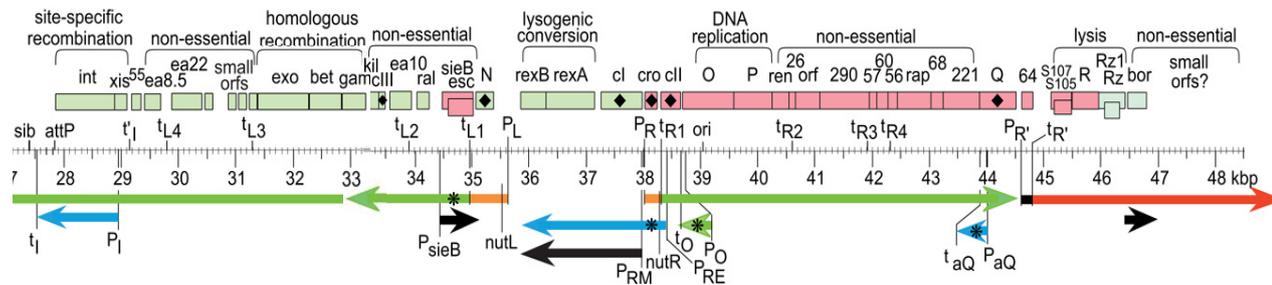


Figure 1.5 Map of the bacteriophage lambda chromosome that decides and carries out the lytic/lysogen decision. Edited from (Casjens & Hendrix, 2015). The linear virion chromosome is shown with a scale in kbp below. Rectangles indicate known genes with names and functional regions shown above; red genes are transcribed rightward and green leftward. Diamonds (♦) mark regulatory genes. Important DNA sites (e.g., P, promoters; t, terminators) are indicated below the genes. Thick horizontal arrows below indicate mRNAs: black, transcripts made in a lysogen; orange, immediate-early transcripts; green, early transcripts; red, late transcripts; blue, transcripts made in response to high CII levels. Asterisks (*) mark RNAs with regulatory activity.

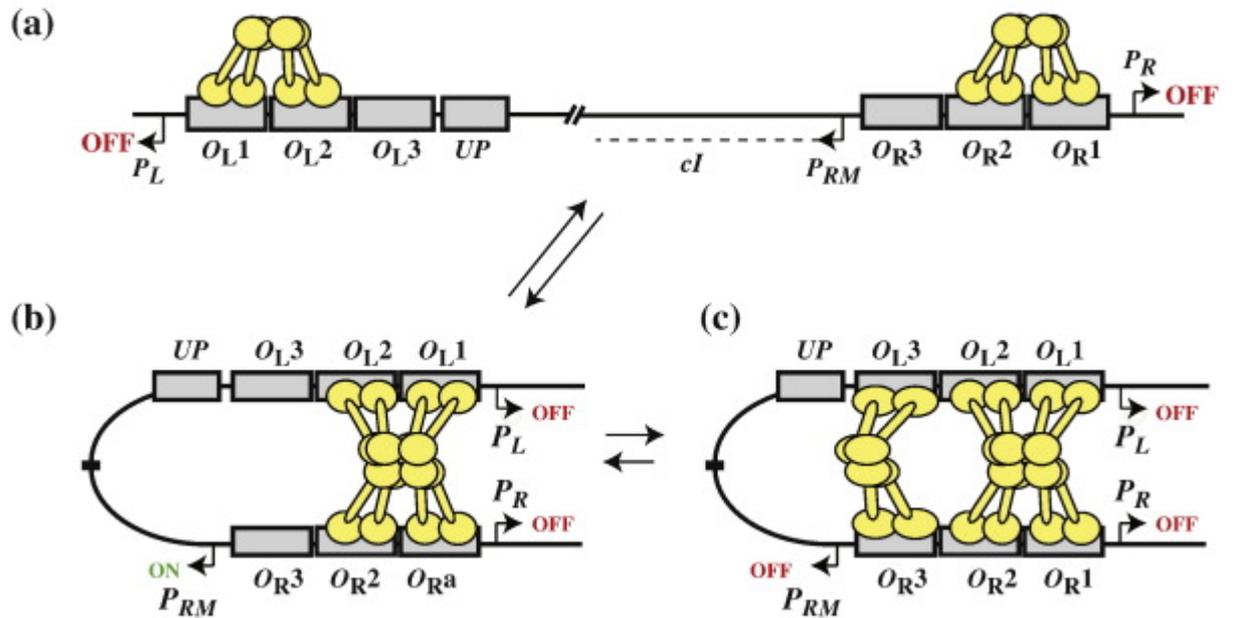


Figure 1.6 Regulation of PR and PL by CI repressor. (a) Cooperative binding of CI dimers to $O_{L1} \sim O_{L2}$ and $O_{R1} \sim O_{R2}$ represses P_L and P_R , respectively; binding to O_{R2} alone is sufficient to activate P_{RM} . (b) Looping permits the formation of a repressor octamer, which allows the repression of P_L and P_R and the activation of P_{RM} to occur at lower CI concentrations. In the looped configuration, the α -CTD of RNAPolymerase bound to P_{RM} can interact with the UP element adjacent to O_{L3} , which enhances the activation of P_{RM} . The looped orientation shown is antiparallel, which is the expected configuration in experiments reported here, because the distance between P_L and P_R is only 392 bp. (c) CI dimers bound to O_{R3} repress P_{RM} ; repression is facilitated by the interaction with CI dimers bound to O_{L3} (Lewis, Gussin et al., 2016)

1.3.1.4 The lytic life cycle

Both lytic and temperate viruses can carry out a lytic life cycle. The lytic virus life cycle is obligate, whereas the temperate viruses' lytic cycle is triggered. The lytic life cycle is the process in which a phage forms progeny phage using host mechanisms and exits the cell, usually via cell lysis, without incorporating itself into the host genome.

After immediate early gene expression (see 1.3.1.3) and sufficient N transcription, N protein interacts with RNA polymerase, through a modification of sorts that nullifies the termination sites, tL, tR1, and tR2. This leads to the production of longer transcripts.

The lytic switch starts with *cro* inhibiting the P_{RE} promoter as well as the O_R and O_L promoters. While inhibition of the P_{RE} promoter has a mild impact on the lytic/lysogenic switch with some inhibition of *cII* production, effects on the operator region play a larger role. *Cro*'s repression of the promoters at O_R and O_L significantly inhibits the production of *CI* and *CII* and maintains higher *CI* binding requirements for the operator region (Schubert, Dodd et al., 2007).

Host factors ultimately determine whether delayed early gene expression results in the lytic gene cascade. IHF stimulates leftward transcription and modulates *CII* and *CIII* levels, thus low IHF increases the chance for a lytic cycle. Hostencoded HflB(FtsH)-HflC-HflK protease complex destabilises *CII* stability, meaning higher levels gives rightward transcription a greater chance. *Rnase III* helps the anti-sense OOP RNA in destabilising *cII* mRNA. *HflD* directly inhibits the DNA binding by *CII* proteins. Leftward transcription may also be affected by *DnaA* and *SeqA*. *LexA* represses the OOP promoter, thus lower levels of *LexA* increases the chance for *cro* to outcompete for the operator region.

If *CRO* accumulation is high enough it inhibits *CI* and *CII* enough to prevent longer transcripts toward lysogenic conversion. While unhindered longer rightward transcription continues. This includes the *Q* gene and genes for proteins needed in viral replication. The *Q* protein accumulates in the cell allowing the RNA polymerase to continue its transcription from $P_{R'}$ promoter located in front of the *Q* gene. This extends transcription into the late genes downstream of it, which includes the transcription of structural proteins for virion assembly.

The late genes encode the proteins needed to complete the lytic infection including the head, tail, and lysis proteins (see section 1.4). The *R* and *S* proteins allow the release of the fully formed virions. The *S* protein puts a hole in the inner membrane giving cell wall access to the endolysin (*R* protein). The *R* protein degrades the peptidoglycan cell wall effectively lysing the cell and releasing the progeny phage. In lambda the lytic cycle take ~35 minutes, and culminates in the unleashing of ~100 virions.

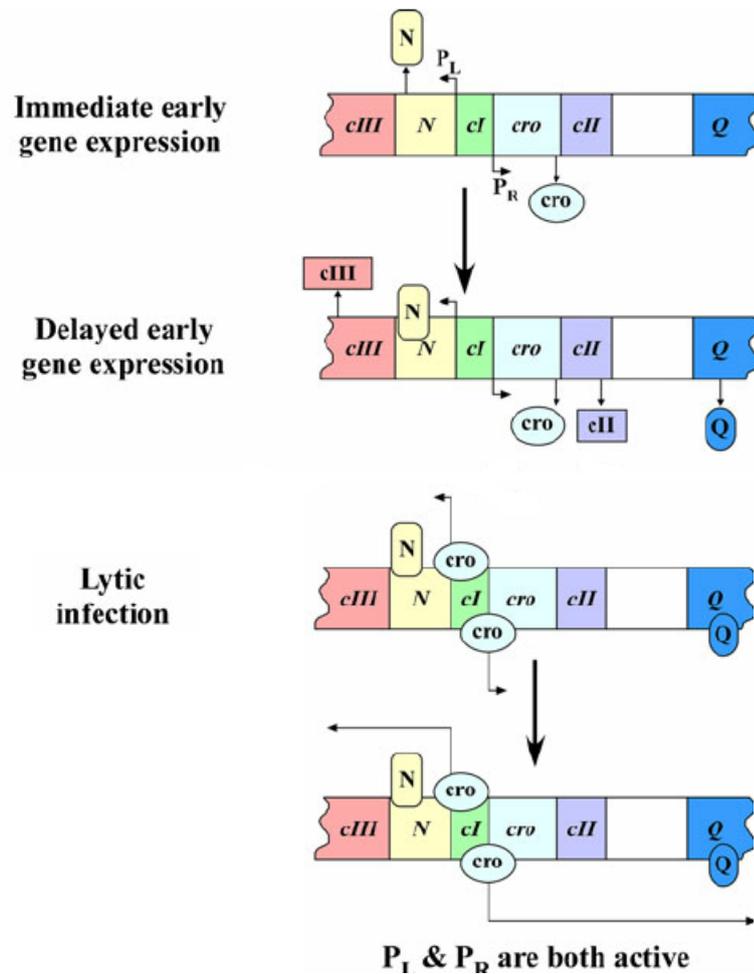


Figure 1.7 Basic core genes driving lytic infection of phage lambda. Edited from (Golais, Holly et al., 2013)

1.3.1.5 The temperate life cycle and induction

Only temperate viruses are capable of entering a lysogenic life cycle. The standard lysogenic cycle is the process in which a phage integrates into the host genome (prophage), and is replicated by the host with each cell division, until induction occurs. However some lysogenic phage can act as an independent circular DNA molecule that replicates in synchronisation with the host. Induction can occur spontaneously or through a stress based SOS response, which can be caused by a number of factors, DNA damage being the most efficient. Once induction is triggered, the virus enters the lytic cycle. Once entering the lytic cycle it forms progeny phage using host mechanisms and exits the cell via cell lysis. The primary upregulated genes associated to the

lysogenic cycle are CI, CII and CIII (for lysogeny transcription see Figure 1.8, for more general detail see Figure 1.5)

In the lambda model after immediate expression (see 1.4.1.3), lysogeny is dependent on CII outcompeting *cro*. IHF levels are key to lysogeny both in establishment (CI, CII and CIII expression) and eventual integration. If CII prevails, CI is produced, initially from the P_{RE} promoter and eventually CI activates the P_{RM} promoter ensuring a continuous supply of CI. CI outcompetes *cro* and commences the lysogenic gene cascade by binding to the operator region (see 1.3.1.3). Once CI has bound to each site within the operator region and formed the DNA loop, it becomes relatively self regulating. This leads to sufficient CI protein accumulation to activate the P_I promoter and thus Int gene transcription. At this point integration into the host genome begins.

Recombination of lambda DNA into the chromosome occurs at the attP site on lambda DNA and at the attB site in the bacterial chromosome. This recombination requires Integrase and IHF protein. Once in the chromosome, phage DNA is bounded by hybrid att sites, attL and attR. Phage excision from the chromosome requires Int, IHF, and Xis protein. Recombination always occurs at specific sites with specific enzymes, known as a site-specific recombination event. Without disrupting genes, the attB site on the chromosome lies between the gal and bio genes. Except for the continued production of CI from the P_{RM} promoter, the attB site is quiet. The expression of late genes is prevented by the lambda repressor, it binds to the operator sequences O_R and O_L, which blocks transcription from P_L and P_R. Long range and short range interaction involving the binding of CI to both O_L and O_R sites inhibits P_{RM} promoter (key to maintaining a lysogenic state) (Cui, Murchland et al., 2013).

Lysogeny can still be destabilised under the right conditions, i.e. spontaneous or chemical induction. The inhibition of P_{RE} by *cro* is essential in prophage induction due to its important prevention of CI resurgence (Schubert et al., 2007). O_{R3} enables CI to repress P_{RM}, maintaining a lysogenic level of CI that is low enough to be effectively removed by RecA upon induction of the SOS system. O_{R3} also prevents the recovery of CI levels that would otherwise impede or even halt lytic development after prophage establishment (Schubert et al., 2007). DNA damage triggers the SOS cellular response to deal with the damage. The RecA protein is activated in the SOS response

1.4 Bacteriophage structure and assembly

There are a variation of phage structures, with some commonalities across family and genera. The structural types include virion size, capsid shape and construct, tail presence/type, tail/terminal spikes, baseplate presence/type, and tail fiber presence/type. The phage of particular focus here-in is a lambda-like phage. Lambda-like phage typically have non-enveloped capsids, tails, baseplate and tail fibers, and use the injection method of infection. The mechanisms of lambda virion assembly (see Figure 1.9) are discussed within this section.

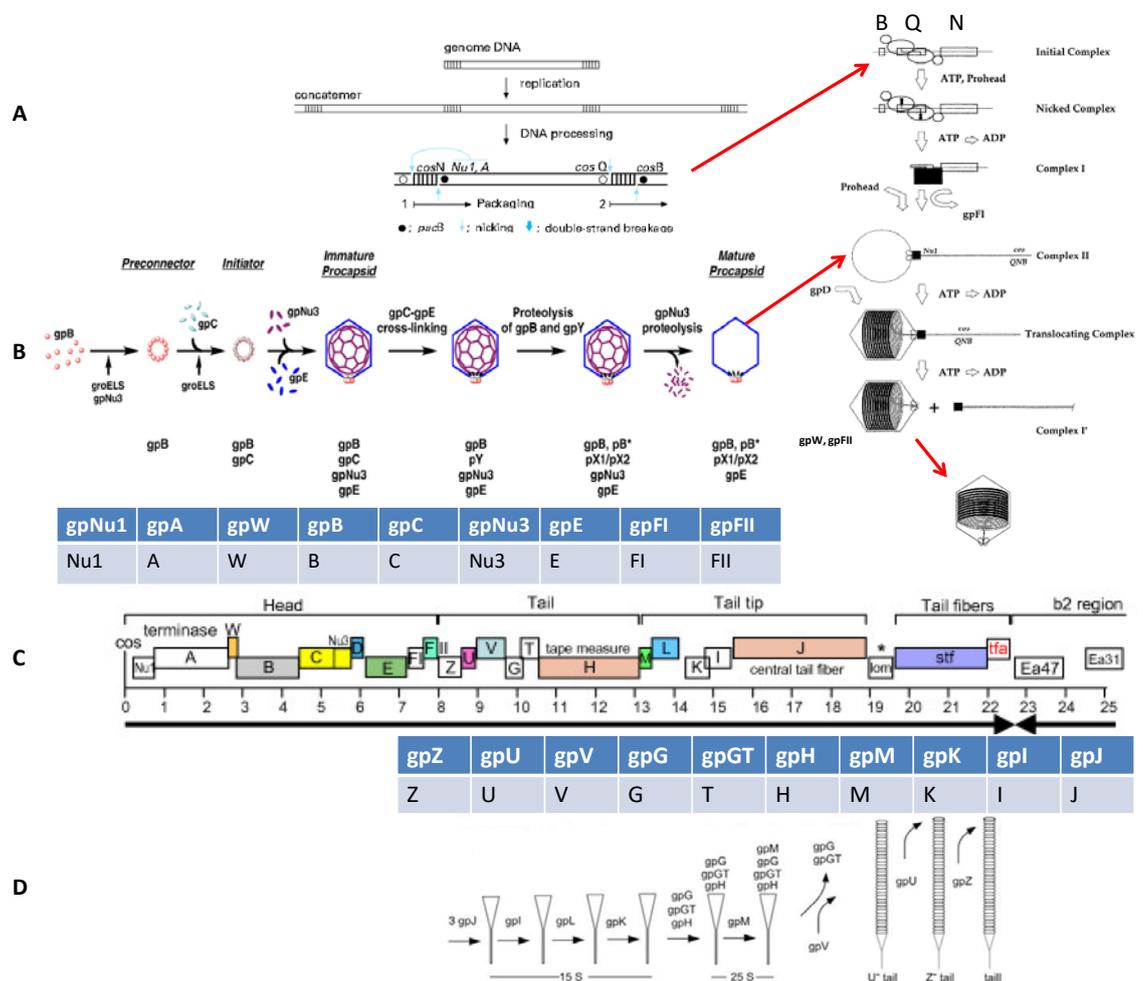


Figure 1.9 Lambda virion assembly. A: Concatemer formation, B: Capsid assembly and DNA packaging, C: Structural gene layout, D: tail assembly. Figure constructed using an accumulation of publications and UniProt (Duffy & Feiss, 2002, Fujisawa & Morita, 1997, Medina, Wiczorek et al., 2010, Rajagopala, Casjens et al., 2011, Xu, Hendrix et al., 2013)

1.4.1 Capsid and assembly

The primary function of the phage capsid is to encase the viral genetic material, a secondary function includes receptor recognition in the tail-less phages (Gowen, Bamford et al., 2003). Phage capsids can be either icosohedral or filamentous, with diameters ranging from ~43 nm to ~320 nm. The capsid has three main states; immature procapsid, mature procapsid, and nucleocapsid. Capsids are relatively complex protein structures, in the Lambda model the procapsid is formed from protein subunits arranged around scaffold proteins (gpNu3) (Caspar & Klug, 1962). The proteinous subunits consist of protomers and capsomers. Several oligomeric structural protein subunits form the protomers, the aggregation of these protomers make up the capsomers, which assemble to create the mature procapsid. Have published a detailed account of phage head assembly associated to the Hong Kong phage 97 (HK97) (Hendrix & Duda, 1998). In lambda phage the final capsid is made up of eight proteins, E, D, B, W, FII, B*, X1, and X2. Upon completion of the mature procapsid the scaffold proteins are ejected by protease C (gpC) and recycled. Capsid faces (Q) can consist of one or more proteins, the vertices/caps (T) are the points where 3 or more faces meet. This protective shell is unique to true viruses and distinguishes them from selfish genetic elements like transposons and plasmids (Krupovic & Bamford, 2009). In many phages, particularly lambda, the capsid forms the icosohedral shape after the first nucleic acids enter it, as they cause a change in conformation of the E protein. Some viruses have an additional structure encompassing the capsid, called an envelope, the envelope consists of glycoproteins and lipids derived from their host. Though relatively common in eukaryotic viruses and to some extent archaeal viruses, only 2 bacterial viral families (cystoviridae and plasmaviridae) have capsids with an envelope. The viral envelope is usually formed via 'budding', however envelope construction is poorly understood in phage, with the only known method of envelope assembly occurring in the cytoplasm.

1.4.2 DNA packaging: Mature procapsid to Nucleocapsid

The final stage of capsid formation is the packaging of the mature procapsid, which is instructed by the *cos* site. Based off the lambda model, this can be split into 3 main processes; Initiation of DNA packaging, translocation of DNA into the mature procapsid, and the termination of DNA packaging.

The genomic viral DNA is replicated to form a concatemer, at which point the initiation of DNA packaging starts (Catalano, Cue et al., 1995). The *cos* sites on the concatemer both initiate and terminate the DNA packaging of each virion (Catalano et al., 1995). The *cos* site is found in each replication of the genome and consists of three subsites, *cos*; N, B, and Q (Cue & Feiss, 2001, Wieczorek & Feiss, 2003). The *cos* subsites, *cos*N and *cos*B, are found on either ends of a genome replicate, and *cos*Q can be found just left of *cos*B. Packing machinery (IHF and terminase: gpNu1 and gpA) assemble at the *cos*N subsite forming the ‘initial complex’ (Catalano et al., 1995). Driven by terminase, nicks are introduced to the concatemer at the *cos*Q subsite forming the sticky ends of the mature virion DNA (Wieczorek & Feiss, 2003). The *cos*B subsite aids in the accuracy and efficiency of the concatemer nicking (Hang, Catalano et al., 2001), and introduces an intrinsic bend into the concatemer. This forms the ‘nicked complex’, the terminase then separates the sticky ends via ATP hydrolysis forming ‘Complex I’. Interaction between the gpA protein on terminase end of ‘complex I’ and the capsid portal protein gpB, helps the docking of complex I to the mature procapsid portal vertex, forming ‘complex II’ (Yeo & Feiss, 1995).

The translocation of the DNA into the mature procapsid is then carried out via further ATP hydrolysis until the next *cos* site on the concatemer reaches the packaging complex (Hwang & Feiss, 1995, Rubinchik, Parris et al., 1994, Yang & Catalano, 2003). At this point termination occurs with the terminase cutting the downstream *cos*, where *cos*Q works in concert with *cos*N to promote efficient termination (Cue & Feiss, 1998). Termination causes the undocking of the remaining concatemer from the DNA filled capsid (nucleocapsid), the terminase undocks with and remains bound to the concatemer, to continue the DNA packaging of mature procapsids. gpW and gpFII are added to the capsid portal, where it is presumed that gpW and gpFII prevent the DNA loss from the filled capsid (Perucchetti, Parris et al., 1988).

1.4.3 Tail fibres

Tailed phages constitute the *Caudovirales* order, phage tails and their associated tail type are; long contractile (*Myoviridae*), long flexible non-contractile (*Siphoviridae*), short non-contractile (*Podoviridae*). The primary function of phage tail fibres is adsorption to the host cell, serving as a conduit for genome infection. In the lambda model, the tail is 140 nm long and encoded by an operon of 11 genes Z, U, V, G, T, H, M, L, K, I and J (Casjens & Hendrix, 1974; Xu et al, 2004)

1.5 Phage/host co-evolution

Ultimately phage is incapable of entirely eliminating its host without negatively affecting its own survival. Phage survival can be improved through regulation of their lytic behaviour, broadening of their host range, and encoding positively selected traits. In natural selection only the most successful mutations or conveyed adaptations are retained within the phage gene pool, and the more generalised beneficial changes can ultimately become co-selected by the bacterial host.

There are many examples of evolutionary changes between phage and bacteria in their ongoing struggle for survival. Potential bacterial hosts can avoid infection by changing receptors associated to phage infection (Bohannan & Lenski, 2000). Phages can counter this through domain exchange of long tail fibre adhesin that govern recognition specificity (Tetart, Desplats et al., 1998). Bacteria can use defence mechanisms that include the translation of restriction enzymes, which prevent integration of foreign DNA (Handa & Kobayashi, 2005) by cleaving and degrading phage DNA. Prevalent phages counter this response through evolutionary alteration at the nucleotide level (Labrie, Samson et al., 2010).

As well as encoding new genetic traits to the host bacterium, phage can also adopt host beneficial characteristics. For example, a phage maintaining a bacterium's key mechanisms in order to supply its own need (i.e. replication and lysis), can be observed by certain cyanophages. SM-17 cyanophage is genetically related to T4 and T7 phages. This phage helps maintain host photosynthetic activity upon integration by providing alternative routes of carbon metabolism during infection. This trait is used by cyanobacteria to produce enough energy for the converting phage to utilise for its replication (Sullivan, Coleman et al., 2005).

1.5.1 Phage encoded protection

A phage can encode protection for its host against environmental stress (Colomer-Lluch, Jofre et al., 2011, McGrath, Seegers et al., 1999), most likely enabled from its *bor* gene (Bik, Bunschoten et al., 1995), via lysogenic conversion (horizontal gene transfer (HGT)) (Vostrov, Vostrukhina et al., 1996). This can allow the bacterial host to incorporate resistance genes against antimicrobials, benefiting the host's survival whilst phage is integrated. Attributable to this, is the likelihood that the trait will be selected for and conserved in future phage.

Phage encoded resistance could classify as symbiosis between host and virus, temporarily establishing not only pathogenic but also superior bacteria, in terms of affiliation with other microbes within a niche. These symbiotic relationships are also capable of taking place over more than just two organisms, for example phage WO-B, *Wolbachia pipientis*, and arthropods have been shown to form a tripartite symbiotic association in which all three are integral to understanding the biology of their widespread endosymbiosis (Chafee, Zecher et al., 2011).

Phage may have evolved encoded protection mechanisms due to their aptitude in carrying ecologically important traits, such as defence against parasitoids or toxigenic substances, within and amongst symbiont and animal host lineages (Oliver, Degnan et al., 2009). Symbiosis can provide beneficial attributes to host bacteria, aiding in outmatching competitors or sanctioning temporary survival in new niches previously hostile/uninhabitable by other members of its species. These traits occur via natural selection of favourable non-silent mutations, promoting co-operative traits between bacteriophages and bacterial strains.

Although the phage/host interaction can be mutually beneficial, ultimately the purpose of phage infection is the exploitation of bacterial mechanisms of replication, to promote self-growth. A phage that encodes advantageous characteristics provides an indirect improvement of self-survival, furthermore, by establishing a more stable environment, additional exasperation of cellular mechanics can be exploited. This innate selfish gene behaviour is epitomised in temperate phage's 'trigger' like switch, which allows it to lyse ('bail from') the host cell when survival appears bleak. Interestingly some phage can lose their lytic capability whilst maintained as a

prophage within the host (cryptic phage), likely driven by an evolved bacterial mechanism to maintain beneficial genetic elements.

Stx-phages transfer Stx, as well as other properties, by HGT, this can promote the evolution of new Stx-producing *E.coli* variants. These variants, along with other emerging bacterial producers of Stx, suggest increases in Stx-phage hosts. The phage of investigation within this thesis is Stx Φ 24_B. The Stx Φ 24_B has an integrase with a broad host range (James, Stanley et al., 2001), and an augmented virulence in terms of its ability in distribution (Figure 1.10). Integration of phage DNA into the bacterial genome is dependent on the *int* gene. Although Φ 24_B is a lambdoid like phage it has some functions that work in an inverted orientation, such as a promoter and *int* gene (Fogg, Rigden et al., 2011). As a result Φ 24_B is not under the control of CI's indirect repression of *int* transcription from pL promoter during lysogeny (as in lambda), and the novel excisionase ORF which would be under the support of CI via pL (Fogg et al., 2011). Instead Φ 24_B's pInt promoter to *int* means its capable of lysogen production at a greater rate, as the integrase is separate to global phage repression control, orientation and lambdoid genetic regulatory strategy leaves *xis* coupled to CI repressor control, maintaining prophage stability (Fogg et al., 2011). A reason for investigating Stx Φ 24_B within this thesis is due to its broad host range, and its inverted *int* gene may impose naturally higher interest for phage encoded resistance, due to increased prophage stabilisation (see Figure 1.10).

There are many bacterial characteristics that can be related to bacteriophages, the most historically known being exotoxin production causing bacterial virulence (Casas, Magbanua et al., 2010). Bacteriophage modification of bacterial characteristics also includes: adhesion, colonisation, resistance to immune defences, toxin production, invasion, spread through human tissues, sensitivity to antibiotics, and transmissibility among humans (Wagner & Waldor, 2002). Some associated mechanisms behind this can be linked to phage Lom and bor proteins, the *Lom* and *bor* genes encode nonessential characteristics, which aid in the growth of the phages lysogenised bacterial host (Bik et al., 1995). Specifically *lom* provides enhanced adhesion to mammalian cells, whereas *bor* increases serum resistance (Barondess & Beckwith, 1990, Barondess & Beckwith, 1995).

As there are many genes studding the backbone of the genomes of phage with no known function, it makes it difficult to identify genetic nuances or positive selection traits that these viral entities horizontally transduce. Due to the broad host range and promiscuous nature of these phages high levels of recombination can occur between either infecting phages, between the phage and the host, or between the infecting phage and remnant prophage harboured in the bacterial chromosome meaning that recombination can be rapid.

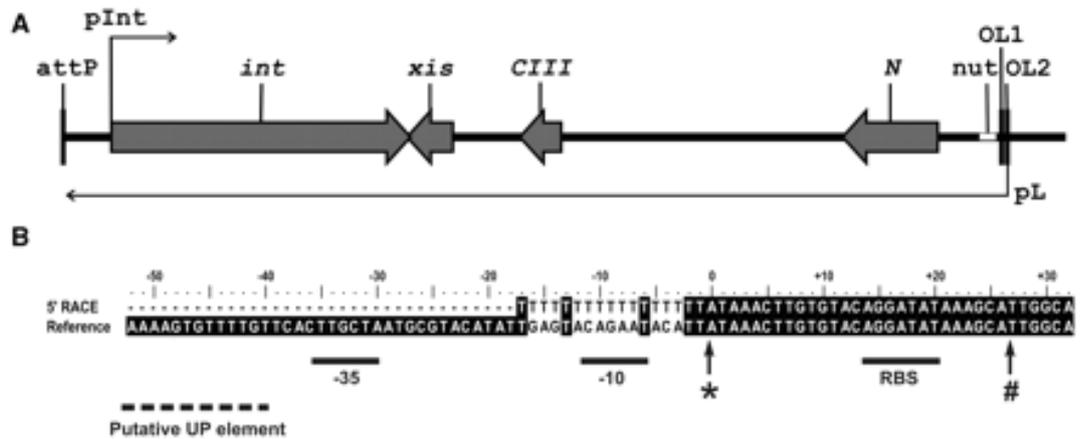


Figure 1.10 Integrase region of $\Phi 24_B$, its Map, and its transcription. A: Schematic map of the $\Phi 24_B$ integrase region. $\Phi 24_B$. B: Schematic of the $\Phi 24_B$ integrase transcription start site and predicted promoter, determined by 5'-rapid amplification of cDNA ends (RACE) and *in silico* analysis. Poly-T start to the 5'-RACE is an artifact of the amplification process; $\Phi 24_B$ genome sequence (HM_208303) utilised for the alignment. Putative -10/-35 promoter constituents and the distal portion of a putative UP-element are indicated by labeled solid lines and a dashed line, respectively. Start of transcription is represented by an asterisks (*), the new putative translational start site by a hash (#) and ribosomal binding site by RBS (Fogg et al., 2011).

1.5.2 Toxin/anti-toxin mechanisms

First discovered in a plasmid present in *E. coli* (Ogura & Hiraga, 1983) the bacterial toxin/anti-toxin (TA) system has several hypothesised functions. Many TA systems are linked to the phage-host arms race, the principle of these system is the bacterial evasion of phage infection. Three major biological functions of TA modules have been discovered, post-segregational killing (“plasmid addiction”), abortive infection (bacteriophage immunity through altruistic suicide), and persister formation (antibiotic tolerance through dormancy). There are several hypothesised functions of the TA system, which include; stabilisation of genomic parasite, gene regulation, junk DNA, growth control, selfish alleles, antiphage, persisters, programmed cell arrest and preservation, and programmed cell death. Functions most relevant to bacteriophage are the stabilisation of the genomic parasite and antiphage functions.

There is an innate relationship between bacteriophage infection and the bacterial TA systems, as it can be an effective bacterial defense mechanism. Phage that trigger the TA system of its host can result in limited or arrested phage translation, replication and proliferation. Bacterial TA mechanisms are key in some effective defensive strategies such as abortive techniques. Abortive techniques include both lytic and bacteriostatic mechanisms. The bacterium *Lactococcus lactis* uses an abortive infection (Abi) mechanism, which is TA system mediated (Fineran, Blower et al., 2009). Over 20 Abi systems have been identified in *Lactococcus lactis* (AbiA to AbiZ) (Chopin, Chopin et al., 2005). The AbiE system encoded by bicistronic operons and functions via a non-interacting bacteriostatic TA mechanism (Type IV), preventing phage proliferation (Dy, Przybilski et al., 2014). AbiP disrupts phage replication and the temporal switch from early to late gene expression (Domingues, Chopin et al., 2004). In phage infected cells premature lysis is induced by AbiZ, thereby preventing complete viral assembly and progeny phage release.

However in the on going phage-host evolutionary arms race, phage mechanisms have developed to evade such TA systems. The T4 phage *pinA* gene which blocks *E. coli* Lon proteases, is one such example of a phage based TA evasion mechanisms (Christensen, Maenhaut-Michel et al., 2004, Skorupski, Tomaschewski et al., 1988).

There are 5 main types of TA systems (type I – type V), see Figure 1.11. Type I antitoxins are unstable antisense sRNAs, which generally functions by destabilising the mRNA of the toxin via basepairing, preventing ribosomal binding, thereby stopping downstream translation (Brantl, 2012, Fozo, Hemm et al., 2008). Examples of TA systems regulated by inhibition of toxin mRNA translation include; *tisB/istR-1* (Wagner & Unoson, 2012), *ibs/sib* (Fozo, 2012), and *symR/symE* (Kawano, Aravind et al., 2007). The *symR/symE* is an *E. coli* module of this TA system, regulation of which is managed by LexA and Lon (Fernandez De Henestrosa, Ogi et al., 2000, Kawano et al., 2007). LexA controls *symE* expression and Lon protease degrades SymE. LexA is an SOS-response regulated transcriptional repressor, previously discussed for its role in phage induction.

Type II TA systems are protein interactions, the antitoxin protein neutralises the toxin protein by forming a protein-protein complex with the toxin. In the type II models the toxin protein is stable, but the antitoxin is degraded by the Lon (Christensen, Mikkelsen et al., 2001, Roberts, Strom et al., 1994, Smith & Rawlings, 1998, VanMelderen, Thi et al., 1996) or Clp family (Aizenman, Engelberg-Kulka et al., 1996, Cherny & Gazit, 2004, Diago-Navarro, Hernandez-Arriaga et al., 2013, Lehnerr & Yarmolinsky, 1995) proteases. Type II TA systems usually code for the antitoxin protein first, as the operons normally consist of two small open reading frames, where the antitoxin gene lays upstream, down-regulating toxin expression. An example of a type II TA system is the F plasmid of *E. coli*. Type II systems vary in the way they exhibit toxicity, the toxin in this example CcdB protein targets DNA gyrase (Bernard & Couturier, 1992).

Type III systems can be assigned to 3 families *toxIN*, *cptIN*, and *tenpIN* (Blower, Short et al., 2012), where most are coded for by bacterial chromosomes, second most by plasmids, and just one encoded by a prophage (*toxIN*). Type III module antitoxins are sRNA. *Pectobacterium carotovorum* is an example of this type system, where it aids in prevention of phage infection (Blower et al., 2012, Fineran et al., 2009). In the *toxI/toxN* TA module of plasmid pECA1039 from *P. carotovorum*, there is an inverted repeat (TA terminator) and tandem array of direct repeats upstream of the *toxN* gene. This TA terminator regulates both the toxin mRNA (ToxN) and antitoxin sRNA (ToxI), while the direct repeats act as a release site for the RNA antitoxin. The

RNA antitoxin release is performed by the RNase activity of the ToxN protein which cleaves the *toxI/toxN* transcript

Type IV TA systems, unlike other TA systems, use toxins and anti-toxins that do not directly interact. Examples of type IV systems include the *E. coli* *ctbA/ctbB* TA and the *E. coli* *cptA/cptB* (Masuda, Tan et al., 2012). In the *ctbA/ctbB* TA system, the *ctbA* toxin interferes with the polymerisation of MreB and FtsZ (bacterial cytoskeleton proteins), which inhibits cytoskeleton assembly. The *ctbB* anti-toxin protein stabilises the *ctbB* toxin effects by interacting with MreB and FtsZ polymers and enhancing the bundling of their filamentous polymers (Masuda et al., 2012).

Type V TA systems are a protein-mRNA interaction, an example being the *E. coli* *ghoS/ghoT* TA module (Wang, Lord et al., 2012). The GhoT toxin is a membrane lytic peptide that causes lysed cells with damaged membranes and increases persister cells that are more tolerant to antibiotics. The GhoS anti-toxin is able to prevent GhoT translation, as it has a sequence specific endoribonuclease activity that cleaves the GhoT toxins mRNA (Wang et al., 2012).

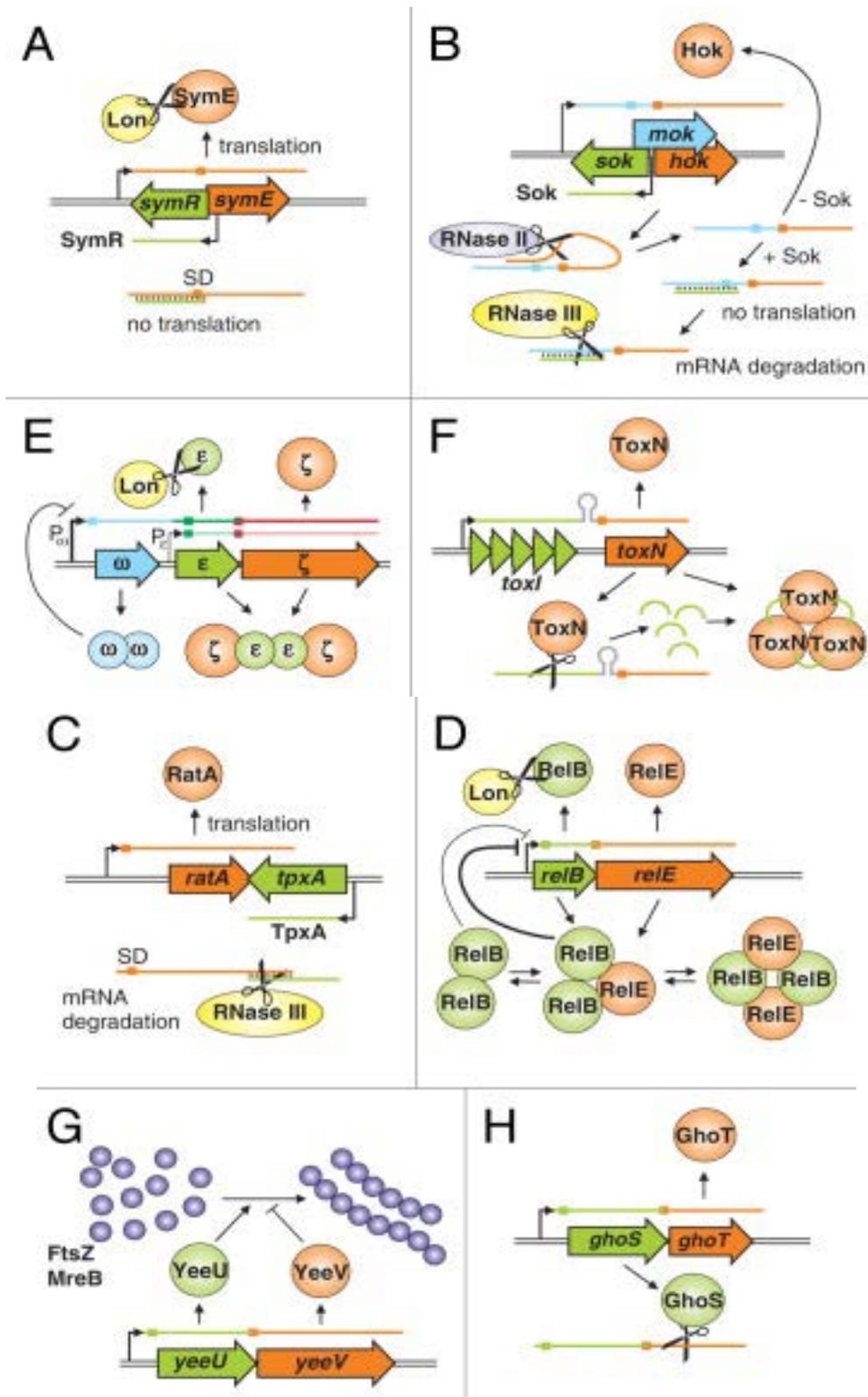


Figure 1.11 Types of TA systems. (A) Type I system regulated by interference of toxin mRNA translation, example; *symR/symE* module of *E. coli*. SD, Shine-Dalgarno sequence. (B) Regulation of the type I system *hok/sok* of plasmid R1. (C) The *ratA/tpxA* module from *Bacillus subtilis* represents a type I system where toxin mRNA degradation is promoted. (D) The *relB/relE* two module type II system from *E. coli*. (E) The ω - ϵ - ζ three module type II systems from *Streptococcus pyogenes* plasmid pSM19035. (F) The *toxI/N* type III system from the *Erwinia carotovora* plasmid pECA1039. (G) The *yeeU/yeeV* type IV system of *E. coli*. (H) The *ghoS/ghoT* type V system of *E. coli*. Toxin and its encoding gene are shown in orange, and antitoxin and its encoding gene are shown in green. Edited from (Unterholzner, Poppenberger et al., 2013)

1.5.3 Clustered regularly interspaced short palindromic repeats (CRISPR)/CRISPR-Associated-protein (CAS)

CRISPR are arrays of bacterial DNA sequences that provide a form of acquired/adaptive phage immunity. The CRISPR locus contains short non-coding leader sequence (casgenes and spacers of foreign DNA) incorporated between CRISPR repeats. First characterised by Francisco Mojica and Ruud Jansen, the CRISPR locus spacer sequences were not identified as complementary to bacteriophage until 2005 (Mojica et al, 2005). CRISPR can be compared to the mammal immune system in that its spacers represent a sort of memory of past exposures of bacteria, phages or plasmids (Marraffini & Sontheimer, 2010). Of all the analysed strains, the CRISPR locus has been detected, to some degree, in 87% of archaea and 70% of bacteria, demonstrating that this is a core mechanism of phage resistance in the bacterial kingdom (POURCEL, 2017).

The CRISPR/CAS system is a two stage system, CRISPR and the CRISPR-associated protein (CAS), see Figure 1.12. The first stage (CAS), involves adaptation, where the bacterial cell gains a new spacer from foreign DNA, thereby establishing recognition for immunity. The second stage (CRISPR), is the active immunity stage, where foreign DNA is targeted by acquired spacers.

When foreign DNA enters the cell a section is cleaved into protospacers by the nuclease protein (CAS), these protospacers are inserted into the CRISPR locus near the leader sequence. The CRISPR locus is transcribed into a single precursor RNA (pre-crRNA). The pre-crRNA is then cleaved by ribonucleases (CAS and/or an alternative bacterial protein) into individual CRISPR RNA (crRNA) units containing one targeting spacer (mature crRNA). The mature crRNA forms a complex with the Cas proteins, this crRNA-Cas complex recognises foreign DNA, cleaving it at complementary sites to the crRNA's protospacers.

In keeping with the phage-host arms race, bacteriophage have evolved anti-CRISPR systems. These systems circumvent CRISPR/Cas via methods that include the phosphorylation of the Cas proteins and the mutation, deletion or recombination of the targeted spacer sequence. Some phage mechanisms actually use CRISPR sequences in lysogenisation, lysogeny maintenance, and prophage induction.

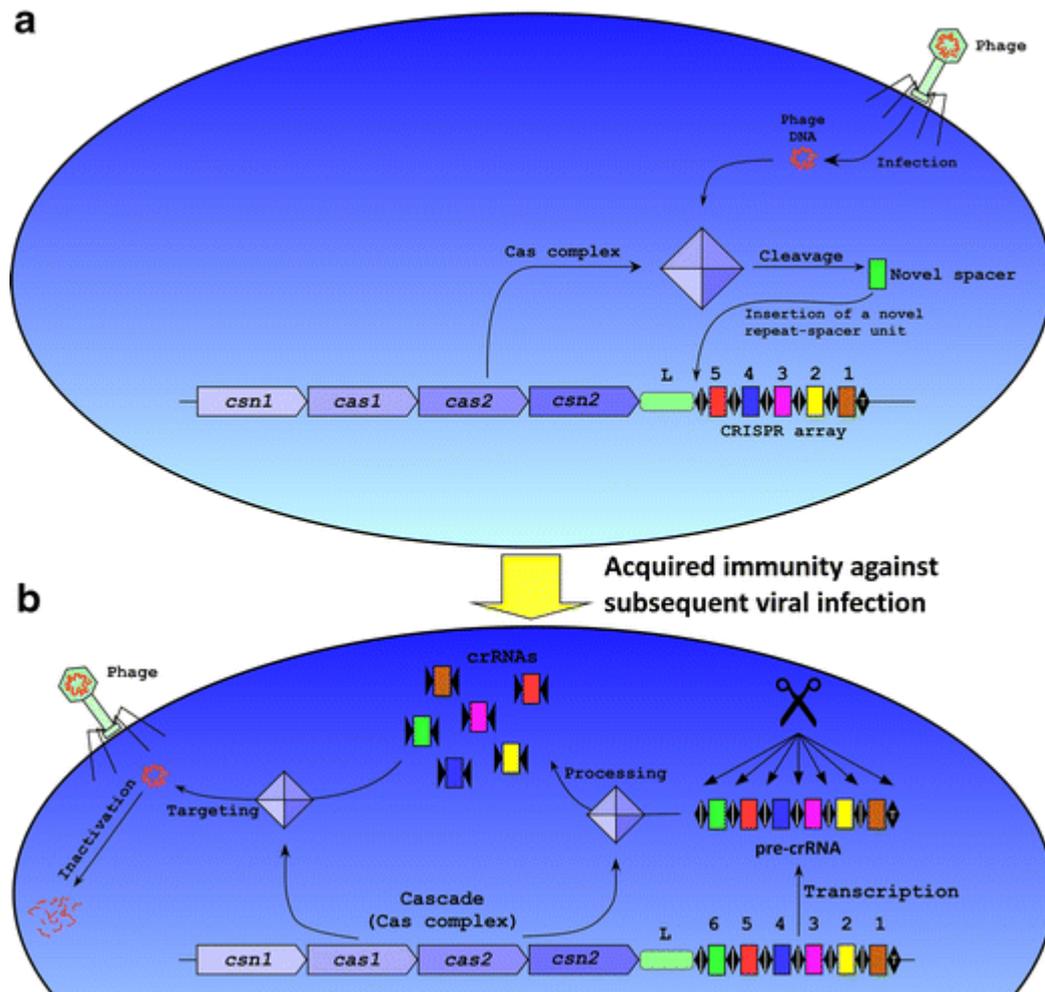


Figure 1.12 CRISPR/Cas mechanism of action. **a** Establishment of immunity. Cas complex recognizes foreign phage DNA, generates a novel repeat (*rectangle*)-spacer (*diamond*) unit, and integrates it at the leader (*L*) end of the CRISPR locus. **b** The process of immunity. The CRISPR repeat-spacer array is transcribed into a pre-crRNA that is processed into mature crRNAs, which then interfere with the corresponding invading phage nucleic acid.

1.6 Human gut microbiota

The human gut microbiota is a dense microbial community consisting of bacteria, viruses and fungi. The gastrointestinal biome starts its naïve form in foetal life and develops rapidly after birth (Collado, Rautava et al., 2016). Early microbial communities and seeding depends on a range of variables that include mode of birth (Neu & Rushing, 2011), birth place (van Nimwegen, Penders et al., 2011), gender (Cong, Xu et al., 2016), and feed (Cong et al., 2016). The microbiota of the gut continues to develop through infancy and into adulthood (see Figure 1.13).

Through developments in DNA sequencing technologies over the last decade, there has been an increase in studies targeting each part of the microbiome (see Figure 1.14). The gastrointestinal microbiome is one of the most studied mammalian biomes (see Figure 1.14), with over 1000 bacterial species now identified, though most are yet to be cultured (Rajilic-Stojanovic & de Vos, 2014). Collectively the gut microbiota has a genome of approximately 150 times that of the human genome, with an estimated ~3.3 million genes (Zhang, Raof et al., 2010), the distribution of bacterial load and community structure through the gastrointestinal tract can be seen in Figure 1.15. There is currently not enough data to plot the gut phageomes community structure, with research thus far elucidating to its richness and diversity, as the taxonomy is mostly unclassified (Manrique, Bolduc et al., 2016). Increased analytical depth, both at a genetic or protein level, has improved our understanding of the gut microbiome role and function, leading to potential methods in manipulating, managing, and treating the intestinal microbiota. Current approaches to treatment include; prebiotics (Gibson & Roberfroid, 1995), probiotics (Fuller, 1989), and faecal transplantation (Landy, Al-Hassi et al., 2011). Increased understanding of the gut microbiome combined with the alarming rise in antibiotic resistance, has raised concern over the use of antibiotics (Carlet, 2012, Francino, 2015), and driven research interest into target specific treatment of infection (phage therapy). The importance and focus toward comprehending the complexity of the gut biome, particularly the viral and bacterial interplay has never been greater.

A given individuals gut biome is relatively unique (Schloissnig, Arumugam et al., 2013), but generalities of healthy and unhealthy guts can be used to identify traits in disease and its progression. Health and some clinical diseases have been linked to changes in the microbiota, this

dysbiosis has been implicated in several diseases, including colon cancer (Ahn, Sinha et al., 2013, Feng, Liang et al., 2015), obesity (Greenblum, Turnbaugh et al., 2012), and autoimmune disorders (Colpitts & Kasper, 2017, Dopkins, Nagarkatti et al., 2018). As well as being indicators of disease, the gut flora can have a significant impact on every day health. The gut microbiome has been linked to; immune development (Macpherson, de Agüero et al., 2017), pathogen inhibition (Kamada, Chen et al., 2013), and nutrient digestion and uptake (Flint, Scott et al., 2012b).

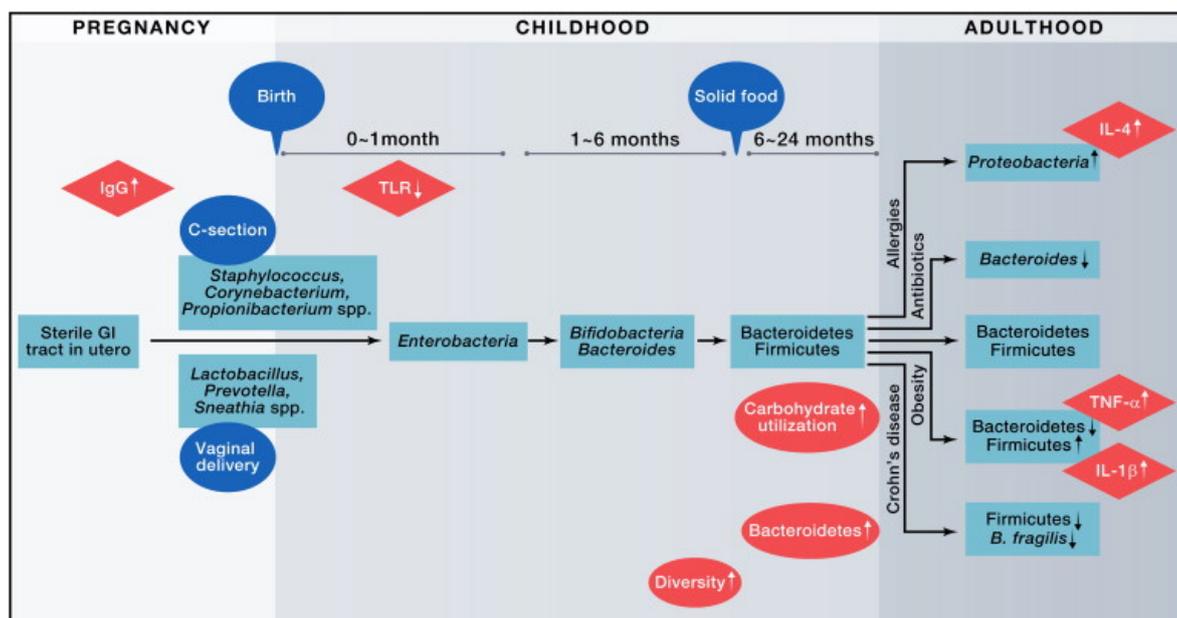


Figure 1.13 Development of the Microbiota (Clemente, Ursell et al., 2012). The initial communities vary depending on mode of delivery, i.e. a skin-like or vaginal-like configuration. During the first weeks of life, there is a reduced activity of toll-like receptors (TLRs), perhaps promoting a stable microbial community. As the infant grows, and with the introduction of solid foods, the microbiota diversity increases, and the community converges toward an adult-like state. At the same time, the immune system “learns” to differentiate between commensal and pathogenic bacteria. By adulthood, a relatively stable community composition (but varying between different individuals) is achieved, dominated mostly by Bacteroidetes and Firmicutes. Different diseases are characterized by significant changes in the microbiota and associated changes in the production of cytokines.

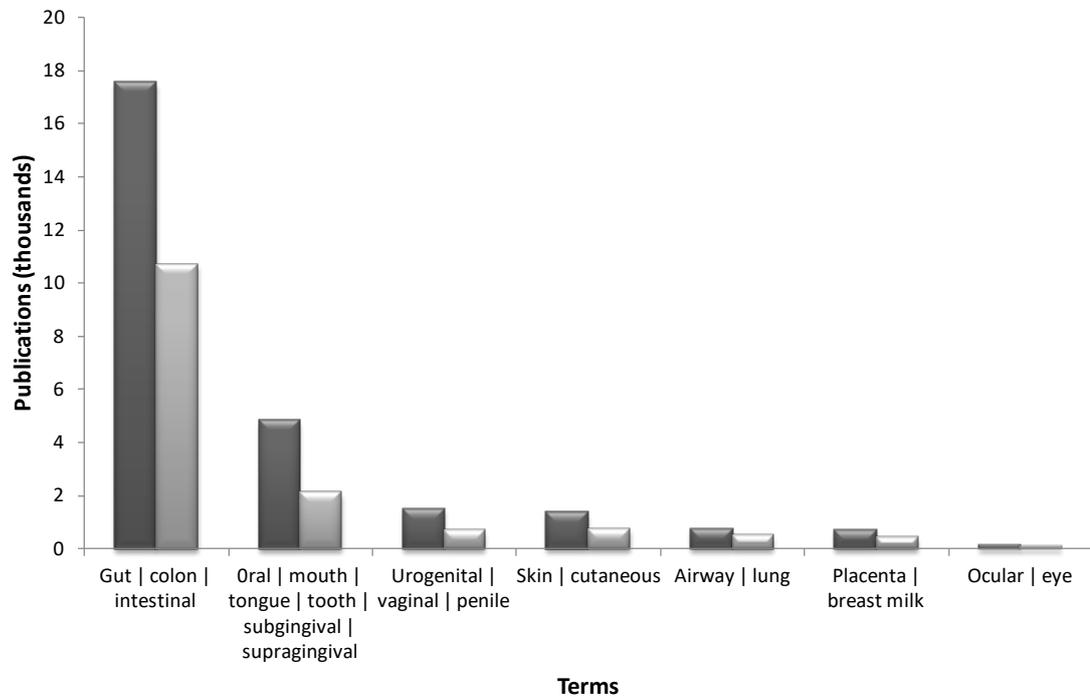


Figure 1.14 Diversity of recent microbiome research. Column graph plotted from the tabulated data in Lloyd-Price, J. *et al*, 2016 (Lloyd-Price, Abu-Ali et al., 2016). It presents the number of results obtained by searching for “(microbiome | microbiota | microflora) (<Terms>)” on PubMed (retrieved 31 March 2016). Dark grey: All publications, light grey: publication from 2011 to 2016.

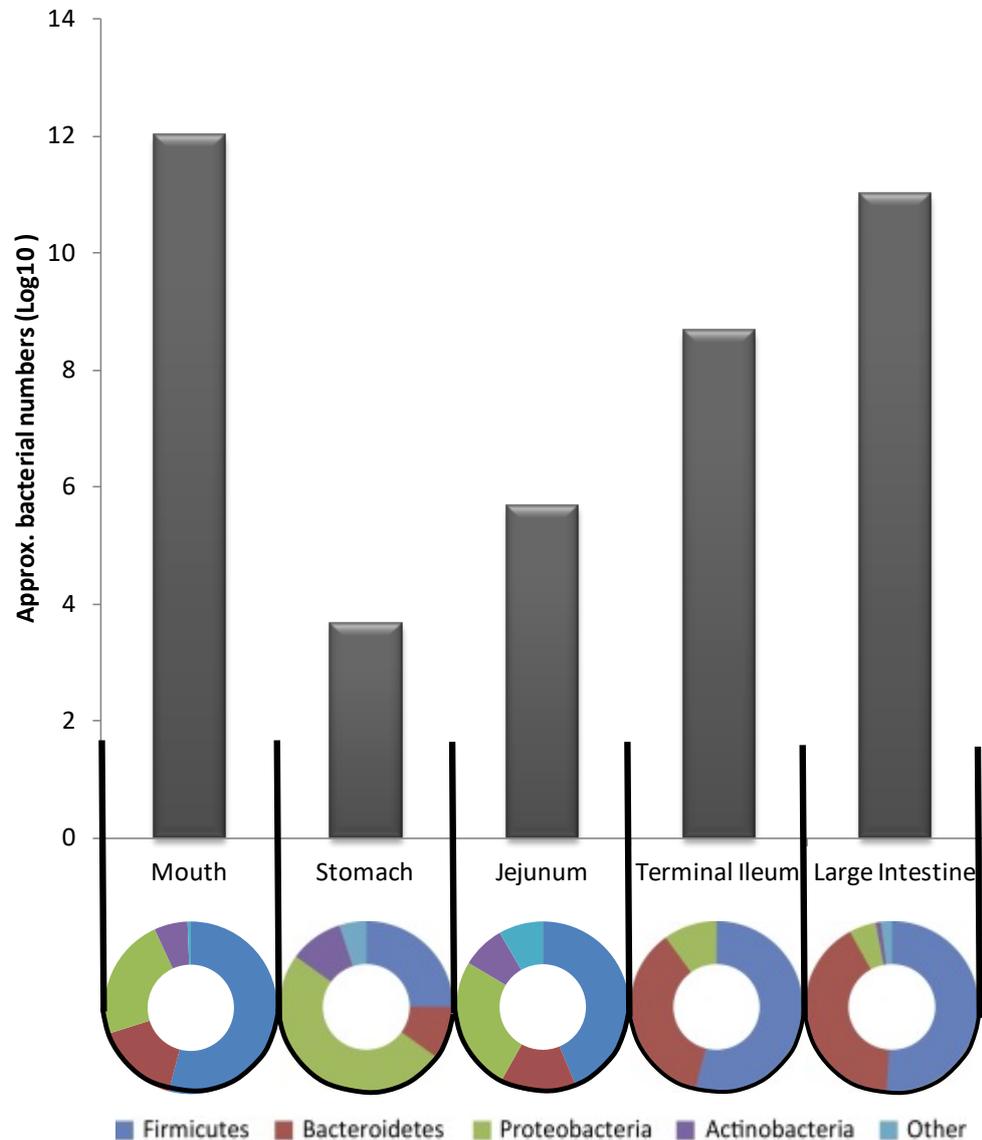


Figure 1.15 The taxonomic diversity and bacterial load through the gastrointestinal tract. Column chart plotted using the tabulated data by Baothman, O. A. *et al.* 2016 (Baothman, Zamzami *et al.*, 2016). The column chart illustrates the number of bacteria in different components of the gastrointestinal tract, note; large intestine is 'Per gram of intestinal contents'. The Doughnut plots show taxonomic diversity through the gastrointestinal tract, presented at phylum level. Doughnut plots of the stomach, ileum and large intestine were constructed from data generated by Derrien, Muriel. *et al.* 2015 (Derrien & Vlieg, 2015), from Bik, Eckburg *et al.*, 2006, Zoetendal, Raes *et al.*, 2012 and Claesson, Cusack *et al.*, 2011 respectively (Bik, Eckburg *et al.*, 2006, Claesson, Cusack *et al.*, 2011, Zoetendal, Raes *et al.*, 2012). The phylum diversity of the mouth (oral) was calculated from the combined data of Oh, C. *et al.* 2015 (Oh, Lee *et al.*, 2015) and Guerrero-Preston, R. *et al.* 2016 (Guerrero-Preston, Godoy-Vitorino *et al.*, 2016), which was averaged and plotted as a doughnut chart. Doughnut chart of the jejunum phylum diversity was plotted from the tabulated data in Sundin, Olof H. *et al.* 2017 (Sundin, Mendoza-Ladd *et al.*, 2017).

1.6.1 The gut flora and metabolic function

The gut flora plays an essential role in our ability to metabolise food, both its breakdown and nutrient adsorption (Krajmalnik-Brown, Ilhan et al., 2012). Humans have an inability to produce enzymes in the gut that break down; polysaccharides, polyphenols and aid synthesis of vitamins. Furthermore bacteria can help in the breakdown of undigested foods such as carbohydrates, proteins, and vitamins, improving their bioavailability (Laparra & Sanz, 2010, Possemiers, Bolca et al., 2011). Bacteria can often rescue or scavenge resources that would otherwise be lost including dietary particles that often fail to be fully digested upon reaching the large intestine. The bacteria present in the large intestine help breakdown these undigested dietary substrates (Flint, Scott et al., 2012a, Macfarlane, Gibson et al., 1992). The gut microbiome has been shown to have a systemic regulatory role of bile acids, allowing influence of the mammalian metabolic status (Ghazalpour, Cespedes et al., 2016). This altering of bile acids is one mechanism in which the microbiome affects our metabolism, but the microbiome also plays a more direct role in metabolism in the gut (Nieuwdorp, Gilijamse et al., 2014).

Saccharolytic bacterial fermentation is the process in which bacteria breakdown carbohydrates, the key products of which are short chain fatty acids and gases (Macfarlane & Macfarlane, 2012, Miller & Wolin, 1979). Saccharolytic bacterial fermentation occurs predominantly in the proximal colon (Macfarlane & Macfarlane, 1993, Macfarlane & Macfarlane, 2003), and saves energy and time in human digestion of carbohydrates whilst producing beneficial metabolites. The most abundant short chain fatty acids are acetate, butyrate, and propionic acid. These fatty acids support a range of beneficial functions and cells such as human colonocytes (Roediger, 1980, Schauber, Svanholm et al., 2003), apoptosis of colon cancer cells (Hague, Elder et al., 1995, Jan, Belzacq et al., 2002), anti-cancer activity via gene expression regulation using histone deacetylase inhibition (Marks & Xu, 2009, Waldecker, Kautenburger et al., 2008), potential activation of intestinal gluconeogenesis benefiting glucose and energy homeostasis (De Vadder, Kovatcheva-Datchary et al., 2014), satiety signalling/appetite regulation (Chambers, Viardot et al., 2015, Frost, Sleeth et al., 2014, Karaki, Tazoe et al., 2008), cholesterol metabolism (Hara, Haga et al., 1999), and lipogenesis (den Besten, Bleeker et al., 2015, Moreau & He, 2017).

The microbiota also aids in protein metabolism, converting both ingested and endogenous protein into short and branched-chain fatty acids, shorter peptides, amino acids and derivatives, and gases (Portune, Beaumont et al., 2016). Bacterial proteolysis predominantly occurs in the distal colon, where they play a considerable role in amino-acid metabolism and bioavailability in the gut (Gill, Pop et al., 2006).

There are many vitamins that can be synthesised by the gut microbiota, notably vitamin B, and K group vitamins including biotin, pyridoxine, folates, nicotinic acid, cobalamin, pantothenic acid, thiamine, and riboflavin (LeBlanc, Milani et al., 2013). The synthesised vitamins help supplement mammalian health, and without them the host can become significantly deficient without sufficient dietary supplementation (Ikeda, Hosotani et al., 1979, Sumi, Miyakawa et al., 1977, Wostmann, 1981, Wostmann & Knight, 1965). Such deficiencies have been shown to result in haemorrhages, anaemia, and neurological disorders (Ahmad, Mirza et al., 2013).

The bioavailability and impact of polyphenols (acquired mostly from fruits and vegetables) greatly depends on the microbiota for metabolism (Ozdal, Sela et al., 2016). This process is less direct than other processes and often requires a broad range of microbes and cross-feeding between them (Ozdal et al., 2016). Most polyphenols in the human diet are glycosides, other polyphenols include proanthocyanidins and ellagitannins. An example of cross-feeding can be described in glucoside (glycoside derived from glucose) metabolism. Hydrolysis of polyphenols like glucoside forms alkycones, which is required to improve bioavailability. Although intestinal mucosal enzymes can catalyse some glucoside hydrolysis, the majority passes into the colon, where it is hydrolysed by the microbiota (Kuhnau, 1976). The resulting alkycones are cross-fed to, and further metabolised by, other bacteria within the microbiota, ultimately generating simpler phenolic compounds that can be easily absorbed (Duda-Chodak, Tarko et al., 2015).

1.6.2 The gut flora and immune system

The gut microbiota is unique in each individual at a genus and species level, but it is generally conserved at the phylum level, populated mostly by Firmicutes and Bacteroides, and secondly by Proteobacteria and Actinobacteria. The gut microbiota plays an important role in the development of immunity to pathogenic bacteria, as such the dysbiosis of the gut microbiota can

cause susceptibility to infectious diseases. Furthermore, some microbiotas can promote the growth or increase the virulence of pathogenic bacteria. The microbiota can aid our immunity in several ways, which can be broadly categorised under; nutrient competition, repression of colonisation factors, and activation/de-activation of antimicrobial activities (see Table 1.1). The gut microbiota has been shown to aid the immune system in fighting pathogenic bacteria by out-competing for nutrients. Pathogens establishing intestinal colonisation is difficult and often opportunistic, as the commensal microbiota are highly adapted to the environment and diet. Infections by *E. coli* O157:H7 is on such example, where depending on commensal *E. coli* strains, colonisation can be prevented in mice (Maltby, Leatham-Jensen et al., 2013). This out-competing for nutrients requires the metabolisms of all five key sugars utilised by *E. coli* O157:H7, which can be achieved with the presence of two commensal strains of *E. coli*.

The gut microbiota can also directly repress colonisation by pathogenic bacteria (Payne, Gibson et al., 2003). In the case of *Vibrio cholerae* infection, the commensal bacteria *Ruminococcus obeum* represses colonisation via the quorum sensing molecule AI-2. *Ruminococcus obeum* AI-2 molecule interferes with the expression of the *V. cholerae* toxin co-regulated pilus operon. The expression of the toxin co-regulated pilus operon is required for *V. cholerae* infection of the intestinal tract (Hsiao, Ahmed et al., 2014).

Enteric bacteria have been shown to play a role in modulation of the bacterial community in the gut (Sellon, Tonkonogy et al., 1998). Several studies have identified the production of bacteriocins and microcins by commensal Enterobacteriaceae, *E. faecalis* strain carrying bacteriocin 21 cleared vancomycin-resistant enterococci (Kommineni, Bretl et al., 2015), and microcinproducing probiotic *E. coli* has been shown to limit growth of Enterobacteriaceae (including pathogenic strains) during intestinal inflammation (Sassone-Corsi, Nuccio et al., 2016).

Short chain fatty acids have been shown to have both bacteriostatic and bactericidal effects, for example, short chain fatty acids produced by the commensal bacteria can suppress the growth of *E. coli* strain O157:H7 (Shin, Suzuki et al., 2002). Commensal gut microbes have been shown to enhance bile acid activity (Nie, Hu et al., 2015), *Clostridium scindens* is one such bacterium. *Clostridium scindens* produces 7 α -hydroxysteroid dehydrogenase which is involved in the

conversion of bile acids into secondary bile acids, secondary bile acids, such as deoxycholate and lithocholate have been shown to inhibit the growth of pathogenic bacteria like *Clostridium difficile* (Buffie, Bucci et al., 2015).

Table 1.1 Commensal bacterial species that confer protection against pathogens. Edited from Ubeda, Djukovic et al., 2017 (Ubeda, Djukovic et al., 2017).

Commensal	Pathogen	Mechanism
<i>Staphylococcus lugdunensis</i>	<i>Staphylococcus aureus</i>	Peptide antibiotic with bactericidal activity
<i>Enterococcus faecalis</i> with pPD1 plasmid	Vancomycin-resistant <i>Enterococcus</i>	Plasmid-encoded bacteriocin that inhibits pathogen growth
<i>Bacillus thuringiensis</i>	<i>Clostridium difficile</i>	Bacteriocin with bactericidal activity
<i>Escherichia coli</i> strain Nissle 1917	<i>Salmonella typhimurium</i>	Microcins with antimicrobial activity
<i>Clostridium scidens</i>	<i>C. difficile</i>	Conversion of primary to secondary bile acids which inhibit pathogen growth
<i>Ruminococcus obeum</i>	<i>Vibrio cholerae</i>	Quorum-sensing signals that interfere with pathogen gene expression
<i>E. coli</i> strains HS and Nissle 1917	<i>E. coli</i> O157:H7	Competition for carbohydrates
<i>E. coli</i> , <i>Bacteroides thetaiotaomicron</i>	<i>Citrobacter rodentium</i>	Competition for carbohydrates
<i>E. coli</i> strain Nissle 1917	<i>S. typhimurium</i>	Competition for iron
<i>B. thetaiotaomicron</i>	<i>Candida albicans</i>	LL-37 antimicrobial peptide induction
<i>Bifidobacterium</i>	<i>E. coli</i> O157:H7	Inhibition of Shiga toxin dissemination
<i>Lactobacillus reuteri</i>	<i>C. albicans</i>	Induction of type 3 innate lymphoid cells expansion and interleukin 22 production through tryptophan conversion to an aryl hydrocarbon receptor ligand
Segmented filamentous bacterium	<i>C. rodentium</i>	Induction of T helper 17 cells differentiation and subsequent expression of antimicrobial peptides
<i>E. coli</i>	<i>S. typhimurium</i>	Systemic induction of IgG

1.6.3 The gut flora and disease markers, progression and prevention

The co-evolution between microbes and their eukaryotic hosts has led to an important cross-kingdom relationship. Several studies have highlighted how movement from a 'healthy' gut biome relates to disease, illustrated in Figure 1.16 and Table 1.2.

Current research suggests the gut microbiota can impact several types of cancer via roles in inflammation, DNA damage and apoptosis. The relationship between the immune system, microbiome and cancer can be seen in appendices section 10.1.2. The gut microbiome and its metabolome can be affected by environmental factors, diet in particular. Interestingly while the microbiome can be affected by diet, diet and its nutritional value can itself be affected by the microbiome. The microbiota can affect the way in which we metabolise and store energy, this in turn has implications in obesity and fat storage (see appendices section 10.1.3). It has been hypothesised that a diverse gut microbiota is important for host diet regulation, as dominance by certain groups of microbes could create constant biased nutritional drives on the host, potentially causing dietary patterns and/or preferences (Alcock, Maley et al., 2014).

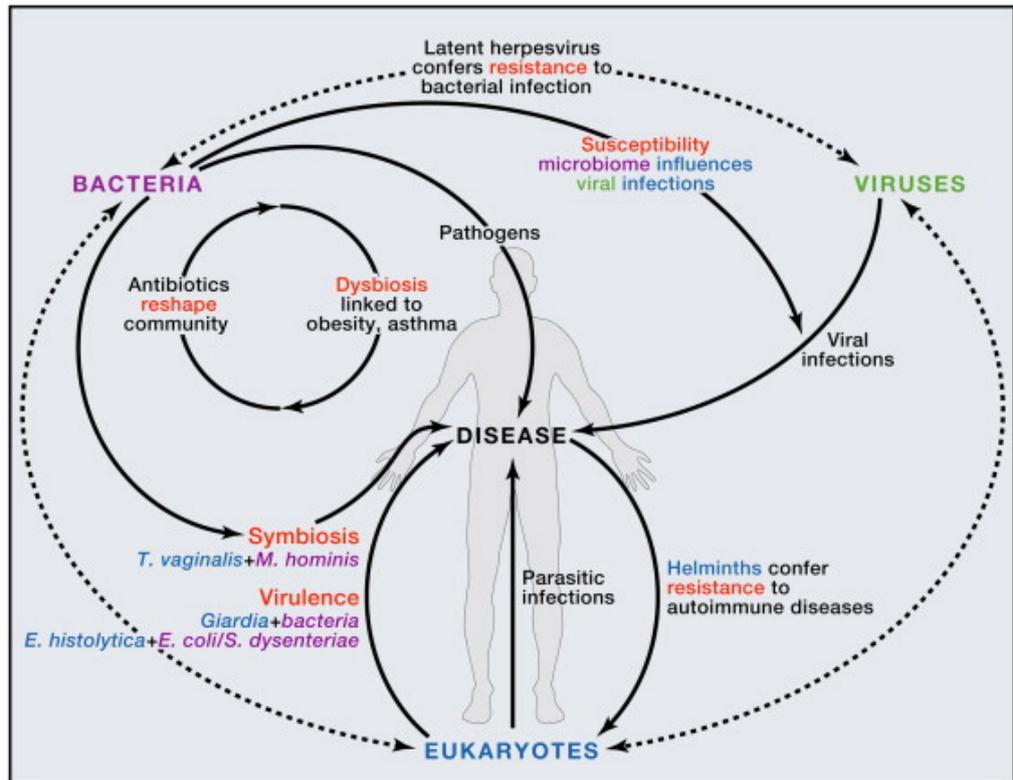


Figure 1.16 Effect of Interactions of Bacteria, Viruses, and Eukaryotes in Health and Disease (Clemente et al., 2012). In the past, disease and disease progression were investigated under the pretence that ‘cause’ is related to individual microbial entities. However there is an emerging understanding that disease phenotypes are a result of pan-kingdom interactions. Virulence of some eukaryotes is, for instance, linked to the presence of certain bacteria, such as in the case of *E. histolytica* and *E. coli* or *S. dysenteriae*. The susceptibility of the host to viral infections is conditioned by the particular configuration of the microbiota, whereas herpesvirus infection can confer resistance to certain bacterial infections. Antibiotics can significantly reshape the composition of the microbiota. As a clear correlation has been observed between many diseases and dysbiosis, the widespread use of antibiotics may be linked to the dramatic increase observed in autoimmune diseases over the last years. Conversely, helminthes confer resistance to autoimmune diseases (Clemente et al., 2012).

Table 1.2 Changes in the Gut Microbiota Associated with Disease. Information accumulated from the following publications: (Clemente et al., 2012), (Rowland, Gibson et al., 2018), and (Rooks & Garrett, 2016)

Human disease and preclinical models	Microbial metabolites, components or mechanisms	Associated microbes
Allergic and immune disorders		
Asthma	SCFAs	outgrowth of bacteria from the Bacteroidetes phylum
Inflammatory bowel disease	SCFAs, B vitamins	
Autoimmune arthritis	induce T helper 17 cells	segmented filamentous bacteria
allergy	early colonization with Lactobacillus associated	<i>Lactobacillus spp.</i> : decrease
	early colonization with more diverse microbiota might prevent H. pylori tolerance mediated by Tregs that suppress asthma	<i>Bifidobacterium adolescentis</i> : decrease
	higher diversity (Shannon-Wiener index) in Celiac's disease patients versus controls	<i>Helicobacter pylori</i> : decrease
Celiacs disease		<i>Bacteroides vulgatus</i> : increase, <i>Escherichia coli</i> : decrease, <i>Clostridium coccooides</i> : decrease
Cancer		
Colorectal cancer	SCFAs, B vitamins, N1,N12-diacetylspermine	
Gastric cancer	important element in carcinogenic pathway for developing gastric adenocarcinomas	<i>H. pylori</i> : increase
Gynaecological and reproductive disorders		
Bacterial vaginosis and other sexually transmitted infections	Polyamines, HBP	
Preterm labour	SCFAs	
Metabolic disorders		
Cardiovascular disease	TMAO	
Kidney disease	SCFAs, p-Cresol	
Obesity and metabolic syndrome	TMAO	
Type 2 diabetes	TMAO	
Obesity	significant changes in gut microbiota are associated with increasing obesity	Bacteroidetes: decrease, Lactobacillus: increase Firmicutes/Bacteroidetes ratio: decrease

Metabolic disorders		
Cardiovascular disease	TMAO	
Kidney disease	SCFAs, p-Cresol	
Obesity and metabolic syndrome	TMAO	
Type 2 diabetes	TMAO	
Obesity	significant changes in gut microbiota are associated with increasing obesity	Bacteroidetes: decrease, Lactobacillus: increase Firmicutes/Bacteroidetes ratio: decrease
Neurological disorders		
Autism spectrum disorder	4-EPS	
Central nervous system dysfunction	SCFAs	
Autism	increased bacterial diversity in feces of autistic children compared to controls	Bacteroidetes: increase, Proteobacteria: increase, Actinobacteria: decrease, Firmicutes: decrease
Other gastrointestinal disorders		
Infectious colitis	Bile acids	<i>Clostridium difficile</i>

1.6.3.1 Autoimmune diseases

The adaptive immune system deals with foreign material within the body using lymphocyte cells. Lymphocyte cells consist of B cells and T cells, where B cells flag foreign bodies via antibody-antigen interaction, and T cells carry out cytotoxic activity. Autoimmune diseases are the result of a fault in this normal function, leading to the body attacking its own tissues. In the past, studies into autoimmune diseases focused on the afflicted organ/tissue, however recent studies have shown otherwise seemingly disassociated microbial factors as mechanisms underpinning the disease. The gut microbiota have been associated to a number of autoimmune diseases that include arthritis (Wu, Ivanov et al., 2010), joint disease (Rehakova, Capkova et al., 2000), diabetes (Wen, Ley et al., 2008), inflammatory bowel disease (Palm, de Zoete et al., 2014) and colitis (Bohn, Bechtold et al., 2006).

1.6.4 Gut microbiota and antibiotic treatment

Antibiotics are widely distributed in animal feed as growth promoters and for therapeutic purposes. Antibiotic resistant strains of foodborne pathogens are developing due to the widespread practices of antibiotic use (Jimenez, Velazquez et al., 1994, Nonga & Muhairwa, 2010). Antibiotic resistant bacteria are a global public health problem, with resistant species shown to persist in the human gut (Andersson & Hughes, 2011, Jakobsson, Jernberg et al., 2010, Jernberg, Lofmark et al., 2007). Antibiotics have a significant negative impact on the gut flora due to their lack of specificity in the removal of infection, impacting the human metabolism and immune system (Perez-Cobas, Gosalbes et al., 2013). The broad damage to the bacterial microbiota leads to dysregulation of adaptive immune cells, potentially leading to disorders like inflammatory bowel disease (Round & Mazmanian, 2009). Antibiotics also have difficulty in subverting biofilms, due to biofilm resistant mechanisms (Hoiby, Bjarnsholt et al., 2010), which include increased levels of mutations as well as quorum-sensing-regulated mechanisms, chromosomal β -lactamase, upregulated efflux pumps and mutations in antibiotic target molecules (Bjarnsholt, Jensen et al., 2005, Driffield, Miller et al., 2008, Giwercman, Lambert et al., 1990, Molin & Tolker-Nielsen, 2003, Soto, 2013).

1.6.5 Gut dysbiosis and therapy

The interaction between the gut microbiota and host has many benefits to host health, however gut dysbiosis or lack of diversity in its microbiota has serious implications. Dysbiosis of the gut microbiota has been linked to the pathogenesis of many intestinal and non-intestinal disorders (see section 1.6.3). The development of a healthy relationship between host and microbiota early in life is important for maintaining intestinal homeostasis. There are a number of current therapeutic methods in the treatment of an 'unhealthy' gut biome, these include; prebiotics (see appendices section 10.1.4), probiotics, and faecal microbiota transplantation. Probiotics are commonly used to promote healthy gut biomes in vulnerable individuals such as new born infants (neonates), particularly those that are preterm (Deshpande, Rao et al., 2010).

Probiotics are live cultures of identified healthy gut bacteria e.g. bifidobacterium and lactobacillus, which ideally confer health benefits to the host if given adequate levels (see appendices 10.1.4). While the benefits of probiotics are observable in the literature, the colonising

of the gut is not easily established or confirmed. The fold difference in relative abundance of ingested bacteria in comparison to commensal gut bacteria drops several factors of 10 after passing the stomach (see Table 1.3). Ingested strains have been mostly detectable and viable for just a few days, but rarely after a week (Derrien & Vlieg, 2015, Firmesse, Mogenet et al., 2008, Fujimoto, Matsuki et al., 2008, Sanders, Guarner et al., 2013).

Table 1.3 Transit and abundance of ingested bacteria. Edited from Derrien, Muriel. *et al.* 2015 (Derrien & Vlieg, 2015).

Location	Transit time	Relative abundance of ingested bacteria compared to resident bacteria
Stomach	15 min–3 h	100 to 10 000-fold
Small intestine (ileum)	2–5 h	0.01 to 1-fold
Colon (feces)	12–24 h	0.0001 to 0.00001-fold

Faecal microbiota transplantation is the perfusion of treated faeces from a healthy donor via the upper or lower gastrointestinal route (Bakken, Borody et al., 2011). Human faecal transplantation was first described approximately 1700 years ago, but was not broadly practiced until after the first reported use of it as a therapy for *Clostridium difficile* infection in 1983. Unlike probiotics faecal microbiota transplantation includes the transfer of the virome, and shows promising therapeutic effects. The therapeutic potential of faecal microbiota transplantation has been identified in several disorders including; neuropsychiatric conditions (Cenit, Sanz et al., 2017, Evrensel & Ceylan, 2016, Wallis, Ball et al., 2018), irritable bowel syndrome (Johnsen, Hilpusch et al., 2018), autoimmune diseases (Berer, Gerdes et al., 2017), chronic fatigue syndrome (Evrensel & Ceylan, 2016), inflammatory bowel diseases (Paramsothy, Paramsothy et al., 2017), allergic disorders (Liu, Li et al., 2017), and metabolic diseases (Vrieze, Van Nood et al., 2012).

More direct targeting methods for resolving gut microbiome dysbiosis are a particular area of interest in research. Prebiotics and probiotics only promote healthy bacteria, rather than eliminate problem bacteria. Phage therapy is one potential method for regulating/structuring the gut biome as well as promoting healthy metabolic pathways and bacteria (see Figure 1.17), though more research into the gut virome is needed.

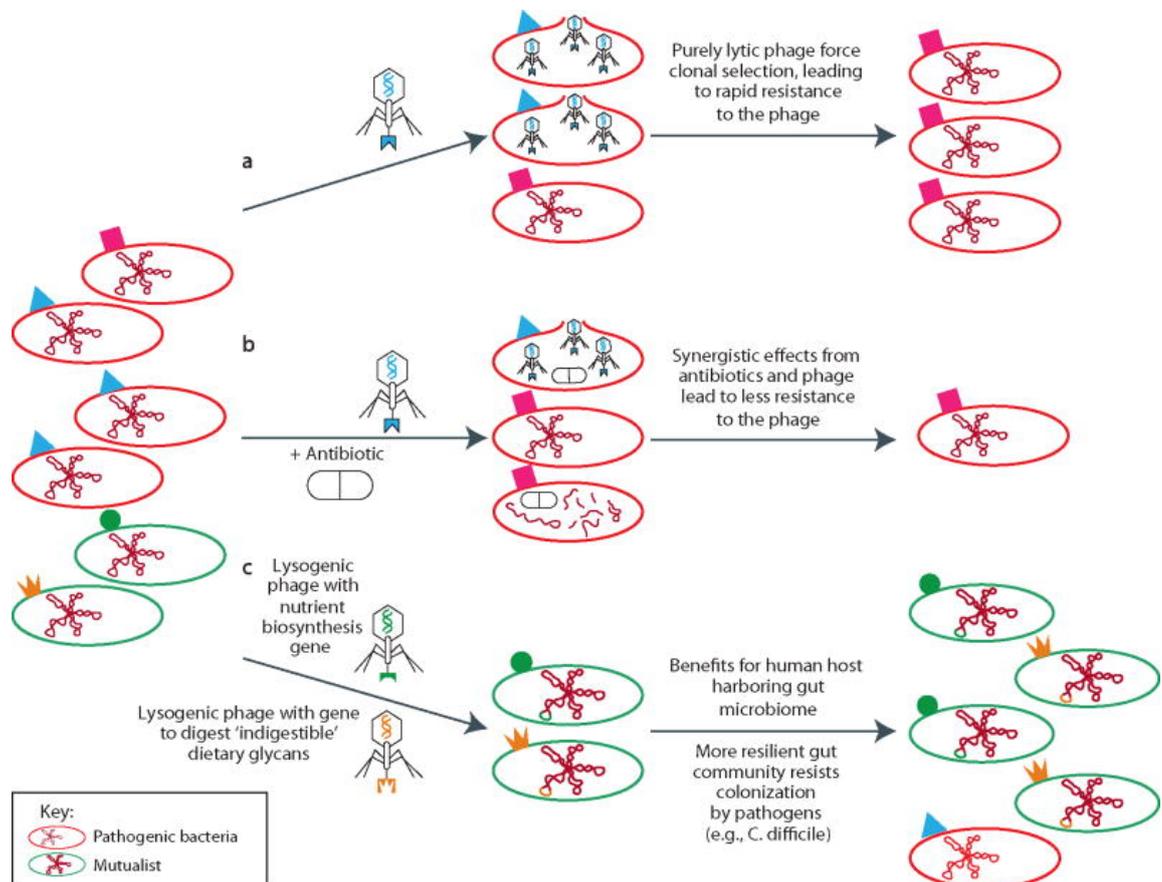


Figure 1.17 Potential strategies for phage therapy.(a) Lytic phage strategies against pathogenic bacteria (b) phage and antibiotics synergy, where lysogenic phages decrease survival of pathogenic bacteria against antibiotics (c) selectively manipulating (enhancing) microbial community functions or clearing the way for invasion by probiotic consortia (Reyes, Semenkovich et al., 2012).

1.6.6 The gut virome

The gut metavirome is a relatively recent area of research with most studies having taken place within the last decade due to the development of DNA sequencing technologies. In comparison to the bacterial fraction a lot less is known and inferred on the significance of the viral community of the gut (see Table 1.4), in particular the significance and community of bacterial viruses or bacteriophages (phages). Recent discoveries have shown that both prokaryotic and eukaryotic enteric viruses have a direct role similar to many of the core aspects of the bacterial community, and an in-direct role via manipulation of the bacterial microbiota, emphasising the importance of understanding the gut virome (Berger & Mainou, 2018, Cadwell, 2015, Kernbauer, Ding et al., 2014, Lukes, Stensvold et al., 2015, Yang, Kim et al., 2016b).

Table 1.4 Known gut viro types according to culture techniques and metagenomics(Scarpellini, Ianiro et al., 2015).

Virus type	Genome type	Environment	Associated disease
<i>Eukaryotic virus</i>			
Rotavirus, Astrovirus, Calicivirus, Norovirus, Hepatitis E virus, Coronavirus and Torovirus, Adenovirus (serotypes 40 and 41)	All RNA except Adenovirus (DNA)	Human small bowel and colon	Gastroenteritis (small bowel epithelium and the absorptive villi disruption, with consequent malabsorption of water and an electrolyte imbalance) (all the mentioned eukaryotic viruses)
Adenoviridae, Picornaviridae and Reoviridae (genus enterovirus)	RNA	Human intestine	Unknown (all the mentioned viruses)
<i>Plant derived virus</i>			
Pepper mild mottle virus (PMMV), oat blue dwarf virus, Grapevine asteroid mosaic-associated virus, maize chlorotic mottle virus, Oat chlorotic stunt virus, Panicum mosaic virus, Tobacco mosaic virus	RNA	Plants and human faeces	Pathogenic or plants Non pathogenic for humans (all the mentioned plant derived viruses)
<i>Giant virus (>300 kb)</i>			
Mimiviridae, Mamaviridae, Marseilleviridae, Poxviridae, Iridoviridae, Ascoviridae, Phycodnaviridae, Asfaviridae	DNA	Human faecal protists, amoebae in lake, river and seawater	Pneumonitis, Children diarrhoea (Mimiviridae only)
<i>Prophages</i>			
Myoviridae, Siphoviridae, Podoviridae, Tectiviridae, Leviviridae, Inoviridae	dsDNA	Human faeces specimens	Unknown (all the mentioned prophages)
<i>Virus</i>			
Microviridae family (Microvirus, Gokushovirinae, Alpavirinae, Pichovirinae)	ssDNA	Seawater, human gut bacteria	Unknown (all the mentioned Microviridae viruses)

1.6.6.1 The gut and eukaryotic viruses

Eukaryotic viruses directly impact human health, and the outcome of infection and survival of enteric eukaryotic viruses within the host is affected by their interaction with the gut microbiota. There are many enteric eukaryotic viruses, which include; Rotavirus (Liste, Natera et al., 2000), Astrovirus (Guerrero, Noel et al., 1998), Calicivirus (Glass, Noel et al., 2000), Norovirus (Lopman, Vennema et al., 2004), Hepatitis E virus (Jameel, Durgapal et al., 1992), Coronavirus and Torovirus (Gerna, Passarani et al., 1985), Adenoviridae (Cruz, Caceres et al., 1990), Picornaviridae (Holtz, Finkbeiner et al., 2008), and Reoviridae (Morrison, Sidman et al., 1991). Examples of eukaryotic viral infection supported by interaction with gut microbiota include; Norovirus, Poliovirus, and Reovirus infection. Enteric bacteria produce histo-blood group antigens which are used by Norovirus to help protect itself from stressors as well as attach and infect B cells (Jones,

Watanabe et al., 2014, Li, Breiman et al., 2015a). The capsid stability and receptor engagement of Poliovirus are improved by bacteria and their lipopolysaccharides (Robinson, Jesudhasan et al., 2014). Virion thermostability of Reovirus is improved via the binding of Gram-positive and Gram negative bacteria through bacterial envelope components (Berger, Yi et al., 2017).

Though enteric eukaryotic viruses are most well known for their pathogenesis, with the advance in metagenomic sequencing/analysis, their continuity within the gut has highlighted the likelihood of other types of viral/host interactions occurring. Recent discoveries have shown that some enteric eukaryotic viruses have a symbiotic relationship with their host (Duerkop & Hooper, 2013, Kernbauer et al., 2014, Yang et al., 2016b). Kernbauer et al. (2014) have demonstrated this in the murine norovirus. The murine norovirus has been shown to support intestinal homeostasis and mucosal immunity, similarly to commensal bacteria. Without bacterial presence in the gut, the murine norovirus is capable of restoring intestinal morphology and lymphocyte function without inducing noticeable inflammation and disease. Furthermore, the absence of bacteria is associated to the expansion of group 2 innate lymphoid cells, which murine norovirus presence suppressed. Murine norovirus presence also induced transcriptional changes in the intestine associated with immune development and type I interferon (IFN) signalling. Murine norovirus is capable of compensating for bacterial depletion and reducing the deleterious effect of treatment with antibiotics in models of intestinal injury and pathogenic bacterial infection (Kernbauer et al., 2014).

1.6.6.2 Phageome of the gut

In 1896 Ernest Hankin first noticed the bactericidal activity of filtered water (Hankin, 1896), and in 1915 Frederick Twort went on to discover the bacteriolytic agent (Twort, 1915). In 1917 Felix d'herelle independently discovered the bacteriolytic agent and went on to develop topical and systemic treatments (d'Hérelle, 1917). Since then there has been much focus on individual interaction between a single phage and its bacterial host. We premise that the phage communities play an essential role in the bacterial gut flora, but the complexity of the gut environment makes it difficult to pinpoint their role and function as a community. In recent years investigation into the gut phageome has increased, with studies elucidating to biomarkers of health and disease as well as

modulation and stability of host/microbiome (Bakhshinejad & Ghiasvand, 2017, Kim & Bae, 2018, Ma, You et al., 2018, Manrique, Dills et al., 2017a, Mirzaei & Maurice, 2017, Ogilvie & Jones, 2017). However research conducted thus far has highlighted how little of the phageome is really known, with core, common and unique free viral particle groups identified in healthy individuals, most of which with no known or associated taxonomy (Manrique et al., 2016). The faecal virome has been identified as more unique to each individual than the bacterial community (Reyes, Haynes et al., 2010), which suggests a need for alternative and additional tools to monitor and understand their relationship to human health. The unique nature of the virome is likely linked to its sensitivity, in the first weeks of life the viral turn-over is significant with over 50% of the community capable of disappearing/changing (Breitbart, Haynes et al., 2008). The gut virome does not follow the bacterial community in abundance in a normal predator prey relation, instead there is a reversed predator-prey life cycle, with sudden drops in viral community linked to increases in bacterial community (Lim, Zhou et al., 2015b). Changes in the gut virome have been linked to age (Early life dynamics of the human gut virome and bacterial microbiome in infants (Lim, Zhou et al., 2015a)), malnutrition (Gut DNA viromes of Malawian twins discordant for severe acute malnutrition (Reyes, Blanton et al., 2015)), infectious and autoimmune diseases (Munz, Lunemann et al., 2009), metabolic disorders (Honeyman, Coulson et al., 2000) and cardiovascular disease (Guo, Hua et al., 2017).

There have been extensive studies into the gut bacterial community structure and dynamics, though there is still much to discover of the bacterial-host interaction. Comparatively, there has been far less research carried out on the gut virome, though still numerous studies elucidate the significant role the virome plays on human health. Of particular interest is the role lysogeny has on regulating/manipulating the gut biome, and suggestions of known phage interactions with the bacterial hosts that likely have significant effects in the gut can be seen in Figure 1.10.

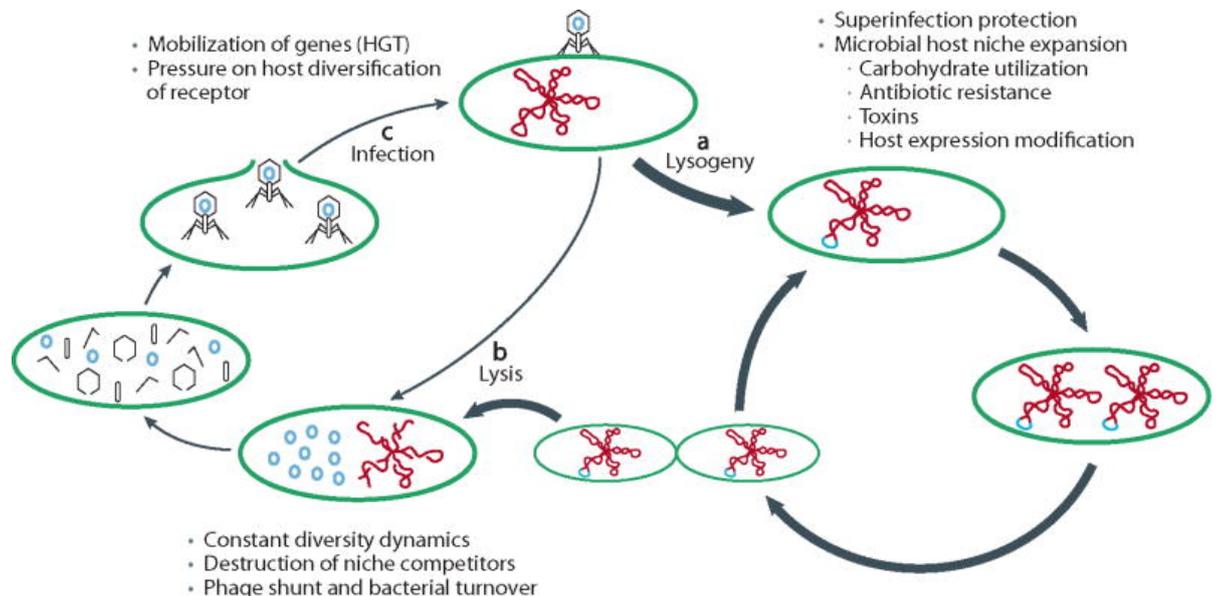


Figure 1.18 Potential consequences of a temperate phage lifecycle in the human gut. (a–c) Illustration of the benefits of this temperate lifestyle on phage-host dynamics (Reyes et al., 2012).

1.7 Aims of the project

This research initially aims to characterise changes in *E. coli* microbial physiology after infection and integration of the gut related phage $\phi 24_B$ (discussed previously in section 1.5.1). It further aims to investigate if phage conversion aids selection compared to its naïve counterpart, looking to infer potential mechanisms. This study aims to characterise the viral influence on the gut microbiota, by comparing the free viral particles and lysogenic phage fraction, using whole genome shotgun sequencing. As well as construct tools to mainstream larger meta-omic analysis with particular focus on viral analytical problems. The aim is to identify differences and uses of pan-kingdom gut analysis, with focus on advantages and additional information the virome provides. Viral analysis includes a novel induced viral strategy, with the aim of understanding more of the lysogenic community. The storage effects on stool samples has never been assessed in relation to viral communities. Therefore as well as investigating further into the viral gut biome, this research also aims to identify common storage methods and their effects on viral community analysis.

Chapter 2. General Materials and Methods

2.1 Materials and growth media constituents

2.1.1 Sterilisation

All media (broth, agar, buffers), glassware, and consumables (pipette tips, que tips etc) used were autoclaved at 121 °C for 20 minutes' cycle at a pressure of 15 psi.

2.1.2 Broth and Agar

All strains were grown in Lysogeny Broth (LB) and on bottom LB agar plates (BA). LB broth; 12.5 g LB broth (Sigma Aldrich, Gillingham, UK), 5 ml of calcium chloride (CaCl₂) and 495 ml of distilled water. LB BA; 12.5 g of broth, 7.5 g of bacteriological agar (Sigma Aldrich, Gillingham, UK), and 500 ml distilled water. Prior to experimentation ϕ 24_B:: Δ Kanamycin MC1061 is always propagated on LB agar containing 50 μ g.ml⁻¹ kanamycin (kan) (Sigma Aldrich, Gillingham, UK). Soft (top) agar (SA) was made up with 5 g of LB broth and 0.8g (0.4% w/v) of phage agar, 0.01M calcium chloride and 200 ml distilled water. All culture work was achieved in 25 ml universals containing 10 ml of LB broth if not stated.

2.1.3 Buffers and inducers

Buffer	Constituents
Phosphate buffered saline (PBS)	137 mM NaCl, 2.7 mM KCl, 10 mM Na ₂ HPO ₄ , 2 mM KH ₂ PO ₄
Calcium Chloride (CaCl ₂)	1 M
Norfloxacin (NFLX) (Sigma Aldrich, Gillingham, UK)	1 mg.ml stocks, in water, add a few drops of 1 M NaOH to alter pH so drug dissolves into solution

2.2 Bacterial and viral strains

$\phi 24_B::\Delta Kan$ was obtained from the University of Liverpool. This strain originates from an Stx-phage induced from strain E86654, a clinical isolate of *E. coli* O157:H7 (Colindale Public Health Laboratories, CHPL) expressing the Stx2 toxin. The Stx2A gene was inactivated with a kanamycin resistance cassette (aph3) from plasmid pUC4K (Pharmacia) (Sergeant, 1998). This phage ($\phi 24_B$ (Stx2A::aph3)) was renamed $\phi 24_B::\Delta kan$. $\phi 24_B::\Delta cat$ was obtained from the University of Liverpool. This strain is a second construct made using the same Stx-phage wild type described above, including a truncated Stx2A gene and the inclusion of a chloramphenicol acetyl transferase gene (cat) from pLysS (Novagen) (Allison et. al., 2003; James, 2002). This phage ($\phi 24_B$ (Stx2A :: Δcat)) was named $\phi 24_B::\Delta cat$. All phage stocks were stored at 4 °C in LB plus 0.01M CaCl₂ (phage buffer). *E. coli* K-12 strain MC1061 was used as a host for the productions of lysogens. *E. coli* strain DM1187 was used as the host for all of the bacteriophage enumeration work done with $\phi 24_B$ in this study. DM1187 contains the mutation 'recA441' that results in a constant expression of recA, which initiates the gene cascade of the $\phi 24_B$ lytic cycle, thus directing lysis.

2.3 Growth and Maintenance

2.3.1 Induction - Checking successfully integrated phage

Unless otherwise stated all strains were grown at 37 °C and all broth cultures were also shaken at 200 rpm. Subcultures were grown to mid exponential (0.5-0.6 OD₆₀₀). The temperate virus was induced through the addition of 10 μ l of 1 mg. ml⁻¹ stock of norfloxacin (NFLX) and grown for 1 hour. The NFLX is diluted by transferring 1 ml of induced culture to a sterile 10 ml buffer solution, and re-incubated for ~2 hours. The bacteriophage lysate was filtered through a 0.22 μ m syringe filter. A 10-fold serial dilution of the phage lysate was run (450 μ l of buffer and 50 μ l of lysate). Soft agar was divided into 5 ml aliquots (kept at 50 °C). 100 μ l inoculum of MC1061 at mid exponential growth was added to each phage lysate dilution. This was incubated at 37 °C for 25 minutes, each were added to individual soft agar aliquots, and individually poured onto separate

bottom agar plates. Plates were incubated for 18 hours. Plates containing between 30-300 plaques were counted. DM1187 cannot withstand the lysogenic integration/infection, therefore no growth presents a control to ensure an integrated MC1061 with the temperate phage.

2.3.2 Stocks, overnights and sub-cultures

Cells for frozen stock were grown at 37 °C and harvested at mid exponential growth phase, the cells were suspended in 50% glycerol and stored at -80 °C. All stock plates were grown overnight at 37 °C from frozen stocks. All overnights were grown from single colonies of a given stock plate for 18 hours at 37 °C. All sub-cultures were prepared from the overnights at a 1% inoculum and grown at 37 °C.

2.4 Bacterial phenotypic microarray

The 2 bacterial strains used are *E. coli* strain MC1061 and MC1061 infected with detoxified model Stx-bacteriophage vB_EcoP 24B. Before inoculation into the Biolog Panel plates, strains were raised from – 80 °C stocks and passaged from a single colony x2 on Luria bertani agar (LBA) or LBA containing Kanamycin (50 µg.ml⁻¹) for the vB_EcoP 24B MC1061 lysogen. Inoculum was made using a cotton swab taken from the associated plate into Inoculation fluid IF-0 (containing 50 µm leucine due to MC1061's auxotrophy), to a transmittance of 42% T on a biolog turbidometer in a 20 mm diameter tube, and inoculated as per manufacturer's instructions.

The panel plates used for this study included Biolog plates PM 1-20 which include a plethora of both metabolic and toxicological (respiration and metabolism of different carbon sources PM 1, 2a; PM3B identifies respiration on different Nitrogen sources; PM4A identifies utilisation of different phosphorous and sulphur sources; PM5 investigates the utilisation of other nutrient supplements including amino acids; PM 6,7, and 8 identifies utilisation of a range of peptide Nitrogen sources; PM 9 identifies the pressure of osmolytes on respiration; PM10 looks at the effect of pH on respiration; PM11C, 12B, 13B, 14A, 15B, 16A, 17A, 18C, 19 and 20B look at a wide range of chemical sensitivity compounds including antibiotics on the respiration of the tester strains. Further details of the components associated with these PM plates can be found at

<http://www.biolog.com/pmMicrobialCells.html>. The Biolog PM plates after inoculation were grown at 37 °C and monitored using the Omnilog plate reader at 30 min intervals over a 47 hour period.

Area under the curve (AUC) for respiration values were calculated using a trapezoid algorithm (Liengme, 2002). Statistical significance of area under the curve (AUC) and comparison at a specific time point during mid exponential growth phase was achieved by checking normal Gaussian distribution by parametric analysis and statistical significance identified using an unpaired *t* test using Stats Direct statistical software package.

2.5 Sub-inhibitory concentration (SIC) assay, 96 well plate assay set up

From previously formed stock plates, 4 new spread plates were made, 2 for the MC1061Φ24_B (lysogen) and 2 for the MC1061 (naïve host). This was conducted by selecting a few individual colonies from each stock plate and transferring them to correspondingly labelled fresh LB agar plates, using a spreading technique. The spread plates were incubated for 18 hours.

1x phosphate buffered saline (pbs) is a 1 in 10 dilution of the 10x pbs stock, which was diluted using sterile water. This 1x pbs is then added to LB and CaCl₂ buffer, in a 3 fold dilution (2/3 buffer, 1/3 1x pbs). This solution was made up in plastic universals, to be used as the inoculant buffer. Sterile swabs (dipped in a 1x pbs solution) were used to remove colonies from the overnight plates and infect individual 3 fold solutions (buffer), one for the MC1061Φ24_B and one for the MC1061. This infection technique was repeated until both inoculants had a transmittance of 42 % T (tested in 1 ml plastic cuvettes, at OD₆₀₀).

2.5.1 SIC assay

Table 2.1 Antimicrobial stock solution content

Antimicrobial drug	Vehicle for dissolving drug (µl)	Weight of drug added	Distilled sterile water added (µl)
8-Hydroxyquinoline	950 of 98% ethanol	0.05 g	Nothing required
Oxolinic acid	10 of 48% sodium hydroxide	0.05 g	940
Chloroxylenol	950 of 98% ethanol	0.05 g	Nothing required

Note* Sodium hydroxide had an MIC between 0.0048% (0.048 µg.ml⁻¹) and 0.00048 % (4.8 ng.ml⁻¹) against both lysogen and naïve host. Naïve host and lysogen showed a greater tolerance to ethanol (stock solution of 98 %), which had no effect at the strongest dilution tested of 9.8 % (0.098 mg.ml⁻¹).

The pinpointed MICs were doubled to account for the dilution when mixed 50/50 with broth culture. The antimicrobials were dissolved as shown in Table 2.1, these 50 mg.ml stocks were then diluted into the range of desired dilutions using LB and CaCl₂ buffer.

Well rows A-H and columns 1-10 are the drug dilutions tested on inoculants, each of these wells were made up with 75 µl of inoculant and 75 µl of diluted drug. Rows A-H were split into 2 groups of 4, one group had the addition of lysogen inoculant and the other group the MC1061 inoculant. 150 µl of drug dilution is added to the 11th column (the blank). The control (column 12) was split into two separate groups corresponding to MC1061Φ24_B and MC1061, 150 µl of inoculant was coherently added to each well. Three plate assays were performed for each antimicrobial, an initial reading was taken at 0 hours, after 18 hours incubation a final reading was taken.

2.6 SIC LCMS analysis comparing metabolic compounds from Naïve host and Lysogens

In broth cultures were grown of naïve host, 24_B::ΔKanamycin MC1061 and 24_B::Δchloramphenicol MC1061. A growth curve was carried out for each to provide a known turbidity and absorption for early, mid and lag phases, 1.5 ml was extracted from the growing culture at early, mid, and late log phase, each of these samples was subjected to spinning at 12000 g for 5 minutes, the supernatant is then removed and the pellet is re-suspended in 1 ml 1x pbs, the

spinning and re-suspension is carried out 3 times (this is a washing step) before finally removing the supernatant and subjecting the pellet to lyophilisation overnight. 100 mg of each sample was then extracted with 1 ml methanol including 0.1% formic acid, this is vortexed for 3 minutes, sonicated for 15 minutes, and spun down for 5 minutes. 750 μ l is removed and filtered through 0.22 μ m pore size nylon and injected into the Finnigan LCQ Advantage LC/MS whereby separation was performed using the C18 column Phenomenex Gemini (110A, 150x2 mm, 5 μ m, flow 0.2 ml/min). Progenesis QI software was used for raw data analysis, consisting of: normalisation, alignment, statistical calculations and compound identification. Further bioinformatics was carried using the self built CRACCD program, following the pipeline from start to finish (scripts can be found in the appendices), where p value, CV%, m/z and retention time settings were ≤ 0.05 , ≤ 10 , 0.1 and 0.5 respectively. Plots were carried out as per CRACCD R scripts, which can be found in the appendices.

2.7 SIC assay bacterial strains and growth conditions - Buffer and Agar

All bacterial strains were grown in Lysogeny Broth + 0.01 M CaCl₂ (LB). Growth of the MC1061(ϕ 24_B::Kan), a lysogen of the bacteriophage ϕ 24_B::Kan and growth of the MC1061(ϕ 24_B::Cat), a lysogen of the bacteriophage ϕ 24_B::Cat was supplemented with 50 μ g.ml⁻¹ kanamycin (kan) and Chloramphenicol (cat) respectively. Bottom agar plates for plaque assay included LB broth including 7 % (w/v) grade 1 agar. Soft top agar was contained LB broth plus 0.4 % (w/v) grade 1 agar. Unless otherwise stated culture conditions were at 37 °C, and broth cultures were shaken at 200 rpm.

2.8 Growth curve of single and double lysogens

A single colony of either naïve MC1061, single or double lysogen was cultured overnight for 18 h (200 rpm). LB with 0.01M CaCl₂ (100 ml) was inoculated with 1 % (v/v) of the overnight culture. Samples were taken over a 7 hour period, subject to serial, ten-fold dilutions and spread plated on LB agar plates.

2.9 Bacterial phenotypic microarray - Biolog

The Biolog assay utilises a redox dye where a tetrazolium violet salt acts as an electron receptor from the tricarboxylic cycle and reduction to NADH. The transfer alters the clear salt to a purple formazan dye that is inexplicably linked to the cellular activity, specifically cell respiration. An inoculum was taken from an 18 h streaked plate of either MC1061 or ϕ 24_B lysogen, raised through 2 rounds of passage from single colony amplification from cryo-stock. A single colony was added to fluid IF-0 (containing 50 μ m leucine due to MC1061's auxotrophy), to a transmittance of 42% T on a Biolog turbidometer in a 20 mm diameter tube as per manufacturer's instructions and used to inoculate Biolog Phenotypic Microarray plates.

The panel plates used for this study included Biolog plates PM 1-20, which include a range of both metabolic and toxicological additives (see SI). Further details of the components associated with these PM plates can be found at <http://www.biolog.com/pmMicrobialCells.html>. The Biolog PM plates were grown at 37 °C and monitored using the Omnilog plate reader at 30 min intervals over 47 hours.

2.10 SIC assay: LCMS and MSMS preparation and run.

Replicated in triplicate therefore n=9 bacterial cultures (10 ml) were grown as previously described under standard growth conditions and challenged with antimicrobials, the cells were harvested at early, mid, and late log phase. The cells were harvested by centrifugation (5,000 g for 5 min) and the pellet washed (x3) in ice cold 1 x PBS prior to lyophilisation. Lyophilised samples (x3) were pooled, normalised for weight/vol (normalised to 1 mg.ml) with methanol and 0.1% formic acid, this was vortexed and then sonicated (Bandelin Sonorex, Sonicator) for 1 hour and centrifuged (5000 g for 5 mins). The supernatant was recovered and filtered through 0.22 μ m pore-sized, nylon filter and injected into the Q-Exactive LC-MS (Thermo-Fisher) after separation on a Phenomenex Gemini column (110A, 150x2mm, 5 μ m, flow 0.2 ml.min). LCMS mobile phase parameters were: 0-6 mins at 20% ACN, 8 mins 60% ACN, 12 mins 95% ACN, 17 mins 95% ACN, 17.1 -23 mins 5% CAN. MS conditions were: full MS mode, resolution 70, 000, AGC target 1×10^6 , maximum IT 200ms, scan range 150-2000, column temperature 35 °C. Metabolites were

confirmed by analysing pure standards and MSMS fragmentation analysis run under identical analytical conditions.

2.11 Biotin quantification assay

Inoculums were prepared in the same manner as for the SIC and Biolog assays. Optical density values were taken at 0 and 18 h, incubation at 37 °C. The cultures were diluted to the lowest OD₆₀₀ reading to normalise cell number between naïve MC1061 and lysogen. Dilutions of both naïve MC1061 and lysogen were made in LB (1:100 and 1:1000). Cells were harvested by centrifugation (5,000 rpm for 5 mins). The biotin assay was completed using the Bio Vision® (Cambridge, UK) Biotin Quantitation Kit (Colorimetric) according to the manufacturer's protocol. The Biotin Quantitation kit is based on the differential binding of Streptavidin to Biotin and the 2-(4-hydroxyazobenzene) benzoic acid (HABA) dye. The biotinylation of the Streptavidin and dye complex results in the displacement of HABA, which in turn changes the absorption readings of the solution. In brief, the supernatant was discarded and the pellet re-suspended in 10 µl 1x pbs and heated to 100 °C for 3 min and then immediately placed on ice. Diluted naïve and lysogen cells (10 µl) were added to individual aliquots of Biotin Assay buffer (20 µl) and 300 µl of biotin reaction mix, pre-prepared as described in the Biotin Quantitation Kit protocol (version 7.6), was added to the buffer and cells. The mix was incubated at 21 °C for 15 min. Each sample mix (150 µl) was then pipetted into a microtitre plate and read at 500 nm. A standard curve was prepared as per manufacturer's instructions.

2.12 Cell harvesting of MC1061 and ϕ 24_B grown under sub-inhibitory antimicrobial challenge

The MC1061 and MC1061 ϕ 24_B::Kan subcultures were prepared in 500 ml glass conicals containing 100 ml LB broth and grown for 1 hour to establish stable early growth. Cultures were then normalised to an absorbance of 0.100 at 600 nm. All culture conditions were run to n=9. Upon normalising the abundance, either 8-hydroxyquinoline or chloroxylenol were added to the cultures at 0 hours to make a final concentration of 50 µm. Cultures were then incubated for 6 hours at 37

°C and shaken continuously at 200 rpm. Cells were harvested from the 100 ml culture using centrifugation at 5000 rpm for 10 minutes at 4 °C. Pelleted cells were kept on ice and washed with 3 ml sterile ice cold Millipore water, this was repeated a further two times. After the final wash, cells were centrifuged again at 5000 g for 10 minutes at 4 °C, any remaining supernatant was removed, and pellets were then stored at -80 °C for 20 minutes. After the 20 minute incubation at -80 °C, samples were freeze dried overnight (18 hours). Controls were run alongside in an identical manner, but without antimicrobial challenge.

2.13 Growth and cell harvesting of MC1061 and ϕ 24_B under increasing antimicrobial challenge

2.13.1 Preliminary growth curves

The MC1061 and MC1061 ϕ 24_B::Kan subcultures were prepared in 250 ml glass conicals containing 100 ml LB broth and grown for 18 hours, both cultures were run to n=9. Growth was monitored at 0, 3, 6, 9, 12, 15, and 18 hours using a spectrophotometer set at 600 nm. At each time point 10 μ l of culture was removed (n=3), a serial dilution was run of each from 1×10^1 to 1×10^{12} (10 μ l in 90 μ l) and all dilutions were plated. Plates were incubated at 37 °C overnight, and plate dilutions containing 30-300 colonies were counted.

2.13.2 Growth and cell harvest

The MC1061 and MC1061 ϕ 24_B::Kan subcultures were prepared in 2.5 L glass conicals containing 1 L LB broth and grown for 1 hour to establish stable early growth. Cultures were then normalised to an absorbance of 0.100 at 600 nm. All culture conditions were run to n=9. Cultures were subjected to an increased total concentration of antimicrobial (8-hydroxyquinoline or chloroxylenol) at 0, 3, 6, 9, 12, 15, and 18 hours. The total antimicrobial concentrations at each step were 50 μ m, 100 μ m, 150 μ m, 200 μ m, 250 μ m, 300 μ m, 350 μ m respectively. Growth was monitored at each concentration increment using a spectrophotometer set at 600 nm, and 100 ml of solution was removed for harvesting cells. Cells were harvested from the 100 ml aliquots using

centrifugation at 5000 g for 10 minutes at 4 °C. Pelleted cells were kept on ice and washed with 3 ml sterile ice cold Millipore water, this was repeated a further 2 times. After the final wash, cells were centrifuged again at 5000 rpm for 10 minutes at 4 °C, any remaining supernatant was removed, and pellets were collected in the -80 freezer. After harvesting cells from each time point, they were removed from the -80 and freeze dried overnight.

Several controls were run, alongside the above test group, these included cultures with: no antimicrobial given, final antimicrobial concentration set at 0 hr, and final antimicrobial concentration set at 18 hr. Cell counts were retrospectively calculated from previous preliminary growth curves.

2.14 Cell wall fatty acid methyl ester isolation

Strains were harvested as per methods section 2.13.2. and 2.12 Fatty acid methyl esters (FAMES) were prepared following the procedure described by Suzuki and Komagata (1987) (Suzuki & Komagata, 1983). Dried biomass preparations (ca. 25 mg) in 8.5 ml test tubes fitted with Teflon-lined screw caps (Aldrich Ltd., The Old Brickyard, New Road, Gillingham, Dorset, UK) were treated with 2 ml of dry-methanol-sulphuric acid (98.5:1.5, v/v) at 50 °C overnight. After cooling to room temperature, 1 ml distilled water and 3 ml n-hexane were added to the resultant preparations which were shaken and left to stand for five minutes when the hexane extracts were transferred to clean test tubes. n-Hexane (3 ml) was added to each of these preparations and the resultant mixtures shaken, left to stand for five minutes and the hexane extracts transferred to clean test tubes, a step that was repeated. Distilled water (9 ml) was added to each of the hexane extracts, the mixtures gently inverted, left to stand for five minutes when the upper hexane layers were transferred to clean test tubes containing anhydrous sodium sulphate (Sigma). The purified hexane extracts were concentrated under nitrogen to give a final volume of 0.25 ml. The resultant preparations were stored at -20 °C until required.

2.15 Gas chromatographic analysis of fatty acid methyl esters

The purified FAMES were separated and quantified using a 610 Series Gas Chromatograph (ATI Unicam, York Street, Cambridge, UK) equipped with a high resolution polar gas

chromatography column (30 m x 0.25 mm; J & W Scientific, UK); the temperature for both the injector and flame-induced-detector (FID) was kept at 270°C. The column was programmed to operate from 100°C to 240°C with 3°C increases per minute, using nitrogen as the carrier gas; the pressure of the nitrogen, air and hydrogen were kept stable at 20 pounds per square inch. Peak areas and retention times were recorded using an ATI Unicam Software Package (ATI Unicam, York Street, Cambridge, UK) and eluted components identified by using a FAMES standard mixture (C8-C22) prepared from fatty acid methyl ester standards (Sigma-Aldrich Company Ltd., Fancy Road, Pool Dorset, UK). The percentage fatty acid composition of each isolate was tabulated with percentages below 1.99 % labelled as trace amounts.

2.16 Stool sample collection, storage, preparation, and DNA extraction:

2.16.1 Patient cohort

Participating infants were all enrolled in the ongoing and previously reported sample salvage study, SERVIS (Stewart, Nelson et al., 2013a). SERVIS has received ethical approval and with all participants are included following individual signed parental consent. Samples chosen for this study were designed to explore twin differences and similarities and differences occurring over a significant time interval (4 weeks). Patient inclusion in the study was opportunistic, comprising a cohort of two twin pairs and a singleton. Infants' demographic and clinical details are shown in Table 7.1. All infants received maternal breast milk, prophylactic antifungals (fluconazole) until fully fed, probiotics (Infloran) from first milk tolerance to 34 weeks corrected age, and were managed with a standardised feeding protocol.

2.16.2 Stool sample collection, storage,

Stool samples were collected from 5 infants, including 2 twin pairs, on two occasions, once towards the end of the first month of life, and again towards the end of the second month (Table 7.1). All patients were housed within the NICU Royal Victoria Infirmary at Newcastle upon Tyne. Samples were stored at 4 °C and processed within 12 hours of sampling. Storage of the samples was – 80 ° C until second time point. It should be noted that a single removal of 120 mg was used from each sample, where each 120 mg were not subdivided (except for the induction step). Instead

all 4 types of community analysis were extracted from different phases of the single 120 mg extraction, to maximise comparability.

2.16.3 Free viral particle (FVP) isolation

The isolation of free viral particles is the process of filtering for any viral particles within the stool, without chemical induction. This approach means only lytic and spontaneously induced viruses are isolated. 120 mg of stool was homogenised in 3 ml of ice cold, sterile 1x pbs and allowed to settle for 5 min. 1ml of the supernatant was used for viral DNA extraction. This was centrifuged at 4000 rpm for 10 minutes, and the supernatant removed. The pellet was used for the inducible virus protocol.

2.16.4 Induced viral particle isolation

The pellet from 'Free viral particle isolation' was resuspended in 1 ml sterile, room temperature, 1x pbs. Virus induction was achieved with addition of norfloxacin at 1 µg/ml and incubated for 1 hour at 37 °C. After incubation each sample was centrifuged at 4000 g for 10 minutes.

2.16.5 Viral DNA extraction

DNA was extracted from the free and chemically induced viruses using the NORGEN Phage DNA Isolation Kits (Geneflow Limited, Lichfield, UK). The manufacturers protocol was modified as per Tariq *et al.* (2015), for removal of bacterial/eukaryotic chromosomal DNA. Kit negatives were processed with each batch.

2.16.6 Total DNA isolation for microbial community analysis

DNA was extracted from 1 ml of homogenised stool, harvested as a pellet after centrifugation at 4000 g for 10 minutes, and the supernatant discarded. Prior to DNA extraction, extracellular DNA was depleted through 1 x round of 1 µL of TURBO DNase and 1 µL of RNase Cocktail (Life Technologies Limited), the solution was incubated at 37 °C for 30 min. The DNase and RNase were inactivated using heat at 65 °C and 15 mM EDTA final concentration for 10 min. QIAGEN DNeasy PowerLyzer PowerSoil DNA Isolation Kits (Geneflow Limited, Lichfield, UK)

was used as per manufacturers instructions. Kit negatives were processed with each batch. All DNA samples were stored at -80 °C.

2.17 Genome sequencing

2.17.1 Viral metagenomic sequencing

Whole genome sequencing of the viral samples was carried out using Illumina sequencing by synthesis. Libraries were made using Nextera XT (Illumina, Saffron Waldon, UK) library preparation kit and sequenced using the V3 600 cycle kit (Illumina, Saffron Waldon, UK). Sequencing was carried out by the NU-OMICS sequencing service (NU-OMICS, Northumbria University at Newcastle, UK).

2.17.2 Bacterial community amplicon sequencing analysis

The 16S rRNA gene, V4 region was used as a target for amplicon sequencing using the approach detailed by Kozich et al. 2013 (NU-OMICS, Northumbria University at Newcastle, UK). Libraries were made using the Patch Schloss mothur SOP (Schloss et al. 2013), and sequenced using the V2 500 cycle kit (Illumina, Saffron Waldon, UK). Sequencing was carried out by the NU-OMICS sequencing service (NU-OMICS, Northumbria University at Newcastle, UK).

2.17.3 Fungal community amplicon sequencing analysis

The ITS1 and ITS2 region (Gardes & Bruns, 1993, J White, Bruns et al., 1990) of simple eukaryotes was targeted for amplicon sequencing, using the barcodes from Kozich et al 2013 to offer a paired-end approach sequencing (NU-OMICS, Northumbria University at Newcastle, UK). Libraries were prepared using an edited Patch Schloss mothur SOP (Schloss et al. 2013), and sequenced using the V2 500 cycle kit (Illumina, Saffron Waldon, UK). Sequencing was carried out by the NU-OMICS sequencing service (NU-OMICS, Northumbria University at Newcastle, UK).

2.17.4 Data clean-up and read count/distribution

All fastq files were de-multiplexed and quality filtered using cutadapt (Martin, 2011) and sickle (Joshi NA, 2011). Cutadapt (version 1.18) was then used to search for, and remove, all potential primer variations used within the data including full primer lengths and separate oligo

flow cell attachment sites, all of which was repeated using the reverse complements. Sickle (version 1.7) quality and length thresholds were set at 30 and 15, respectively. Bacterial and fungal operational taxonomic units (OTU's) were defined using the OptiClust method in Mothur (Schloss, Westcott et al., 2009). Taxonomy for each OTU was assigned using the Silva database (Quast, Pruesse et al., 2013) for bacterial OTUs and the UNITE database (Koljalg, Larsson et al., 2005) for fungal OTUs. Bacterial and fungal reads were trimmed, merged and processed using Mothur (Kozich, Westcott et al. 2013). After which, all taxa associated with kit and sequencing negatives were removed. Rarefaction analysis was performed in R and data was rarefied accordingly.

Viral community analysis was performed in *blastn* (version 2.5.0) and MEGAN (version 6.12) (Huson, Auch et al., 2007a). Viral sequences were compared through local alignment to the NCBI viral database (2016 *viral_genomic.fna*) using *blastn* (Altschul, Gish et al., 1990). *blastn* data was normalised using MEGAN, and read assignment and read distribution was calculated via MEGAN's use of LCA and GRAMMY (Xia, Cram et al., 2011a). The estimated read abundance data was exported as a table and then normalised against the kit and sequencing negative controls, the remaining data was used for plotting.

2.17.5 Data plotting

Analysis of communities was performed using the *vegan* (Oksanen *et al.* 2015), *phyloseq* (McMurdie & Holmes, 2013), and *stats* packages in R studio (version 3.3.2) (R Core Team, 2014). Communities were normalised prior to comparison by rarefaction, and/or calculation of relative abundance. Taxonomic diversity was calculated by Reciprocal Simpson index and Bray-Curtis dissimilarity was used to compare communities. Total and taxon counts and average counts per taxon between frozen and non-frozen samples were compared by simple z-test (Pocock, 2016). PERMANOVA was used to identify differences in communities between clinical parameters and pairwise PERMANOVA was used to identify individual features responsible for such differences. Results were plotted using *ggbiplot* (Vincent, 2011) and *ggplot2* (Whickham, 2009).

2.18 Statistical analysis

2.18.1 Determining p values

2.18.1.1 Growth study

To determine statistically significant difference between growth rates of the single and double $\phi 24_B$ lysogen compared to MC1061, paired-sample T-tests in the statistical package SPSS was used. The two tailed p values are given at 95% confidence limits.

2.18.1.2 SIC study

The statistically significant difference in SIC between lysogen and MC1061 was calculated using an independent t-test, using the SPSS platform (> 95 % confidence limits).

2.18.1.3 Biolog study

The Biolog area under the respiration curve (AURC) for respiration values were calculated using a trapezoid algorithm. Statistical significance of area under the respiration curve (AURC) and comparison at a specific time point during mid exponential growth phase was achieved by determining normal Gaussian distribution by parametric analysis and statistical significance identified using an un-paired *t* test (> 95% confidence limits).

2.18.1.4 Metabolomic study

Metabolomic analysis was carried out initially by Progenesis QI software (version 2.1), this software provided alignment, peak picking, pairwise statistical analysis and putative metabolite ID based on accurate mass. CRACCD was used for final tabulation of the metabolomic profile. Further multivariate analysis was performed using SIMCA-P and CRACCD. IDs were obtained through the QI plugin 'Progenesis metascope' and filtered through a range of databases using sdf files (ECMDB, HMDB, small molecules drugs, Biomolecules, analgesics mix, Lipid MBD, Basic lipids, and Yeast DB). A paired sample t test was used to determine statistically significant differences between intensities of metabolites identified during metabolomic analyses (> 95% confidence limits).

2.18.1.5 Fatty acid study

A paired sample t test was used to determine statistically significant differences between intensities of fatty acids identified from GC-MS analysis (> 95% confidence limits). Averages and standard error were calculated for the fatty acids identified.

2.18.2 GC-MS Fatty acid methyl ester identification, data gathering and plotting

Peaks from the chromatograms were identified against the standards run using retention time and the mass to charge ratio. The intensities of the given peaks were grouped and tabulated. Basic figures such as histograms, stacked graphs and scatter plots were created in excel. Multivariate analysis was carried out using the online software 'MetaboAnalyst', plots created using this software were: PLS-DA, VIP, boxplot, heatmap, and biplot. All data was normalised via log transformation prior to further plotting. PLS-DA plots display the 95% confidence range and are validated with R2 and Q2 scores using the LOOCV method. VIP plots were derived from the PLS-DA measures and present the variable importance in projection of the primary principle component. The boxplots display the raw tabulated data and the log transformed data, where log transformation was carried out within the program. The heatmap used the euclidean distance measure and the Ward criterion was used for the hierarchical clustering algorithm. The biplot was plotted from the 2 principle components that explained the greatest percent of the data.

Chapter 3. Shigatoxin encoding Bacteriophage ϕ 24B modulates bacterial metabolism to raise antimicrobial tolerance

3.1 Introduction

3.1.1 STEC and Stx-phages

Colonisation by Shiga toxin-encoding *Escherichia coli* (STEC) causes a potentially fatal gastrointestinal infection in humans. There are currently > 500 different characterised STEC serogroups that cause disease including O157:H7 and more recently O104:H4 (Allison, 2007, Muniesa, Hammerl et al., 2012). Stx-phages enter one of two replication pathways, a productive lytic life cycle or a more passive lysogenic cycle where the prophage is replicated by the bacterium as any other genetic loci. The lytic-lysogen decision of lambdoid-like bacteriophages is regulated by early gene expression and sequential binding of proteins to a well characterised genetic switch (Echols & Green, 1971, Reichardt, 1975, Takeda, Matsubara et al., 1975). STEC and Stx-phages are discussed in detail in previous sections 1.1.4 and 1.1.5 respectively.

3.1.2 Phage-bacteria co-evolution

3.1.2.1 Phage accessory genome

The co-evolutionary interaction between a phage and its bacterial host is dynamic, with interplay linked to rounds of inhibition, selection and evolution, often referred to as an ‘arms race’ (Stern & Sorek, 2011). Smith et al. (2007) used a multi-loci PCR typing approach to demonstrate the heterogeneity of Stx-phages. They found that no 2 Stx phage isolates had the same genotype in a regional location (Smith, Wareing et al., 2007b). This diversity is further supported by Bonanno et al. (2016) who identified multiple Stx-phage morphologies not previously reported (Bonanno, Petit et al., 2016). Stx-phages are closely related to bacteriophage lambda, with a comparable genome organisation. In comparison to lambda, Stx-phages carry significantly larger amounts of DNA (~20-25 Kbp) with up to 73 % of the genome and putative coding genes having no known function when analysed at either the nucleotide or protein level (Smith, Rooks et al., 2012b). Nevertheless these genes are well conserved across many Stx-phages and thus likely to be important to the

biology of the phage or its bacterial host (Smith et al., 2012b). Upon phage infection and conversion to a lysogen, genes that are accessory to the core biology of the phage may offer a selective advantage to the host. Mis-excision, mis-packaging of phage DNA (Coren, Pierce et al., 1995) and recombination (Gottesma.Mm, Gottesma.Me et al., 1974) play a large role in phage genome variation. This may leave phage DNA regions, remnant or cryptic prophages that positively impact on the selection and survival of both the phage and the bacterium (Wang, Kim et al., 2010). This is further supported by the common occurrence of prophage regions, usually multiple, in the chromosomes of many bacterial pathogens (Hayashi, Makino et al., 2001).

3.1.2.2 Stx phage and naïve host infectivity, virulence and pathogenicity

There are a number of features of the Shigatoxigenic phage vB_EcoP ϕ 24_B or ϕ 24_B that are particularly relevant to the success and persistence of Stx prophages in *E. coli*. In contrast to the lambda infection model, ϕ 24_B can multiply infect a bacterial host (Allison, Sergeant et al., 2003, Fogg, Allison et al., 2010, Fogg et al., 2011). Stx-phages have been isolated from a wide variety of environments where *E.coli* is present and this has undoubtedly been promoted by the use of an essential outer membrane protein BamA as the adsorption site (Smith, James et al., 2007a). This interaction is conserved in Stx-phages as the incidence of the tail and host recognition protein is widespread (Smith et al., 2007a). ϕ 24_B has also been shown to survive well in compost models (Johannessen, James et al., 2005), also showing infectivity after 30 days in bovine manure and slurry (Nyambe, Burgess et al., 2016). ϕ 24_B is genetically similar to phages isolated from sporadic outbreaks of STEC with high virulence and therefore a good model of the viruses circulating in *E. coli* populations in the environment (Smith et al., 2012b).

The ability of lambdoid like phages to increase virulence by carriage of toxins in their accessory genome (Willshaw, Smith et al., 1985) is well described, e.g. the cholera toxin (CTX) carried by *Vibrio cholera* phage (Davis, Moyer et al., 2000, Sakaguchi, Hayashi et al., 2005). In Stx-phage genomes the shigatoxin genes are always located at the same position on the phage genome, upstream of the Q antiterminator gene therefore organisation and gene location is important (Unkmeir & Schmidt, 2000). It has been hypothesised that presence of *stx* also offers selection and stability for the lysogens (Livny & Friedman, 2004). Colon et al. (2016) observed that Stx-

prophages show greater levels of spontaneous induction than lambda but this more readily correlates to Rec dependent and independent control of the CI repressor protein rather than presence or absence of *stx* (Colon, Chakraborty et al., 2016).

Other accessory genes that are seemingly superfluous to viral replication have been shown to aid microbial selection against environmental stress. Examples include: antibiotic resistance (Colomer-Lluch et al., 2011, McGrath et al., 1999), acid tolerance (Su, Lu et al., 2010, Veses-Garcia, Liu et al., 2015) and polylysogeny (Vostrov et al., 1996). Phage gene expression has also been shown to aid adhesion and colonisation, for example the expression of the λ -encoded *lom* gene promotes adhesion to buccal epithelial cells (Reeve & Shaw, 1979), and the λ -encoded *bor* confers serum resistance (Barondess & Beckwith, 1990). Other phage encoded virulence traits include exotoxin production in *E. coli* (Newland, Strockbine et al., 1985) and increase in bacterial invasion via *Staphylococcus* phage encoded kinase that influences fibrinolysis (Sako, Sawaki et al., 1983). Bacteriophage $\phi 24_B$ has also been shown to encode a mi-RNA in the *lom* region that alters expression of anti-repressor *d-ant* and downstream activity of CI, leading to rapid induction (Nejman-Falenczyk, Bloch et al., 2015). Tree et al. (2014) identified 55 prophage regions encoding small regulatory RNA within the Sakai *E. coli* O157:H7 strain (Tree, Granneman et al., 2014). These small prophage anti-sRNA had the ability to form complexes or mimic core genome regulatory sRNA to aid selective advantage to the bacterial host in bovine rectal mucus. Stx-phage $\phi 24_B$ shows 98% sequence homology at the nucleotide level to the remnant Stx2 phage present in the Sakai genome (Smith et al., 2012b).

The function of the large numbers of hypothetical proteins encoded by $\phi 24_B$ and other converting phage is difficult to determine, and a focus of this study, as gene expression or interaction may be specific to an environment or subject to selective pressure. Therefore, current approaches *in vitro* using synchronous cultures and standard laboratory conditions to investigate the role of these prophages are challenging. The function of these hypothetical gene products and how they impact the host in either an advantageous or deleterious way may be missed. This study/chapter focuses on phage mediated antimicrobial tolerance to antibiotics found in the livestock farm setting which is the primary reservoir of pathogenic shiga toxinogenic *E. coli*.

3.1.3 Aim

The principal aim of this study is to identify how infection and integration of $\phi 24_B$ changes microbial physiology, and how phage conversion aids selection compared to its naïve counterpart. Therefore this study aims to investigate cell proliferation, antimicrobial tolerances, and metabolomic profiles of *E. coli* lysogen $\phi 24_B$ MC1061 and naïve host *E. coli* MC1061. This investigation uses a novel approach to metabolomic study of phage conversion, by monitoring metabolic change under antimicrobial pressures. This chapter uses an untargeted metabolomics approach with the aim of identifying subverted metabolic pathways that occur through conversion by $\phi 24_B$. Such adaptation has been suggested from respiration profiles, where integration of $\phi 24_B$ into its primary integration site located 250 bp upstream of the *IntS* gene (Fogg, Gossage et al., 2007) allows converted *E. coli* MC1061 to grow using alternative sources of phosphate compared to the naïve bacterial host. Respiration data was collected by Dr Darren Smith, and is incorporated into this chapter due to its pertinence and this chapter's publication.

3.2 Results

3.2.1 Effect of $\phi 24_B$ integration on naïve host growth and respiration

3.2.1.1 $\phi 24_B$ integration increases cell proliferation

Growth rates of bacteria can differ due to a range of environmental parameters. To investigate the impact of $\phi 24_B$ on *E. coli*, viable cell counts were determined during growth comparing *E. coli* B strain MC1061 to single and double lysogens (see methods section 2.8), the latter integrated into separate locations in the MC1061 chromosome (Fogg et al., 2007). Under standard growth conditions, the single and double lysogens showed significantly higher early growth compared to the naïve MC1061 (>200 %, Figure 3.1). This alongside a statistically significant increase in doubling time of 18 minutes for the single lysogen compared to 20 minutes for the naïve MC1061 ($p < 0.006$), calculated from each growth curve d (data not shown, $n=9$). As the cultures reached mid to late exponential growth, the differences in growth rates diminished (Figure 3.1). The greatest difference in growth was identified in early growth (Figure 3.1). This was supported by a shorter lag time in the single and double lysogen compared to MC1061 with a 0.5 and 1.8-fold increase respectively in cell number after the first hour of growth. Stationary phase in the double lysogen is achieved earlier compared to MC1061 and the single lysogen.

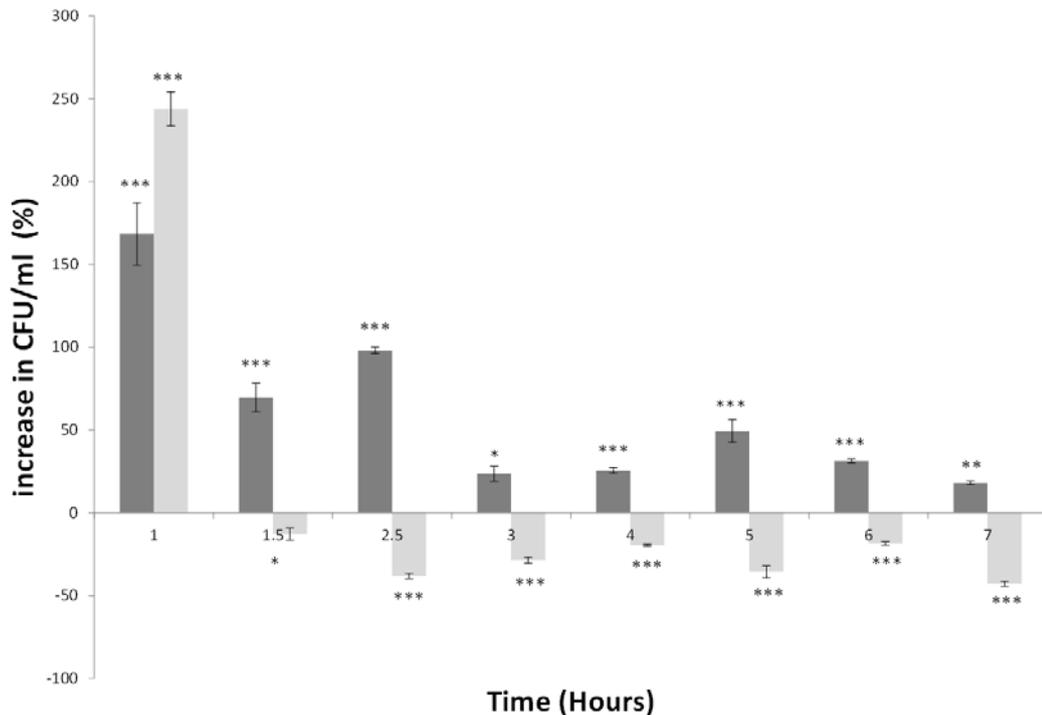


Figure 3.1 Clustered column graph representing percentage increase in cell proliferation of single ($\phi 24B::\Delta Kan$, dark grey) and double ($\phi 24B::\Delta Kan$, $\phi 24B::\Delta Cat$, light grey) MC1061 lysogens. Cultures were grown at 37 °C (CFU.ml) and samples taken over a 7 hour period including experimental and technical replicates (n=9). Percentage increases or decreases show differences in growth of the lysogens compared to the uninfected MC1061 represented here as 0 on the x axis. Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

3.2.1.2 $\phi 24B$ integration alters utilisation of different mono-phosphates and inability to respire using β -D-Allose

To explore the single lysogen related differences in cell respiration through growth the Biolog Phenotype MicroArray was previously carried out by Dr Darren Smith (see methods section 2.9). This determined functional changes in respiration resulting from phage conversion over a 48 h period with recordings taken every 15 min. The lysogen acquired the ability to respire and grow utilising uridine-2-monophosphate (U-2-P) when compared to the naïve MC1061 (Figure 3.2, panel A). Phage mediated subversion of pyrimidine and purine synthesis by lytic phages has been previously reported and will be discussed later. Conversely, integration of the phage inhibited the lysogens ability to use D-Allose for respiration.

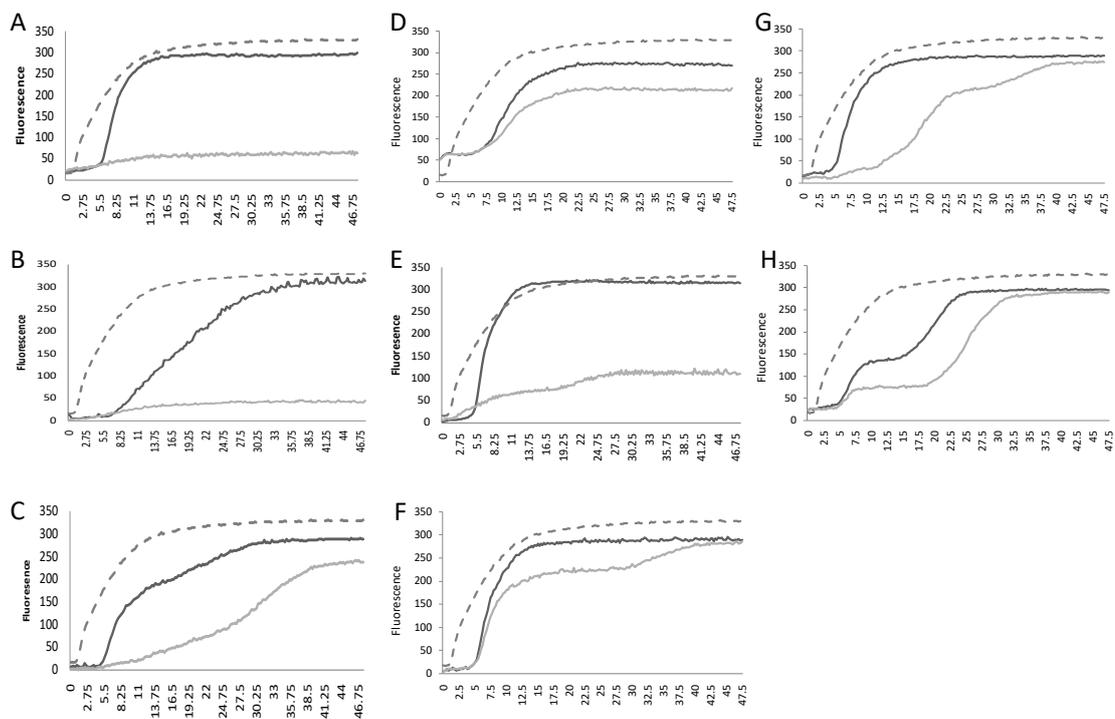


Figure 3.2 Respiration traces from raw Biolog data comparing naïve MC1061 respiration (light grey line) to lysogen (dark grey line), the hashed line represents (n=3) rates of respiration of both naïve MC1061 under standard growth conditions in the absence of challenge. Panel A illustrates the lysogen's ability to now utilise a different phosphate source for respiration. B-H show respiration in the presence of other antimicrobials. Test compounds; A - Uridine 2-monophosphate; B - 8- Hydroxyquinoline; C – Chloroxylenol; D – Cefoxitin; E – Niaproof; F – Cefamandole; G- Amoxicillin; H – Cefmetazole. Statistically significant differences using area under the curve can be found in the appendices Table 10.8).

3.2.1.3 $\phi 24_B$ integration alters resistance to osmotic stress or antimicrobials

The Biolog phenotypic array also determined that the single lysogen is able to tolerate a range of antimicrobial agents that have both extracellular and intracellular targets (Figure 3.2). The respiration curves derived for this experiment are observable in Figure 3.2 and Figure 3.3. Tests showing differences in respiration profile were determined in the presence of 22 antimicrobials and 7 increases in salt concentration (appendices Table 10.8). Of these 29 different tests, the lysogen showed a level of tolerance to 17 antimicrobials (appendices Table 10.8). Data presented in Figure 3.4 (n=3) are comparisons of the area under the respiration curve illustrating those that were altered significantly. $\phi 24_B$ infection promotes tolerance to 8-hydroxyquinoline ($P < 0.000$), chloroxylenol ($P < 0.0037$), and cefmetazole ($P < 0.0026$), cefoxitin, ($P < 0.015$) cefemendole ($P < 0.0239$) and amoxicillin ($P < 0.057$). Integration of $\phi 24_B$ into the primary site 250 bp upstream of *IntS* inhibits respiration utilising B-D-allose. Lysogeny also limits cell respiration in the presence of oxolinic

acid although this is linked to phage induction as the cellular target is DNA gyrase. Inhibition of DNA gyrase has been previously shown to stimulate temperate phages to the lytic life cycle as cellular stress stimulates RecA, *lexA* and proteolytic cleavage of the repressor protein promoting phage induction (Matsushiro, Sato et al., 1999).

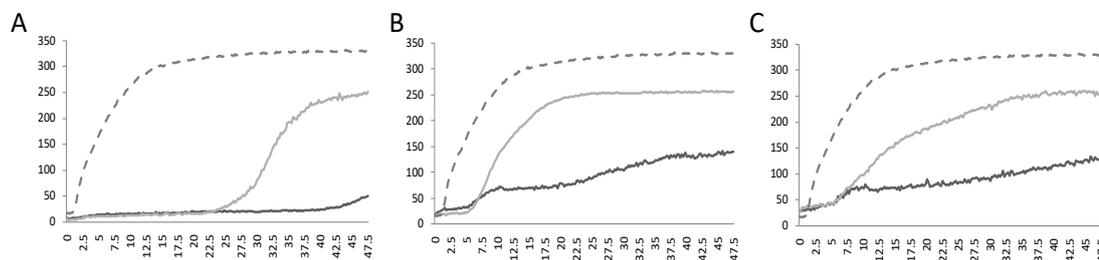


Figure 3.3 Respiration traces from raw Biolog data comparing naïve MC1061 respiration (light grey line) to lysogen (dark grey line), the hashed line represents (n=6) the combine respiration control data of both naïve MC1061 and Lysogen. Test compounds; A - b-D-Allose; B – Ofloxacin; C – Oxolinic acid. Statistical values for each of these individual graphs can be found in appendices Table 10.8. These traces show an inverse response when compared to Figure 3.2, where conversion by $\phi 24_B::Kan$ has a negative effect on the respiration of MC1061.

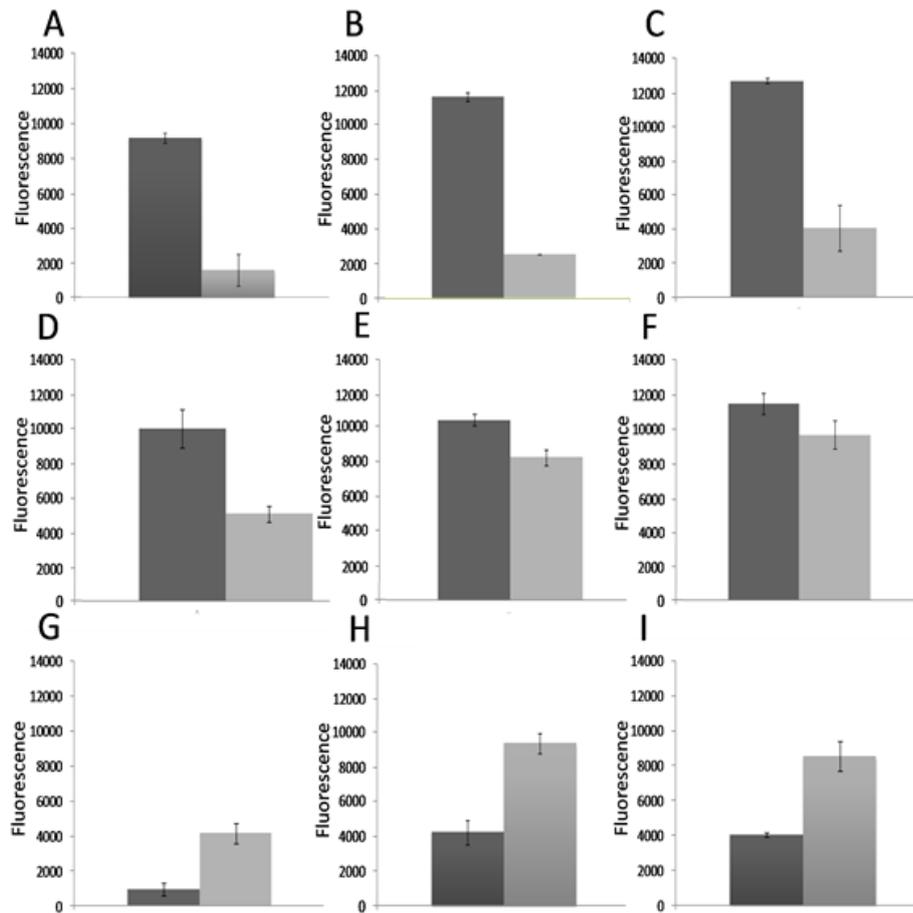


Figure 3.4 A comparison of Area Under the Respiration Curve (AURC) data from the Biolog bacterial phenotypic microarray. Data plotted shows the addition of supplemented nutrients or chemical challenge showed statistically significant difference in rates of respiration between the lysogen and naïve MC1061 host (for *P* values see appendicesTable 10.8). Arbitrary Omnilog fluorescence values (y-axis) show differences between the naïve MC1061 (light grey) host and the lysogen (dark grey) over a 47.5 h time period (n=3). Error bars represent SEM. Graphs A-F show significantly higher amount of respiration of the lysogen compared to the naïve host under the following conditions; (A) U-2-monophosphate, (B) 8-hydroxyquinoline, (C) chloroxylenol, (D) cefoxitin, (E) cefomendole and (F) amoxicillin. Graphs G-I show mean AURC values where growth on different carbon sources or chemical challenge that has a detrimental effect on the respiration of MC1061 when converted by ϕ 24_B, these include; (G) β _D-Allose, (H) ofloxacin and (I) oxolinic acid.

3.2.1.4 ϕ 24B integration increases MC1061 tolerance to sub-inhibitory concentrations of chloroxylenol and 8-hydroxyquinoline

To better understand the level of antimicrobial tolerance of the single lysogen, sub-inhibitory concentrations (SIC) were first determined against both MC1061 and the lysogen that reduce cell growth by ~ 60 % (see methods section 2.5.1). Antimicrobials representing 3 core groups were selected to validate the Biolog data; chloroxylenol (bacteriostatic), oxolinic acid (DNA gyrase inhibitor) and 8-hydroxyquinoline (bactericidal). These compounds were also selected based on p values and differentiation between the lysogen and the naive host. Prior to comparison, an approximate SIC range was determined for MC1061 utilising each of the 3 test drugs. Figure 3.5 illustrates increased tolerance by the lysogen in the presence of chloroxylenol and 8-hydroxyquinoline. Conversely, the naïve host shows increased tolerance compared to the lysogen in the presence of oxolinic acid. This also offers a positive control for the assay as oxolinic acid targets DNA gyrase and therefore stimulates phage induction (26). Phage induction was confirmed by the presence of free phage compared to the un- induced control (data not shown).

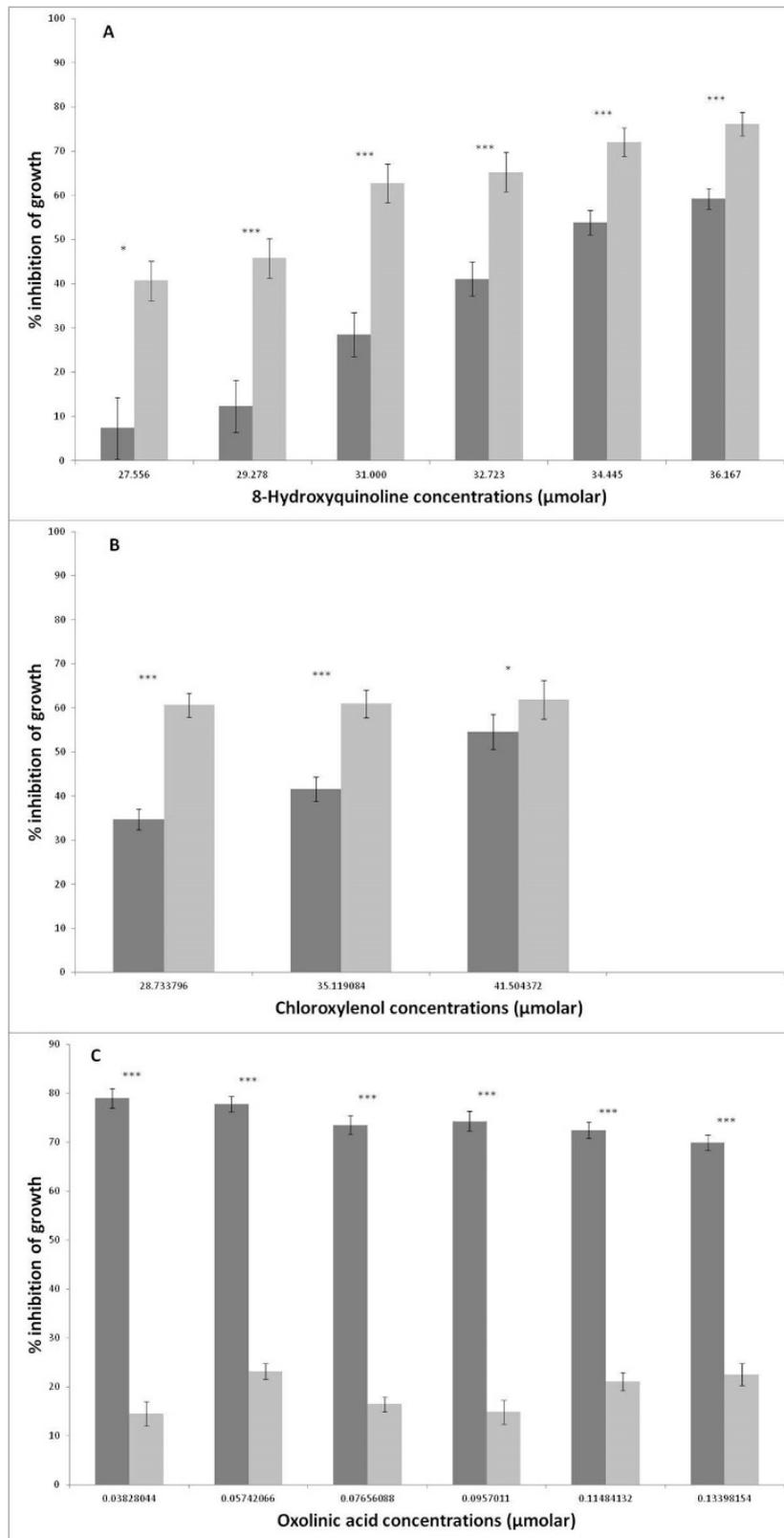


Figure 3.5 Response in growth of both MC1061 (light grey) and the ϕ 24B lysogen (Dark grey) to an increasing concentration of (A) 8-hydroxyquinoline, (B) chloroxylenol, and (C) oxolinic acid. Bacterial growth was measured by increase in optical density at 600nm after 18 hours growth at 37°C, as per original Biolog assay. Error bars represent the standard error of the mean (SEM) (n=12). Significance represented by (P) thresholds; *** <0.001, ** <0.01, * <0.05.

3.2.2 $\phi 24_B$ integration effects naïve host metabolome

3.2.2.1 Metabolic profiles comparing naïve MC1061 to $\phi 24_B$ Lysogen

Untargeted metabolite profiling approach with high resolution LC-MS (≤ 1 ppm mass accuracy in full scan) was used to determine metabolic differences between bacterial host and lysogen during growth and when challenged with a sub-inhibitory concentration of test antibiotic (see methods section 2.6). To broadly compare findings, significant metabolic differences ($p < 0.05$) were observed between both growth phase and antimicrobial challenge. In total, $>11K$ ion features or possible metabolites were determined across all of the different tests performed. Of these 81 showed discrimination between the naïve MC1061 and the $\phi 24_B$ lysogen that had clean chromatogram peaks and $< 5\%$ coefficient variable (CV) (appendices Table 10.6). These 81 metabolites that show differences can be further stratified to each test.

The metabolite data were analysed using supervised and non-supervised methods of multivariate analysis. Principal Component Analysis was first employed to visualise trends in the dataset and identify potential outliers. To further interrogate the data, Partial-Least Squared Discriminant Analysis models (PLS-DA) were generated and score plots are shown in (Figure 3.6 A-C). The PLS-DA models for both hydroxyquinoline and chloroxylenol conditions score plots had good discriminating ability, establishing the metabolic differences between the lysogen and naïve host. During standard growth conditions component 1 failed to discriminate: Q2 -0.556, R2Y 0.262, as R2Y and Q2 < 0.5 , although certain metabolites showed significant differences between the lysogen and MC1061. The 8-hydroxyquinoline component 1: Q2 0.74, R2Y 0.89 and the chloroxylenol component 1: Q2 0.802, R2Y 0.923 were both discriminatory with an R2Y and Q2 > 0.5 . Further model statistics can be found in the appendices Table 10.9. Stx-phage $\phi 24_B$ has been previously shown to undergo spontaneous induction (27) and may impact the metabolite profile through sequestration of host function and movement to lysis. Therefore the metabolite profiles of both the lysogen and MC1061 were compared with a phage inducing agent, oxolinic acid (DNA gyrase inhibitor). No correlation was seen between metabolite profiles of the lysogen or MC1061 when compared to that of the lysogen undergoing induction with oxolinic acid (data not shown).

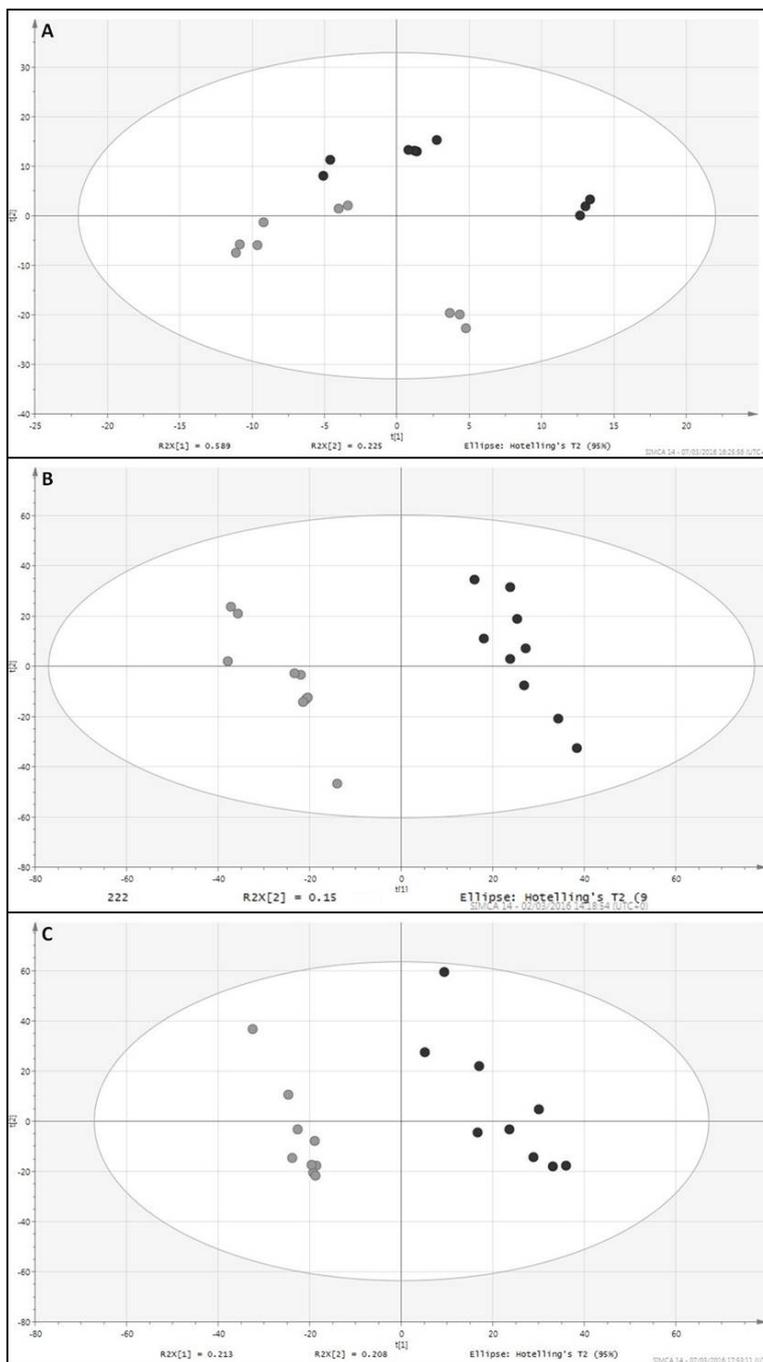


Figure 3.6 A, B and C: The metabolite profiles of MC1061 versus lysogen and multivariate analysis using partial least discriminant analysis (PLS-DA). The panels represent score plots from PLS-DA models of: (A) Standard growth conditions, and supplementation with (B) 8-hydroxyquinoline and (C) chloroxylenol, between the naïve host (light grey spot) and lysogen (dark grey spot), the model discriminatory parameters for the PLS-DA analysis are described in the results section and in appendices Table 10.9.

3.2.2.2 ϕ 24_B integration alters the metabolite profile of MC1061 in standard growth conditions

Out of the 81 discriminatory metabolites determined in this study, only 16 were shown to discriminate between the naïve host and single lysogen under standard culture conditions. Of these 16 metabolites 4 were found in higher levels in the lysogen. This suggests that the lysogen down regulates certain metabolic functions or is directing metabolism along a different pathway. It is likely to support the change in biology reported in this work and increased rates of early growth by the lysogen.

Early growth in the lysogen demonstrates an observable difference in metabolic profile compared to the naïve MC1061. Under standard growth conditions during early growth, 5 metabolites in total were shown to discriminate between the naïve MC1061 and lysogen. Of these, 1 was higher compared to the naïve control (see appendices Figure 10.5). During stationary phase, in standard growth conditions, only 9 metabolites in total showed significant difference and all were found in lower levels in the lysogen (see appendices table Table 10.6).

As phage-mediated metabolic differences are present during standard culture, the differences in metabolite profiles under challenge with sub-inhibitory concentrations of 8-hydroxyquinoline and chloroxylenol were tested (Figure 3.6 B and C). The previous Biolog results showed that the lysogen displays a tolerance to these 2 antibiotics.

3.2.2.3 ϕ 24_B integration alters the metabolite profile of MC1061 during growth under sub-inhibitory concentrations of 8-hydroxyquinoline

Upon treatment with 8-hydroxyquinoline, there were 29 metabolites that showed significant difference between the naïve MC1061 and single lysogen. Of these 29 metabolites, 22 were found in higher levels in the lysogen. Early growth phase in the lysogen demonstrates an observable difference in metabolite profile compared to naïve MC1061. Under 8-hydroxyquinoline stress during early growth, 6 metabolites in total were shown to discriminate between the naïve

MC1061 and lysogen. Of these, 5 were higher compared to the naïve control (see appendices Figure 10.5).

3.2.2.4 ϕ 24_B integration alters the metabolite profile of MC1061 during growth under sub-inhibitory concentrations of chloroxylenol

Under chloroxylenol treatment, 41 metabolites showed significant differences between the naïve MC1061 and lysogen. Of these 41, the lysogen had 22 metabolites with significantly higher levels compared to the naïve host. Early growth phase in the lysogen demonstrates an observable difference in metabolic change compared to the naïve MC1061. Under chloroxylenol stress during early growth, 13 metabolites in total were shown to discriminate between the naïve MC1061 and lysogen. Of these, 9 were higher compared to the naïve control (see appendices Figure 10.5).

3.2.2.5 Alteration in metabolomics profile and antimicrobial tolerance is not linked to kanamycin resistance selective marker used to detoxify ϕ 24_B

The kanamycin gene (*aph3*) used to detoxify the ϕ 24_B phage (Allison et al., 2003) is used as a selective marker only prior to experimentation. The metabolic profiles help confirm that the *aph3* gene is not the cause in antimicrobial tolerance observed. The metabolite profiles are discriminatory to each of the 2 antimicrobials tested with no trend between profiles linking ‘tolerance’ to a common *aph3* associated function.

3.2.2.6 Characterising the metabolites that discriminate between the naïve MC1061 and ϕ 24_B lysogen

The discriminatory metabolites determined from each test were compared with metabolite databases and were putatively identified based on exact mass and empirical formula (see section Table 10.6). The identity of each metabolite was confirmed using fragmentation analysis with a secondary MS/MS stage. Identities with fragment similarity were found for 58 of the 81 metabolites discriminating between the naïve and lysogen. 6 particular metabolites are focused on here, as they have robust identities from fragmentation patterns, retention times, and low accurate mass error (PPM), relating to known curated bacterial metabolites (Table 3.1). The 6 metabolites

are: hexadecanoic acid, 5-Methyluridine, ophthalmic acid, pimelic acid and FAPy-Adenine, with PPM error margins of 0 ± 1 (0.17, -0.64, 0.45, 1.31, 0.56 and -1.00, respectively).

Hexadecanoic acid (palmitic acid) is a fatty acid that is utilised in the construction of lipid A, it's also known to play a role in the biotin pathway. Sphinganine is a putative kinase and also associated to a lipid based response to cellular stress. 5-Methyluridine is a compound in the nucleotide synthesis pathway, specifically a pyrimidine missing its phosphate group. Ophthalmic acid is a glutathione analogue and likewise associated to dealing with oxidative stress. Pimelic acid is the activated form of pimeloyl-CoA, a primary prerequisite for the biotin pathway. Finally, FAPy-Adenine is an oxidised DNA base, which is the result of oxidative stress, and is known to cause damages to cell structure and protein activity.

Table 3.1 Statistics for compound IDs (sustained with reputable MSMS fragmentation) related to known bacterial pathways

m/z	ID	Mass error (ppm)	Adduct	Formula	Anova (p)	Max Fold Change	Up regulated by
174.0396	FAPy-Adenine	0.69	M-H	C ₄ H ₇ N ₄ O ₄	0.0019	1.43	Naïve Host
272.2594	hexadecanoic acid	3.53	M+H	C ₁₆ H ₃₃ NO ₂	0.01	1.08	Lysogen
288.2895	Sphinganine	-0.64	M+H	C ₁₇ H ₃₇ NO ₂	0.0044	1.13	Lysogen
259.0926	5-Methyluridine	0.45	M+H	C ₁₀ H ₁₄ N ₂ O ₆	0.0466	1.06	Naïve Host
289.1277 n	Ophthalmic acid	1.31	M+H or M+Na or M+K	C ₁₁ H ₁₉ N ₃ O ₆	1.2E-06	1.33	Lysogen
178.1075	Pimelic acid	0.56	M+NH ₄	C ₇ H ₁₂ O ₄	0.0277	1.29	Lysogen

The lysogen has significantly higher intensity levels of pimelic acid under all tests. This biotin pathway precursor has consistently higher intensity specifically during early growth (Figure 3.7). It should be noted that the biotin pathway is intrinsically linked to growth. FAPy-Adenine, a bacterial stress marker (Graziewicz, Zastawny et al., 2000), is only seen in stressed conditions in these analyses, with the lysogen expressing significantly lower intensity during early growth and

higher intensity at stationary phase growth (Figure 3.7). Hexadecanoic acid is identified in significantly higher abundance under cellular stress of chloroxylenol, and is further increased in the lysogen during early growth ($P = 0.04$).

Metabolite sphinganine is present under standard conditions in higher intensity in the naïve MC1061. When challenged with chloroxylenol, intensity levels of sphinganine were undetectable in both naïve and lysogen during early growth. During mid-exponential and stationary phase growth under chloroxylenol test there is > 100 fold increase in intensity of sphinganine in both the naïve and lysogen.

5-Methyluridine is present at stationary phase in all conditions, and is also identified in higher intensity when challenged with both antibiotics. Ophthalmic acid was present at all stages of growth under standard conditions where the lysogen shows lower intensity at early and mid-growth, and higher levels at stationary phase. When treated with either antimicrobial agent, ophthalmic acid was only present at stationary growth, with significantly higher intensity found in the lysogen ($P = 0.001$).

During standard culture, there are 16 metabolites responsible for the differences seen between the core metabolic profiles of naïve host and lysogen during the 3 growth phases. Importantly 10 of these, including pimelic acid, are also present when the lysogen is challenged with chloroxylenol and 8-hydroxyquinoline.

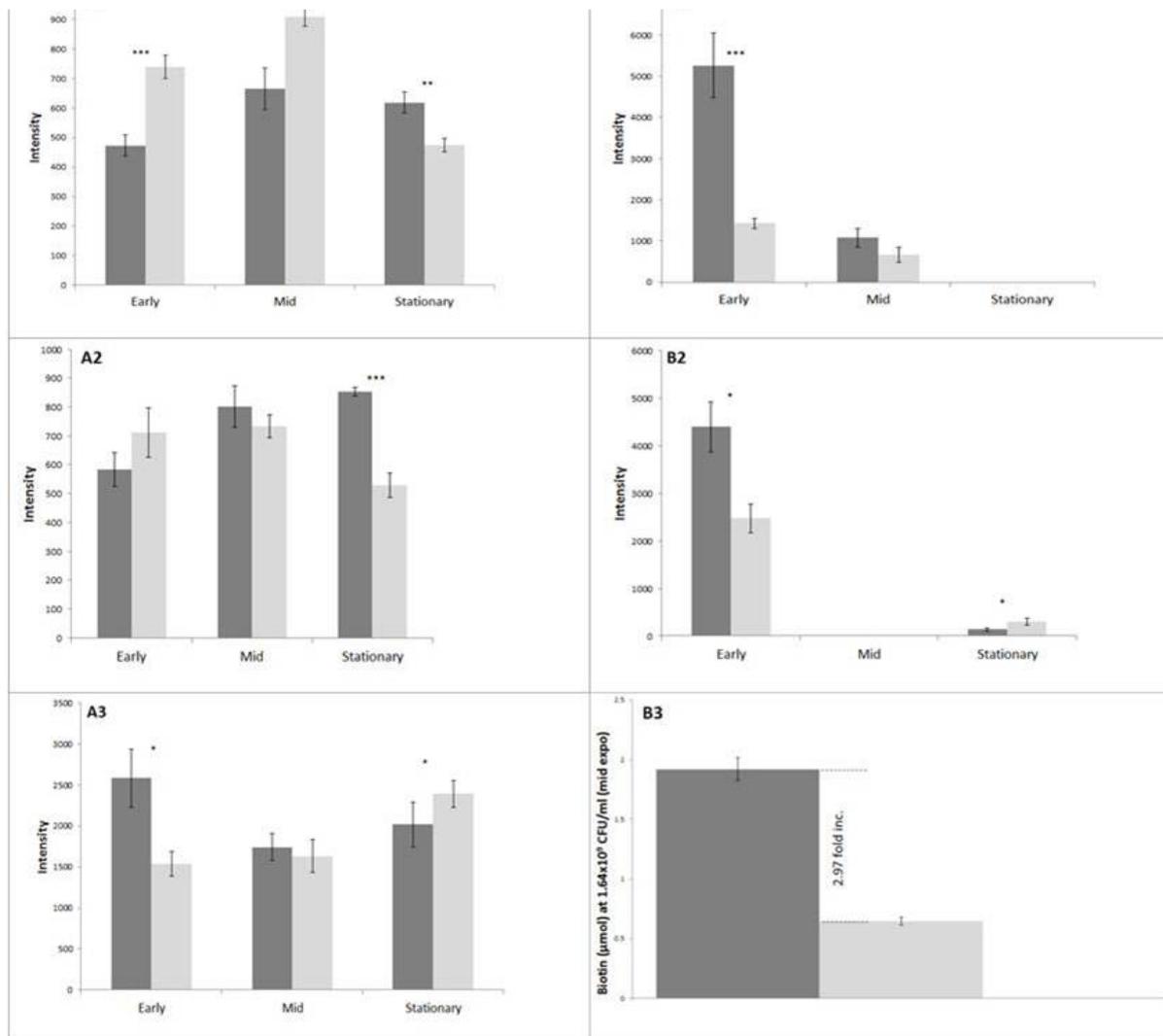


Figure 3.7 Biotin concentration, FAPy-Adenine and pimelic acid intensity showing significant biological differences between naïve host and lysogen during growth and antimicrobial challenge. A1: Changes in cellular stress marker FAPy-Adenine abundances under the challenge of chloroxylenol at early, mid and stationary growth between the lysogen (dark grey) and naïve Host (light grey). B1: Average pimelic acid abundance under chloroxylenol at early, mid and stationary growth between the lysogen and naïve Host. A2: Average FAPy-Adenine abundances under selective pressure of 8-hydroxyquinoline at early, mid and stationary growth between the lysogen and naïve MC1061. B2: Average pimelic acid abundances under challenge with 8-hydroxyquinoline at early, mid and stationary growth between the lysogen and naïve Host. A3: Average pimelic acid abundances under standard conditions at early, mid and stationary growth between the lysogen and naïve Host. B3: Variance in the amounts of Biotin present in samples of Φ24_B lysogen and MC1061 naïve host. Error bars derived from standard error of the mean (n=3). Biotin Quantitation test performed using BioVision® quantitation kit (7.5) using a modified protocol. Two tailed significance represented by *** <0.001, ** <0.01, * <0.05, key: *Inc. = Increase, *expo = exponential growth.

In the absence of antibiotics, the metabolite profile shows less discrimination between the lysogen and host at the 3 stages of growth by PLS-DA (Figure 3.6A). Changes in individual metabolite abundances were measured as before (Figure 3.6 and Figure 3.8), and >100 were deemed possible biologically relevant metabolites. From the confirmed compounds, a total of 16 metabolites (appendices Table 10.6) were shown to discriminate between MC1061 and the $\phi 24_B$ lysogen.

These data were further analysed using Hierarchical cluster analysis (HCA) and Euclidean dissimilarity matrix (DM) to create a heatmap that discriminates between 81 metabolites across all tests in this study (Figure 3.8). The unsupervised heatmap shows that the metabolic profiles have separated by condition.

Figure 3.8 illustrates differences between the metabolic profiles of the naïve MC1061 and $\phi 24_B$ lysogen when comparing both test antimicrobials and the standard culture conditions. Firstly there is the greatest dissimilarity when the naïve host or lysogen has been treated with a sub-inhibitory concentration of chloroxylenol. Within this grouping the naïve host shows the greatest difference in profile at stationary phase for the treatment group. The chloroxylenol group is further stratified by whether the phage is present or absent. Presence of the phage offers the most dissimilar metabolic profile under this antimicrobial challenge. Treatment with 8-hydroxyquinoline has less impact on the metabolic profiles, yet the antimicrobial tolerance is still marked. The difference is also less marked as the profiles are stratified by growth phase rather than presence or absence of phage. Importantly in Figure 3.8 differences between the 81 metabolites in the naïve host and lysogen without challenging with an antimicrobial are still apparent.

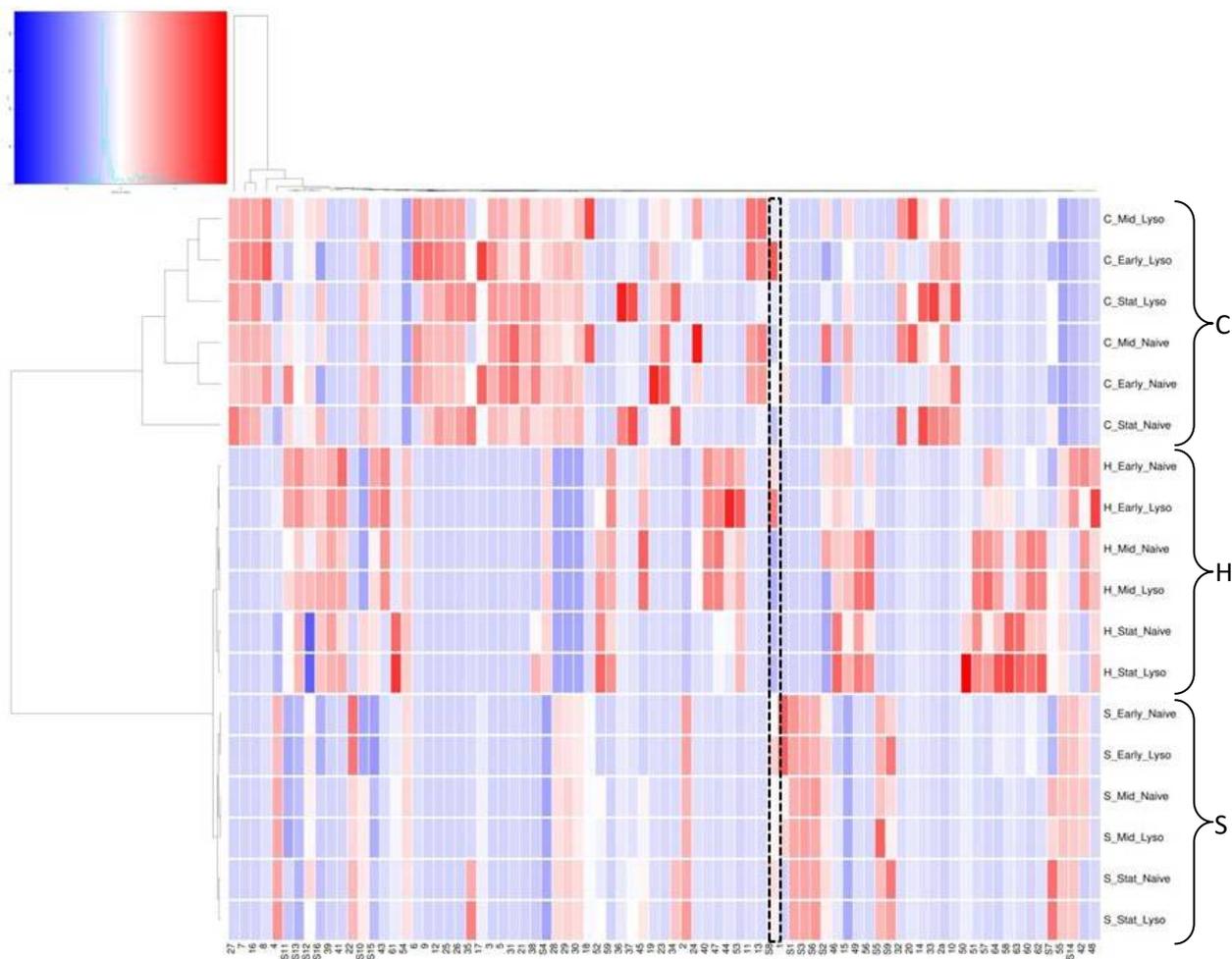


Figure 3.8 Heatmap generated by metabolic levels of 81 metabolites using HCA and DM. Culture conditions and presence or absence of phage can be found alongside each profile (H = 8-hydroxyquinoline, C = chloroxylenol, S = standard). Each individual tile represents a metabolite. The colour of a given tile denotes higher or lower intensity of the metabolite. The colour scale key is: dark blue: lowest levels; white: mid-point; dark red: highest level. The gradient between these colours represents variation in the levels of the metabolite across the colour scale (putative IDs can be found in appendices Table 10.6). Pimelic acid is highlighted across all profiles with a hatched box.

3.3 Discussion

The accessory genome of bacteria promoted through horizontal gene transfer is important in understanding how mobile genetic elements aid selection in the environment. Metagenomics of DNA viruses in environmental and clinical samples has revealed a wide range of antimicrobial resistance genes (ARGs) (Colomer-Lluch, Jofre et al., 2014, Marti, Variatza et al., 2014, Parsley, Consuegra et al., 2010). Enault *et al.*, (2016) demonstrate that caution is needed as ARGs are over-estimated and therefore rarely found in phage genomes and that this over-estimation was further supported by functionality (Enault, Briet et al., 2016). From this data a different mechanism can be proposed; promoted by Stx-phage $\phi 24_B$, through infection and subversion of the cell physiology, promoting tolerance to sub-inhibitory concentrations of antimicrobials 8-hydroxyquinoline and chloroxylenol. Importantly, this data shows that this tolerance is to antimicrobials commonly used in the farming industry globally. The bacteriostatic drug 'chloroxylenol' is widely used in detergent based products, for the direct treatment of livestock e.g. bovine teat treatment (Ascenzi, 1995). Similarly, the bactericidal drug '8-hydroxyquinoline' has a broad antimicrobial activity with agricultural use due to its potency against insects, fungi, and bacteria (Al - Busafi, Suliman et al., 2014).

De Smet et al. (2016) illustrated metabolomic differences during phage infection of *P. Aeruginosa* (De Smet, Zimmermann et al., 2016), whereas this is the first reported use of a metabolic profiling approach to characterise the impact of temperate phage infection on the physiology of the bacteria under antimicrobial pressure. The impact of prophage should not be underestimated as basis for metabolic variation and selection for the bacterial host by heightening or dampening cellular response to stress. With the altered metabolic profile of the $\phi 24_B$ lysogen and increased levels of biotin concentration and fatty acid intensities, it leads to the hypothesis that altered growth and lipids may play a role in altering the cell surface that promotes antimicrobial exclusion.

The metabolite pimelic acid is a precursor for the majority of the carbon atoms of biotin (Lin, Hanson et al., 2010). Biotin plays a crucial role in cell metabolism via carboxylation and decarboxylation reactions. Beyond its function as a cofactor for carboxylases, biotin also plays a

role in gene regulation in mammals (Dakshinamurti, 2005). Unfortunately, the mechanism of its action in *E. coli* is relatively unknown. However, it has been shown that the *E. coli* BioC–BioH pathway uses a methylation and demethylation strategy to complete the necessary pimeloyl moiety (Lin et al., 2010). This methylation approach disguises the biotin synthetic intermediates such that they become substrates for the fatty acid synthetic pathway, and once the pimeloyl moiety is complete it is demethylated (Lin et al., 2010). These data show that the ϕ_{24B} prophage has a significant upregulatory effect on biotin that links to other physiological pathways including fatty acid synthesis (Lin et al., 2010). Differences in pimelic acid intensities between lysogen and naïve host were greatest during early growth (Figure 3.8 1A, 2B, and 3A), which correlates with the differences observed in growth rates during the first 2.5 hours of culture (Figure 3.1). This is associated with a ~3 fold increase in the level of biotin present per cell at mid-exponential growth phase (Figure 3.7 B3), which correlates to metabolite profiling for pimelic acid. This is the first time a phage has been shown to drive the biotin pathway.

The Biolog data confirmed significant differences in rates of respiration between the naïve host and lysogen under different nutrient and chemical challenges. Interestingly the ϕ_{24B} lysogens acquired the ability to respire using Uridine-2-Phosphate (Figure 3.2) where the AURC is illustrated in Figure 3.4 A. U-2-P is involved in cellular metabolism (including biotin metabolism), nucleotide metabolism, pyrimidine metabolism and pyrimidine catabolism (Kanehisa & Goto, 2000). Phages have been shown to subvert purine and pyrimidine synthesis to aid viral construction and proliferation (De Smet et al., 2016, Friedman & Gots, 1953). Lysogen mediated differences encoded by ϕ_{24B} also supports these numerous studies as metabolomics profiling identifies increased pyrimidine catabolism as 5-methyluridine intensity decreases in the lysogen sample. It has also been previously shown through metagenomic analysis that well adapted phages of *Pseudomonas aeruginosa* in the lung carry genes that are involved with purine, pyrimidine and different phosphate utilisation (Tariq, Everest et al., 2015). This is further supported by Chevallereau *et al.* (2016) who show marked changes in RNA metabolism during bacteriophage infection of *P. Aeruginosa* (Chevallereau, Blasdel et al., 2016). Importantly, not only does ϕ_{24B} lysogen show increased pyrimidine utilisation it shows that phages can expand the group of phosphates *E. coli* can use for cell respiration and growth, in this instance U-2-P.

Conversely, in addition to function gain, integration of $\phi 24_B$ into the MC1061 chromosome confers an inability to respire using β -D- allose. The lysogen used in this study has $\phi 24_B$ inserted into the primary integration site on the *E. coli* genome ~ 250 bp downstream of *intS* (Graziewicz et al., 2000). In *E. coli* there are 3 genes, *alsB*, *alsA*, and *alsC*, that are linked to the utilisation of D-allose (Chevallereau et al., 2016), but are disparate (~ 700 Kbp) from any of the 6 integration sites reported by Fogg et al. (2007) (Fogg et al., 2007, Graziewicz et al., 2000). This is significant as it illustrates that integration can yield off target epigenetic effects. This study illustrates that the lysogen associated changes in fatty acid synthesis may exclude D-allose being transported into the bacterial cell, although the mechanism of this restriction is unknown.

Previous research showed infection with λ increased cell growth by the lysogen under cell starvation/supplementation of glucose in a chemostat culture (Edlin, Lin et al., 1977, Edlin, Lin et al., 1975a, Lin, Bitner et al., 1977). This increase in growth rate is also seen with $\phi 24_B$ here. Interestingly a further increase with infection of a secondary, genetically identical phage is seen. The double lysogen is an identical clone to that reported by Fogg et al. (2007), with phage integrating into the secondary integration site (Graziewicz et al., 2000). When monitoring growth the single lysogen confers a doubling time of 17 mins compared to 20 mins for the naïve MC1061. This is also combined with shorter lag phases in both the single and double lysogen. It has been previously shown in many bacteria and yeast that augmenting a growing culture with biotin increases cell growth rates (Porter & Pelczar, 1941, Snell & Mitchell, 1941b, Underkofler, Bantz et al., 1943b, Williams, Eakin et al., 1940).

When stressed with chloroxyleneol, a demonstrated increase in lipid biosynthesis occurs in the lysogen presumably through subversion of the biotin pathway. This is supported through identification of higher intensity levels of hexadecanoic acid in the metabolite data. Hexadecanoic acid is involved in the biosynthesis of lipid A, a core outer cell membrane structure (Hansen-Hagge, Lehmann et al., 1985, Helander, Lindner et al., 1993, Lathe & Lecocq, 1977). Changes in hexadecanoic acid in the lipid A structure of *E. coli* have been previously shown to be associated to mutations in the *firA* gene (Helander et al., 1993, Roy & Coleman, 1994). The *firA* gene is essential for growth and outer membrane synthesis (Vuorio & Vaara, 1992), and has also been shown as

essential for rifampicin resistance associated with certain mutations in the β subunit of the RNA polymerase (Lathe & Lecocq, 1977). This resistance and increase in hexadecanoic acid associated to the *firA* gene, shows that manipulation of this specific fatty acid likely improves antibiotic resistance. It is noteworthy that altering cell wall properties can broadly improve drug resistance (McDonnell & Russell, 1999), and the biotin pathway is intrinsically linked to cell wall synthesis and growth (Lin et al., 2010, Zhang & Rock, 2008b). The ability to adapt fatty acid synthesis is of particular importance in the presence of chloroxylenol, which has a bacteriostatic phenotype potentially derived from its interference in fatty acid synthesis.

The lysogen showed increased tolerance to 8-hydroxyquinoline and chloroxylenol using the Biolog phenotypic array and sub-inhibitory antimicrobial tests. An untargeted metabolomics approach demonstrated that phage conversion offers the bacterial host different metabolic profiles to tolerate the two antimicrobials tested. The tolerance observed also suggests core functional changes allow the cell to resist two disparate antimicrobials. This may suggest that these lysogen associated metabolic differences would likely infer tolerance to other environmental challenges and selective pressures.

Firstly, these data show a metabolic difference in growth under standard culture conditions between the naïve MC1061 and the $\phi 24_B$ lysogen. From 81 metabolites, 16 were discriminatory between the lysogen and MC1061 (appendices Table 10.6). Pimelic acid is present and constitutively raised after infection by $\phi 24_B$. The data also shows a difference between metabolites at the 3 key stages of growth. These again differ between the lysogen and MC1061 (Figure 3.8).

Under treatment with chloroxylenol in early growth, increased intensity of hexadecanoic acid was identified, particularly in the lysogen. The metabolite sphinganine was observed in our data. Sphinganine plays an essential part in the sphingolipid synthesis pathway (Futerman & Riezman, 2005). In both the lysogen and naïve host there is evidence of higher intensities of a sphinganine under chloroxylenol treatment and at stationary growth in standard conditions (Figure 3.8). In *Shigella* species, a pathway associated with mammalian sphingolipid based rafts has been linked to improved binding and mammalian host cell entry (Lafont, Tran Van Nhieu et al., 2002).

Metabolic differences between the lysogen and naïve bacteria are the most disparate when under challenge of a sub-inhibitory concentration of chloroxylenol, illustrated in the PLS-DA plots (Figure 3.6) and heatmap (Figure 3.8). Chloroxylenol is a bactericidal agent and a halophenol that targets microbial membranes (McDonnell & Russell, 1999) with a broad activity as an antimicrobial (Issam Raad, 2004).

Tolerance to antimicrobial 8-hydroxyquinoline is also reported here. Interestingly, compared to stress under chloroxylenol, the 8-hydroxyquinoline tested metabolite profile changes less significantly from standard conditions. Furthermore when treated with 8-hydroxyquinoline, the metabolite profile is less pronounced in the lysogen when compared to the chloroxylenol test. 8-hydroxyquinoline is a lipophilic metal-chelator with intracellular targets (Short, Vargas et al., 2006). It inhibits growth by chelating metal ions, e.g. Zn^{2+} on RNA polymerase (Collins, Alder et al., 1979, Fraser & Creanor, 1975). The changes in the intensity of lipids present at the cell surface, that are suggested earlier to effect uptake of D-allose, are similarly likely to inhibit levels of these 2 antimicrobials entering the cell.

When testing cellular stress it is imperative to find markers of inhibition detailed in the metabolite data. The metabolomic profiles identified 2 discriminatory metabolites that are associated with cellular stress: FAPy-Adenine (Marti et al., 2014) and ophthalmic acid (Desnues, Cuny et al., 2003, Soga, Baran et al., 2006). Our data showed that FAPy-Adenine was only present when cells were challenged by the antimicrobials 8-hydroxyquinoline and chloroxylenol. Interestingly the intensity levels of FAPy-adenine differ greatly depending on the antimicrobial used and also presence or absence of integrated $\phi 24_B$ (Figure 3.7). In the presence of chloroxylenol the lysogen demonstrates lower intensities of the stress marker FAPy-adenine, 0.56 and 0.37 fold less in early and mid-exponential growth phase respectively (Figure 3.7 2a & b). It also shows higher intensity of pimelic acid compared to MC1061. This strengthens the hypothesis of a biotin related lipid increase or change at the cell surface lowering levels of the drug reaching its intracellular target.

When challenging the culture with 8-hydroxyquinoline, FAPy-adenine intensity increases rapidly, even more so than the naïve host (Figure 3.7). Again there is an increase in pimelic acid

intensity that is ubiquitous to the metabolic profiles in the presence of an integrated $\phi 24_B$, however hexadecanoic acid was undetectable within the cell, which may be associated with the lipophilic nature of the drug. The stress response occurs directly after treating with 8-hydroxyquinoline and likely promotes some cell death. Extracellular lipids released through cell lysis binds the drug, forming a matrix. This therefore would reduce the concentration of the available drug present allowing the bacterial culture to grow.

The second stress marker ophthalmic acid is an analogue of glutathione and a reported marker of oxidative stress (Desnues et al., 2003, Soga et al., 2006). Ophthalmic acid intensity mirrored the stationary phase levels of FAPy-adenine, across all tests, however it was also present in standard culture conditions in both the early and mid-exponential growth phase cultures. This observation from our metabolomic analysis implies higher oxidative stress in the lysogen at stationary growth, as the data is supported by Desnues et al. (2003) (Desnues et al., 2003). The oxidative stress also correlates with the reduction in growth rate and a reduction in the intensity of pimelic acid.

This study has established that Stx-phage $\phi 24_B$ provides a 'jump start' in early respiration and increased bacterial growth rates. These phage-mediated alterations in bacterial host metabolic profile may offer positive selection for the lysogen. Subversion of the biotin pathway is core to the changes mediated by $\phi 24_B$ as it links to the bacterium becoming able to tolerate chloroxylenol and 8-hydroxyquinoline during early and mid-exponential growth phase. These tolerances are important as both antimicrobials are used globally in livestock farming. Importantly metabolic shift and subversion offers two mechanisms of controlling this antimicrobial tolerance through increased biotin and fatty acid synthesis. With treatment and tolerance to chloroxylenol, alteration in levels of lipid A and presumably other lipids enables exclusion of the drug from entry. Secondly, 8-hydroxyquinoline treatment drives early cellular stress, cell death and lysis, which increases extracellular lipids that bind free drug, allowing the community to continue to grow.

The mechanism for this is unclear, yet this is importantly linked to subversion of the bacterial cell because no phage-encoded metabolites are present. It is therefore fair to propose that

temperate phages may not carry ARGs but play a larger role subverting metabolic regulation that alters bacterial sensitivity to antimicrobials.

Chapter 4. STX-phage ϕ 24_B hi-jacks microbial cell wall fatty acid synthesis and increases the rate of acquired resistance

4.1 Introduction

Section 1.1 discusses the pathogenesis of Shigatoxigenic *E. coli* infection and how lambdoid-like Stx-bacteriophages encode and horizontally disseminate the Shigatoxin gene that is involved in the pathogenicity of Shigatoxigenic *Escherichia coli* (STEC). The co-evolutionary interaction between a phage and its host is often referred to as an ‘arms race’ (Stern & Sorek, 2011). Phage-host evolution is arguably one of the most fitting models for the Red Queen hypothesis proposed by Leigh Van Valen in 1973 (Valen, 1977), due to the observed continuity of adaptation and counter-adaptation. Evolutionary selection means that only successful mutations or gene accural that leads to adaptation are retained within the phage gene pool, and these can ultimately become co-selected by the host bacteria. On mutation to the integrated prophage that inhibits any future induction (cryptic phage) (Wang et al., 2010), over time only the prophage regions that confer an advantage to the bacterium remain (Saile, Voigt et al., 2016). These are found in abundance in all bacterial chromosomes. Alongside the Stx gene these phages can also disseminate accessory genes with no assigned function that are well conserved throughout sequenced Stx-phages and thus should be important biologically. A good marker of adaptation and evolution is phage genome size as ϕ 24_B has an increased genome size compared to their related Lambda phage ancestor and can carry > 33% extra DNA, where a high proportion of genes located in these regions have no assigned function. Without associated function we miss the intricacies these viral entities disseminate and how they impact the host in either an advantageous or deleterious way. Smith et al. (2012) also reports that 77 of the genes it carries currently have no associated function (Smith et al., 2012b), see Figure 4.1.

4.1.1 Antimicrobial resistance

The development of antimicrobial resistance is a global problem, and has understandably driven significant recent research. The last decade alone accounts for approximately 72% of all publications associated to antibiotic resistance (based off the PubMed publication database 2018), demonstrating this urgent crisis is at the forefront of scientific research.

There are several effective adaptations for broad range resistance against antimicrobials, these include efflux pumps (Webber & Piddock, 2003), porins (De, Basle et al., 2001, Olesky, Zhao et al., 2006), and antimicrobial inactivators (Shaikh, Fatima et al., 2015). One of the core changes to improve broad range resistance is adaption in the cell wall (Nikaido, 2001), as it can provide non-target specific defence mechanisms, such as thickening of the peptidoglycan layer (Hiramatsu, 2001). The cell wall is a vital structure, and bacteria often adapt it to their environment by fluctuating the fatty acid content including; thermal inactivation, lipid oxidation, acidity regulation, cell size, cell replication, cell division, membrane compound translocation (Annous, Kozempel et al., 1999). Any changes at the cell wall can have significant effects in transport of compounds and antimicrobial resistance/tolerance.

Research into antibiotic resistance has thus far been very bacteria focussed. Their phage counterparts have been massively understudied in their role in the occurrence of resistance, being associated to just ~3% of antibiotic resistance publications (based off the PubMed publication database 2018). The development of antimicrobial resistance through evolutionary interaction with bacteriophages is an specifically and understudied area of research, particularly when compared to the many studies attempting to identify bacterial mechanisms or carriage of specific antimicrobial resistance genes (Blair, Webber et al., 2015, Gibson, Forsberg et al., 2015, Lin, Nishino et al., 2015).

4.1.2 Roles of phage-host interactions in antimicrobial resistance

Phage infection and carriage as a prophage has been previously shown to protect STEC against environmental stress (Colomer-Lluch et al., 2011, McGrath et al., 1999, Veses-Garcia et al., 2015), the most widely reported is linked to the phage *lom* (Barondess & Beckwith, 1990) and *bor* genes (Bik et al., 1995), by phage conversion (horizontal gene transfer (HGT)) (Vostrov et al., 1996). HGT is the movement of genetic material between organisms, where under selective evolution it drives the spread of antibiotic resistance genes (Palmer, Kos et al., 2010). There are three genetic mechanisms in which HGT occurs, these are; transformation, conjugation and transduction. Transformation is the process in which bacteria take up DNA from their environment (Lorenz & Wackernagel, 1994), conjugation is the act of directly transferring genes from one bacterial cell to another (Llosa, Gomis-Ruth et al., 2002), and transduction is the movement of genetic material between bacterial cells via phage (Thomason, Costantino et al., 2007). See section 3.1.2 for more detail on the phage-host co-evolution paradigm.

The transfer of genetic material between bacterial cells via HGT is an important aspect of antimicrobial resistance, where resistant gene carriage is performed by plasmids or phage. The impact of phage gene carriage extends further than the bacterial kingdom, as phage are adept in carrying ecologically important traits, such as defence against parasitoids or toxigenic substances, within and amongst symbiont and animal host lineages (Oliver et al., 2009). For example the toxin-encoding bacteriophage APSE can encode either tyrosine-aspartic acid repeat (YD-repeat)-containing protein or a homolog of cytolethal distending toxin (*cdtB*) for the bacterium *Hamiltonella defensa* (Oliver et al., 2009). These phage encoded toxins kill developing wasp larvae that parasitise the bacterium's host aphid *Acyrtosiphon pisum* (Oliver et al., 2009). Thus the extent of phage interaction with bacteria can be extensive.

As discussed, phage integration can provide beneficial attributes to host bacterium, aiding in outmatching competitors, in maintaining its ecological niche, or in promoting survival in hostile environments. Chapter 3 identified several features associated to improved fitness of *E. coli* upon phage $\phi 24_B$ integration. *E. coli* is a common gut bacteria, with both commensal and pathogenic

strains. The gut plays an important role in human health (see section 1.6), and for this reason the following chapter further investigates the phage-host interaction of the gut associated microbe.

4.1.3 Aims

Chapter 3 identified increased growth rate, antimicrobial tolerance and metabolic change in the lysogen $\phi 24_B$ MC1061 compared to the naïve host. This study aims to classify the extent of improved $\phi 24_B$ lysogen growth under a range of *e. coli* relevant conditions, both environmental and clinical. The chapter also aims to identify possible mechanisms of antimicrobial tolerance by investigating cell wall fatty acid content between the lysogen and naïve host. Finally this chapter aims to support any identified changes observed thus far by mining transcriptomic data for associated pathways, such as gene expression of biotin, fatty acid, lipid and peptidoglycan pathways. Most importantly, the study aims to identify the extent of antimicrobial tolerance provided by $\phi 24_B$, and whether it can influence the emergence of antimicrobial resistance.

4.2 Results

4.2.1 Environmental impact on lysogen and naïve host growth

Chapter 3 illustrates that increased bacterial growth rate influences bacterial survival, not just in the presence of antimicrobial challenge. In chapter 3 we identify $\phi 24_B$ infection is responsible for this trait. To investigate the extent of $\phi 24_B$ influence on growth of the lysogen, clinically and environmentally relevant growth conditions were used as selective pressures on both the single and double lysogen (Figure 4.2). Aerobic conditions were monitored at temperatures of 19, 37 and 42 °C, in relation to the outdoor environment, human core temperature, and bovine core temperature respectively. Anaerobic conditions were monitored to establish how oxygen availability might affect the lysogens improved growth as *E. coli* is a facultative anaerobe, and the gastrointestinal tract of both humans and cattle will be oxygen limited. Figure 4.2 shows, under all 3 temperatures tested, that the lysogen growth rate significantly increases, with the greatest differences observed at 19 °C (>600%) and the least difference seen at 42 °C (>50% Inhibition). Aerobic and anaerobic conditions have little effect on lysogen growth in comparison to the uninfected bacterium. The trend for increased growth at early growth phase is consistent at both 37 and 19 °C, which are two conditions that relate to human core and environmental conditions respectively.

Cell growth rate was also monitored at temperatures 37 °C, 42 °C and 19 °C with the addition of bile salts. Bile salts were added at a concentration similar to Macconkey media, 1.5g/l (Macconkey, 1905, MacConkey, 1908), that would be selective for enteric bacteria. Figure 4.3 shows that under all temperatures tested the lysogen still has significantly increased growth, with the greatest differences observed at 19 °C (>400%) and the least difference seen at 37 °C (>20% Inhibition). The trend for increased growth at early growth phase remains consistent at both 37 °C and 19 °C.

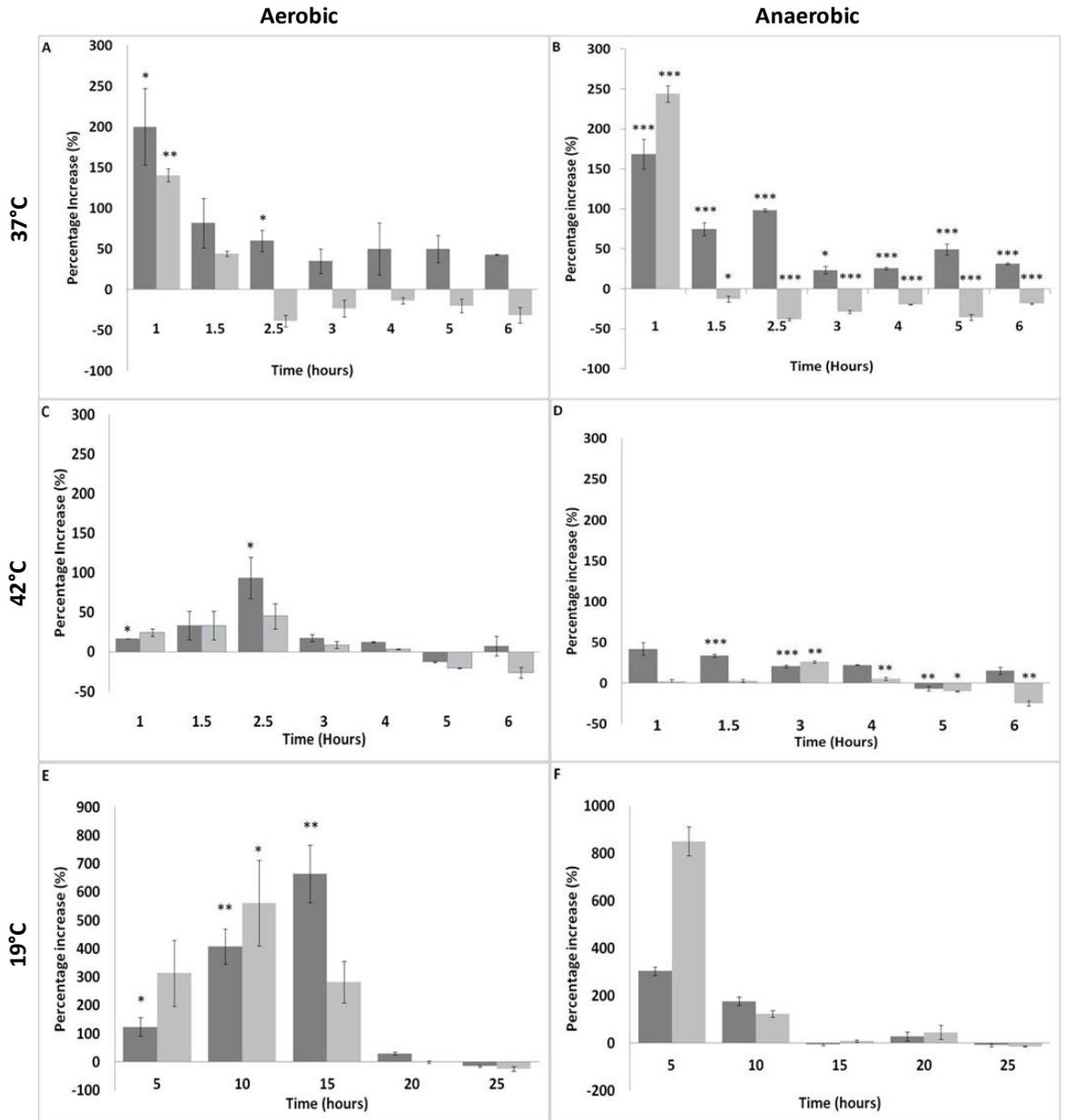


Figure 4.2 Clustered column graph representing percentage increase in CFU of single ($\phi 24B::\Delta Kan$, dark grey) and double ($\phi 24B::\Delta Kan$, $\phi 24B::\Delta Cat$, light grey) MC1061 lysogens. Cultures were grown aerobically at 37 °C (A), 42 °C (C), and 19 °C (E) and anaerobically at 37 °C (CFU.ml) (B), 42 °C (D), and 19 °C (F). At 37 °C, 42 °C samples were taken over a 7 hour period including experimental and technical replicates (n=9). At 19 °C samples were taken over a 40 hour period including experimental and technical replicates (n=9). Percentage increases or decreases show differences in growth of the lysogens compared to the uninfected MC1061 represented here as 0 on the x axis. Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

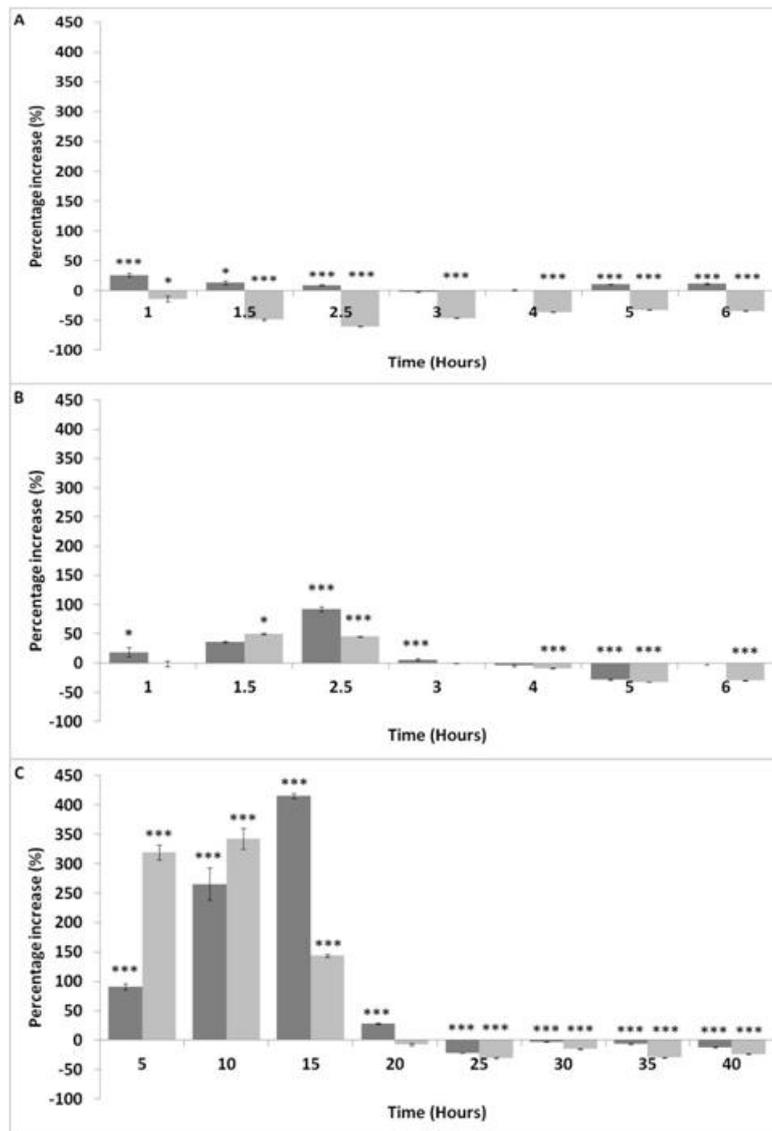


Figure 4.3 Clustered column graph representing percentage increase in CFU of single ($\phi 24B::\Delta Kan$, dark grey) and double ($\phi 24B::\Delta Kan$, $\phi 24B::\Delta Cat$, light grey) MC1061 lysogens, under bile salt conditions. Cultures were grown aerobically for each temperature tested. Samples were taken over a 7 hour period including experimental and technical replicates (n=9) for 37 °C (A) and 42 °C (B). At 19 °C samples (C) were taken over a 40 hour period including experimental and technical replicates (n=9). Percentage increases or decreases show differences in growth of the lysogens compared to the uninfected MC1061 represented here as 0 on the x axis. Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

4.2.1.1 Transcriptomics: Datamining biotin gene expression

Dr Heather Alison's group in Liverpool published a paper illustrating transcriptomic data for both MC1061 and $\phi 24_B$ lysogen under standard and norafloxacin induced conditions (Veses-Garcia et al., 2015). They focused on the top significant changes in gene expression, and thus much of the data is still open for comparison to our work on AMR and FA synthesis. We here further investigate this data by datamining gene expression with focus on the; biotin, fatty acid, lipid, and peptidoglycan pathways. This data would help identify and support our current findings.

Chapter 3 identified biotin as a lysogen driven metabolite, with links to its potential role in increased growth rate. This chapter thus far has identified the lysogens rapid adaptation to environmental changes that would offer the bacterium selection under these conditions over the uninfected bacterium. Figure 4.4 shows all of the genes that were identifiable in the biotin pathway from the transcriptomic data. Eight of the ten genes show an increase in expression in the lysogen, and two of the genes demonstrate a decrease in expression, this supports our hypothesis that the lysogen is manipulating the biotin pathway. The role of the genes in the pathway can be observed in Figure 4.5. This highlights gene expression observed and further supplements the data obtained in chapter 3. The KEGG pathway presented in Figure 4.4 also shows the intrinsic relation between the biotin pathway and fatty acid biosynthesis.

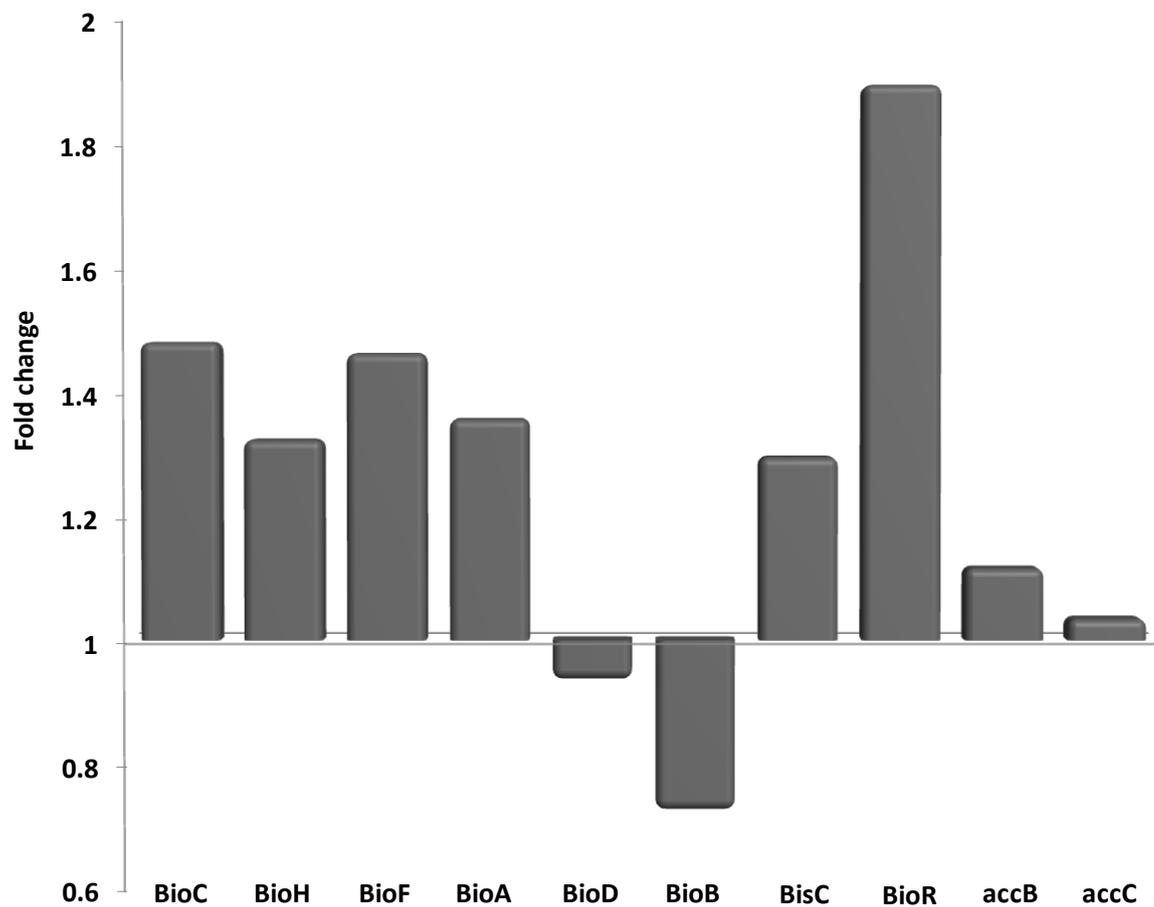


Figure 4.4 Clustered column graph representing $\phi 24B::\Delta Kan$ MC1061 lysogens fold change in biotin gene expression compared to the naïve host MC1061. Order of genes is associated to general or of expression in fatty acid biosynthesis. Samples were taken at mid exponential growth phase.

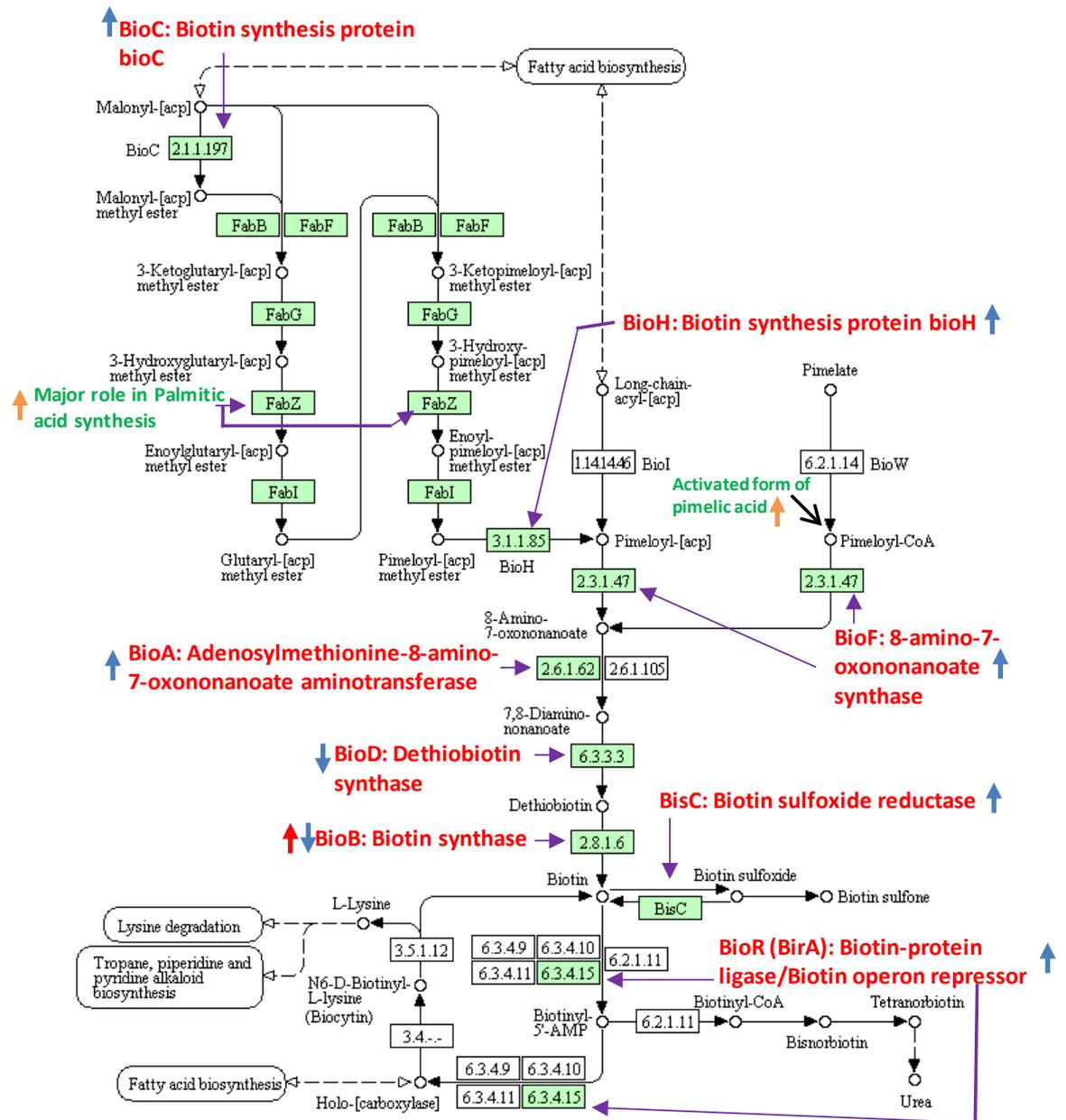


Figure 4.5 Data gathered mapped to the biotin pathway. Red text identifies the genes within the Bio-operon and the associated biotin regulator/biotinylation catalyst. Green text signifies factors in fatty acid synthesis and biotin precursor. Blue, red and orange arrows signify the lysogen associated increase or decrease in expression, concentration and intensity respectively, this is in relation to data gathered by transcriptomics, biotin kit quantification and metabolomic analysis respectively. Gene rectangles coloured in green are Biotin pathway genes associated to k12 *E. coli*.

4.2.2 Cell wall fatty acid profiling of lysogen vs uninfected bacterium

4.2.2.1 Fatty acid (FA) methyl ester profile at 6 hours under different growth conditions

This chapter illustrates subversion of the Biotin pathway that is intrinsically linked to FA synthesis. As we illustrate in Chapter 3 an increase in tolerance to antimicrobials by the $\phi 24_B$ lysogen, we herein target FA content of the cell wall in relation to antimicrobial challenge. This investigation aimed to determine whether an alteration at the cell wall, driven by this biotin pathway phage subversion, stimulates this ability to tolerate an increasing level of antimicrobial. In accordance to method section 2.12 both the lysogen and naïve host test groups were grown separately in 50 μm of chloroxylenol (bacteriostatic) and 50 μm of 8-hydroxyquinoline (bactericidal). The fatty acids were extracted at reaching stationary growth. Standards of each fatty acid were run and sample chromatograms were compared against the standards to validate peaks. In the absence of any antimicrobial (standard conditions) no quantified significant difference was observed between the naïve host and $\phi 24_B$ lysogen, though the overall pattern demonstrates a lower average fatty acid intensity in the lysogen (Figure 4.6). Growth under standard conditions including the presence of 8-hydroxyquinoline shows a complete reversal in fatty acid profile, as the naïve host drops its average fatty acid levels (Figure 4.7). Fatty acids dodecanoic and cyclopropanoic became significantly greater in the lysogen compared to the naïve host. In chloroxylenol conditions the fatty acid profile adapts differently. Of particular interest, when considering chloroxylenol is a bacteriostatic drug, is that no intensity from the chromatograms is matched to heptadecanoic, a key fatty acid in bacterial growth (noted in red in Figure 4.8). The identified fatty acids under chloroxylenol challenge increase in intensity, on average, by a factor of 10 (Figure 4.8). Conversely to 8-hydroxyquinoline, in the presence of chloroxylenol, the lysogen has an average decrease in fatty acids in comparison to the naïve host.

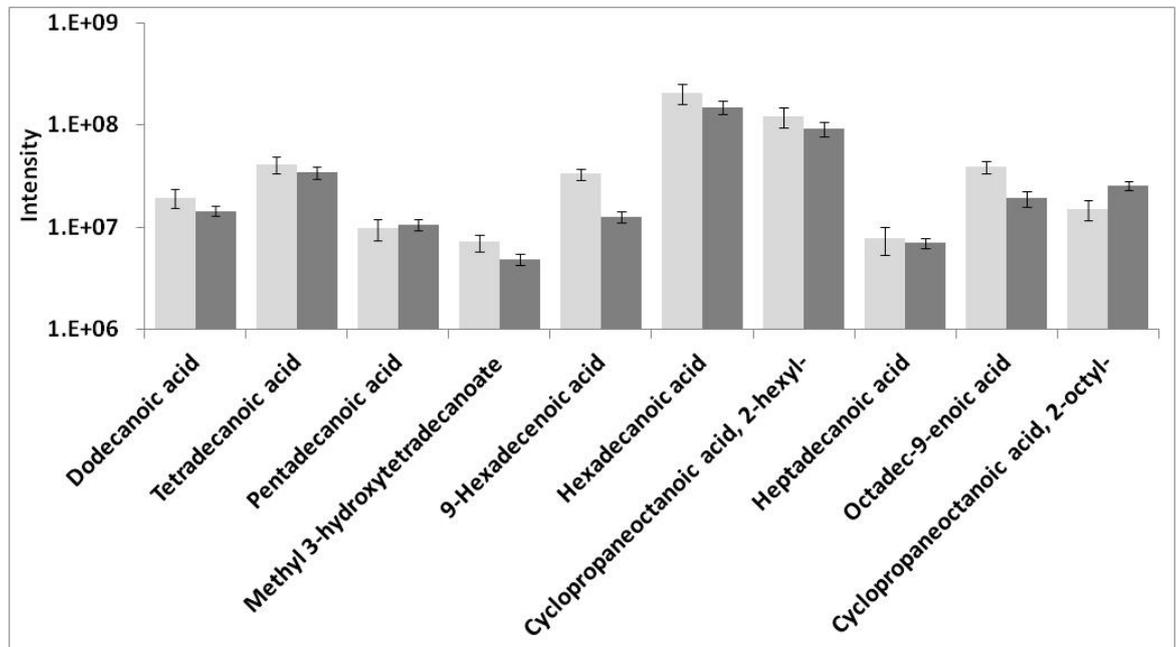


Figure 4.6 Fatty acid methyl ester profile at 6 hr under standard growing conditions. Column graph representing the intensity of fatty acids in the cell wall of both the lysogen $\phi 24_B::\Delta Kan$, dark grey and naïve host MC1061, light grey, under standard growth conditions. Samples were taken at 6 hours including experimental and technical replicates (n=9). Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

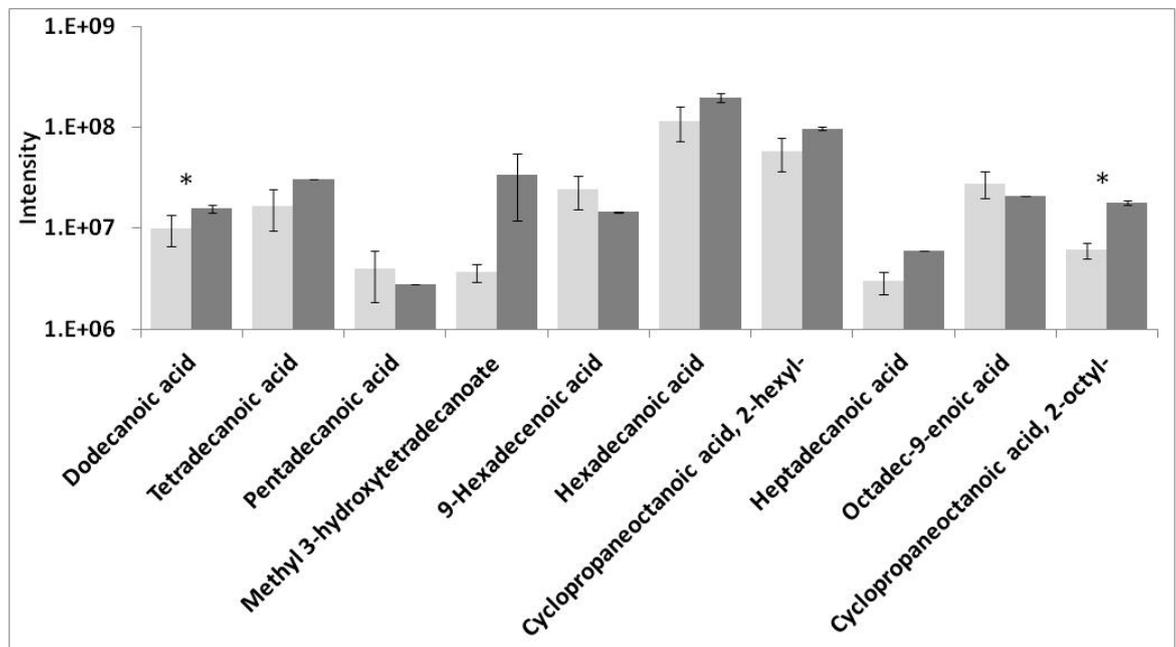


Figure 4.7 Fatty acid methyl ester profile at 6 hr grown in presence of 50 μmol 8-Hydroxyquinoline (bactericidal drug). Column graph representing the intensity of fatty acids in the cell wall of both the lysogen $\phi 24_B::\Delta Kan$, dark grey and naïve host MC1061, light grey, grown in presence of 50 μmol 8-hydroxyquinoline. Samples were taken at 6 hours including experimental and technical replicates (n=9). Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

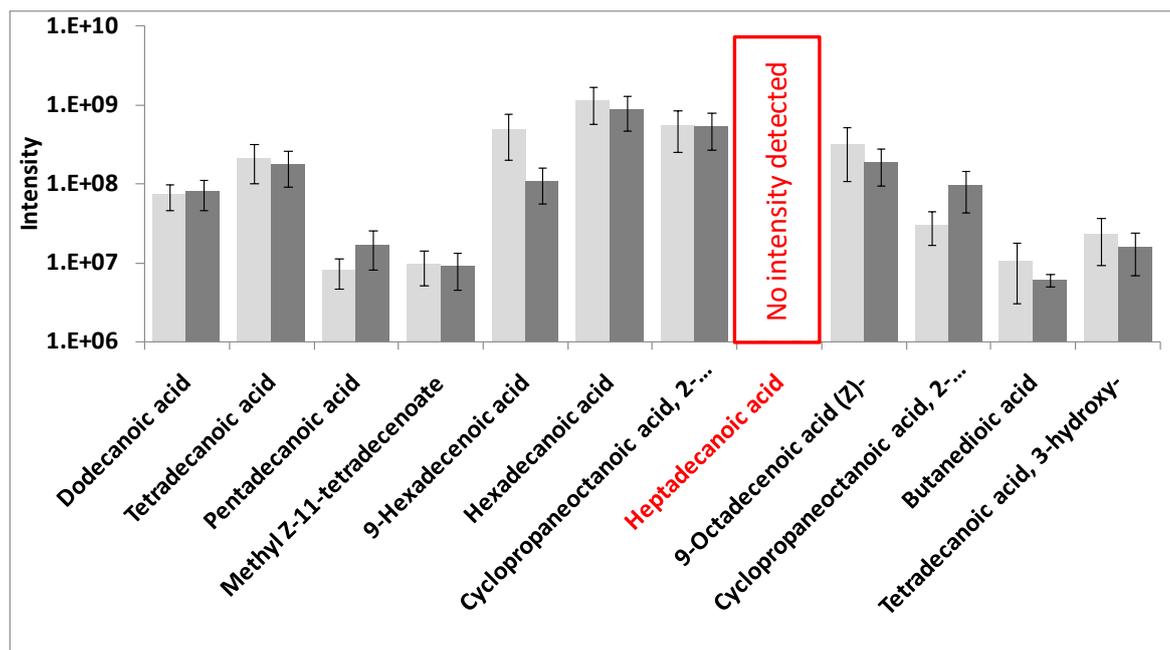


Figure 4.8 Fatty acid methyl ester profile at 6 hr grown in presence of 50 μ mol Chloroxylenol (bacteriostatic drug). Column graph representing the intensity of fatty acids in the cell wall of both the lysogen $\phi 24_B::\Delta Kan$, dark grey and naïve host MC1061, light grey, grown in presence of 50 μ mol chloroxylenol. Samples were taken at 6 hours including experimental and technical replicates (n=9). Significance threshold *P* values *** <0.001, ** <0.01, * <0.05, significance below the x axis demonstrates greater growth from the Naïve host.

4.2.2.2 Transcriptomics: Investigating fatty acid expression

Figure 4.6-4.8 identified some significant changes in the lysogens cell wall fatty acid content compared to the naïve host. Under standard conditions the average lysogens cell wall fatty acid content was less than the naïve host. Figure 4.9 shows 12 genes from the transcriptomic data that can be identified with the fatty acid pathway. All genes associated to fatty acid synthesis and SCFA transport are increased in expression in the lysogen, this is the reverse of what is observed in the cell wall fatty acids. This supports the current hypothesis that the lysogen is redirecting resources, to alternative pathways under standard growth conditions. The role of the genes in the fatty acid pathway can be observed in Figure 4.10. This figure highlights the genes expressed, and secondary highlighting, associated to Figure 4.6 data, has been used to supplement the figure

further. Figure 4.10 also shows the intrinsic relationship between fatty acid and lipid biosynthesis, where gene expression of lipid precursor is shown to decrease in lysogeny.

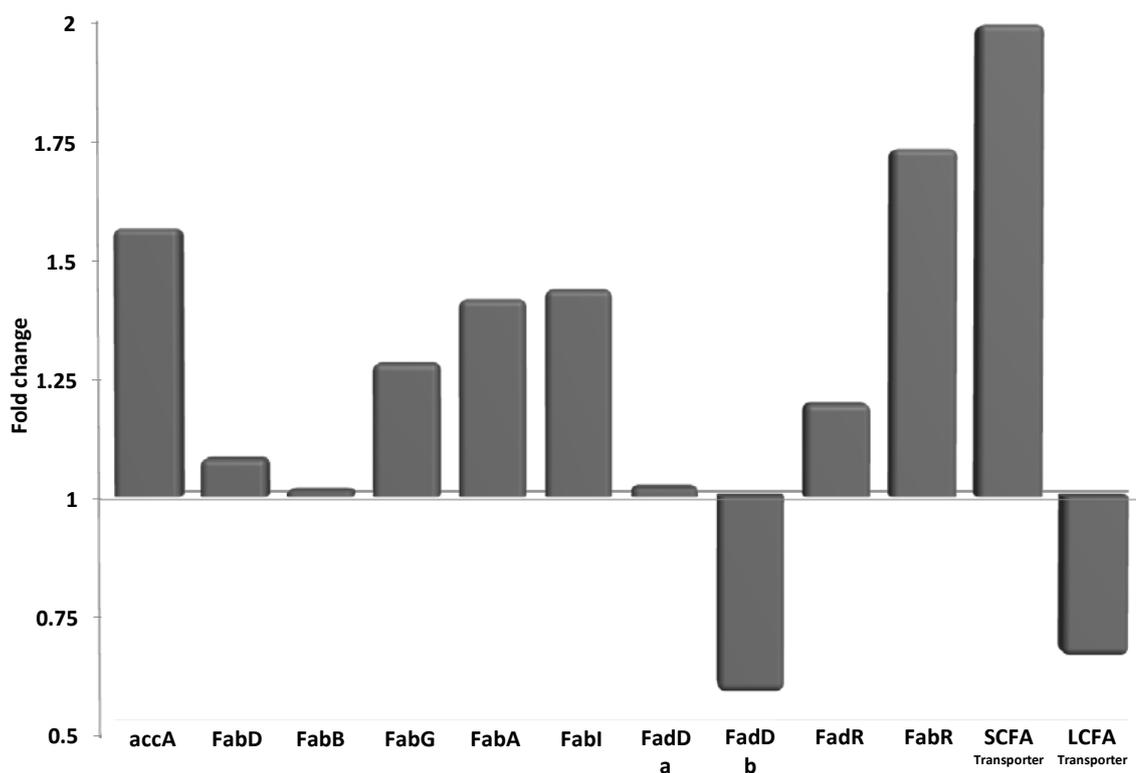


Figure 4.9 Clustered column graph representing $\phi 24B::\Delta Kan$ MC1061 lysogens fold change in fatty acid gene expression compared to the naïve host MC1061. Order of genes is associated to general or of expression in fatty acid biosynthesis. Samples were taken at mid exponential growth phase.

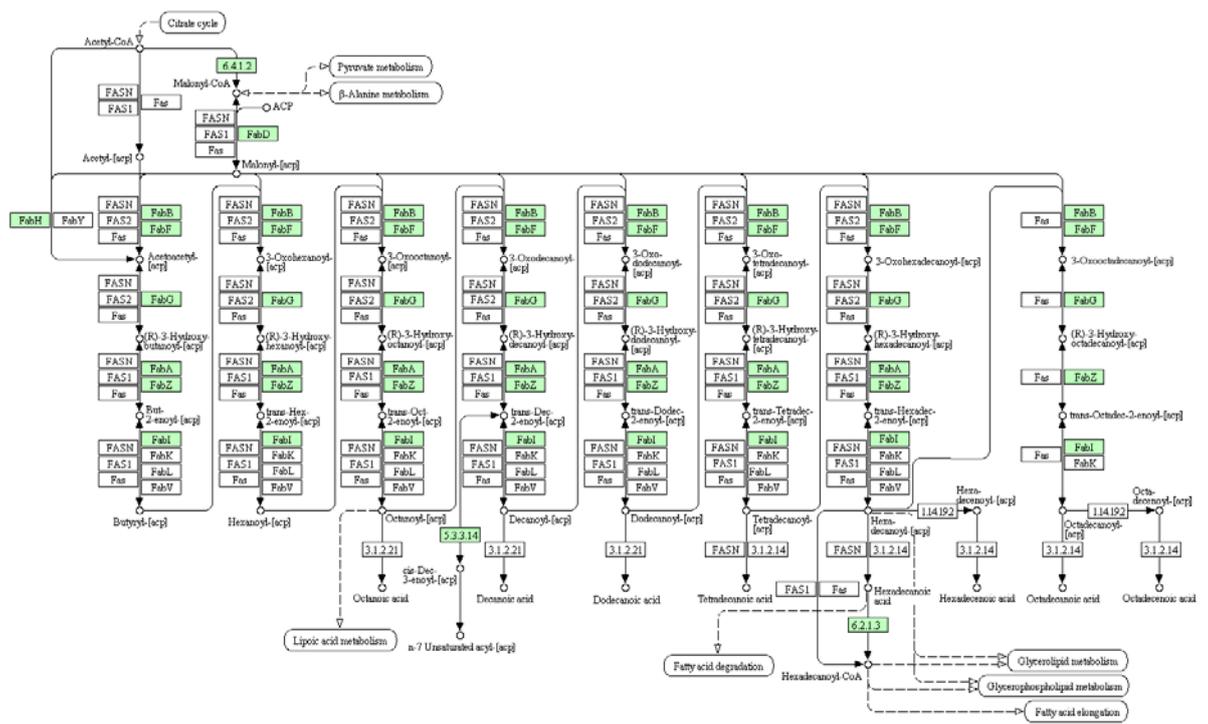


Figure 4.10 Map of the fatty acid pathway. Gene rectangles coloured in green are fatty acid pathway genes associated to K12 *E. coli* (Kanehisa, Furumichi et al., 2017).

4.2.2.3 Transcriptomics: investigating gene expression in pathways linked to cell wall structure

4.2.2.3.1 Lipid gene expression

The lipid pathway is intrinsically linked to the fatty acid and biotin pathway, and has been previously suggested (chapter 3) as a pathway manipulated by $\phi 24_B$ as a mechanism of antimicrobial resistance. Figure 4.11 shows the 15 genes from the transcriptomic data, identified to the lipid pathway, all genes show an increase in expression in the lysogen.

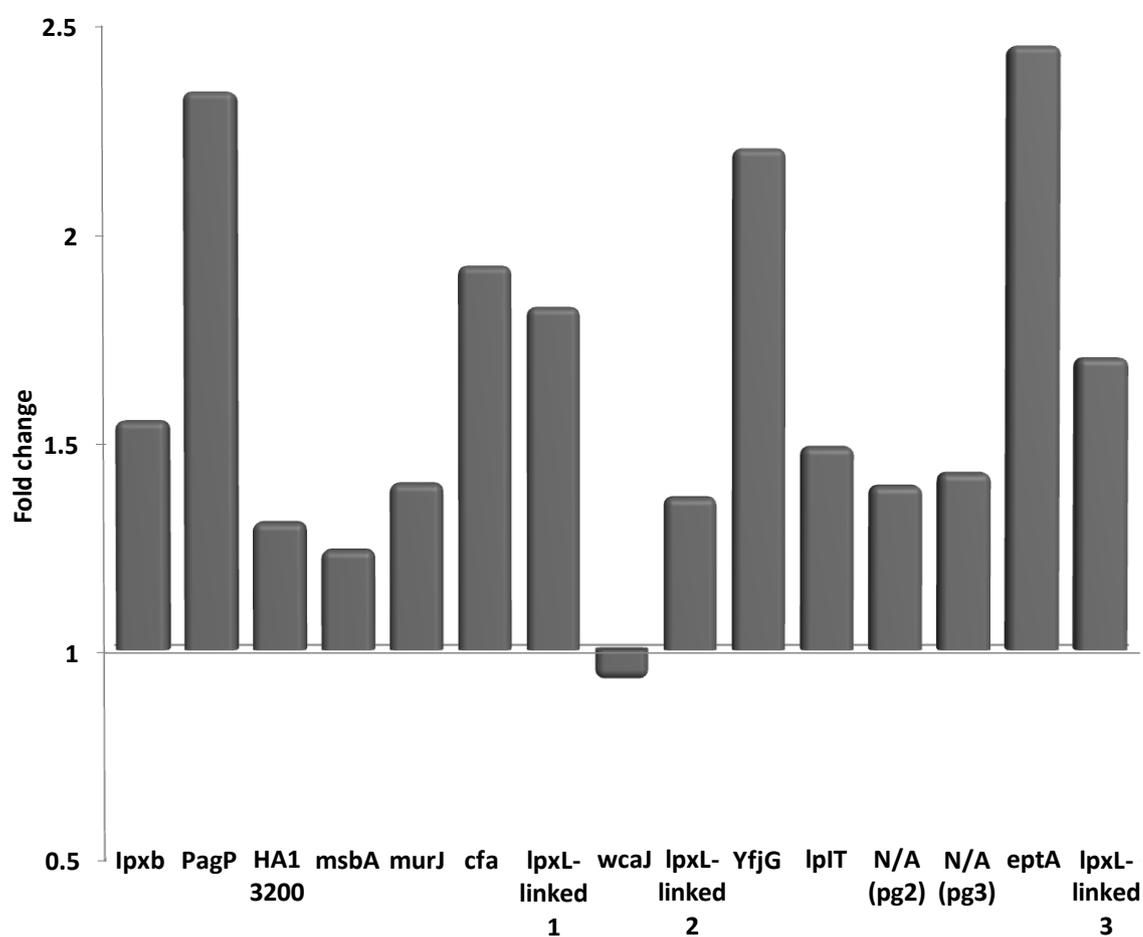


Figure 4.11 Clustered column graph representing $\phi 24_B::\Delta Kan$ MC1061 lysogens fold change in lipid gene expression compared to the naïve host MC1061. Samples were taken at mid exponential growth.

4.2.2.3.2 Peptidoglycan gene expression

The cell wall changes observed in Figure 4.6-4.8 and the broad resistant properties of the lysogen, prompted investigation into lysogen driven changes in peptidoglycan expression as again this is a downstream process of biotin. Gene expression changes may play a role in structure of the peptidoglycan layer in the lysogen, which could have significant effects on cell function and survival. Figure 4.12 shows the 14 genes from the transcriptomic data, identified to the peptidoglycan pathway. Figure 4.12 demonstrates that the lysogen has minimal effect on peptidoglycan synthesis (A), however all genes with peptidoglycan associated structures in the cell wall show an increase in expression in the lysogen (B). The role of the genes in the peptidoglycan pathway can be observed in appendices Figure 10.6, where identified gene expression has been highlighted.

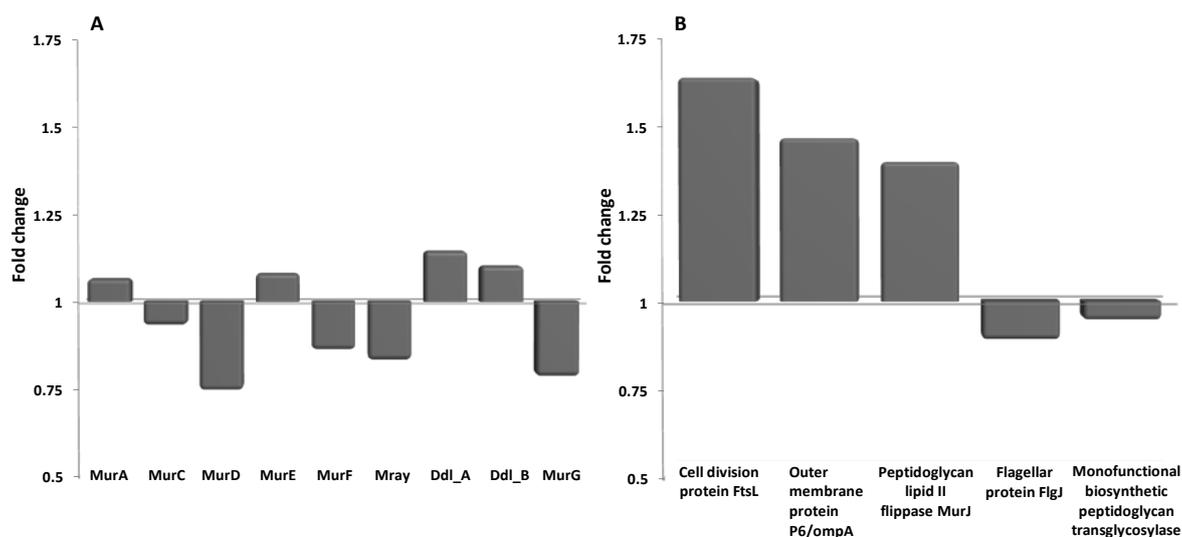


Figure 4.12 Clustered column graph representing $\phi 24B::\Delta Kan$ MC1061 lysogens fold change in gene expression of peptidoglycan synthesis (A) and peptidoglycan associated factors (B) compared to the naïve host MC1061. Samples were taken at mid exponential growth.

4.2.2.4 Fatty acid methyl esters identified periodically over 18 hours under increasing concentrations of chloroxylenol or 8-hydroxyquinoline

Chapter 3 demonstrated that the lysogen $\phi 24_B$ significantly increased tolerance to 50 μm of chloroxylenol and 8-hydroxyquinoline compared to the naïve host. Here we investigated the possibility of acquiring true or full genetic resistance to each antimicrobial. Optical density (OD), viable cell counts (CFU), and cell wall fatty acid content were monitored over 18 hours under increasing concentrations of either 8-hydroxyquinoline or chloroxylenol (Figures 4.12-4.13).

4.2.2.4.1 Cell wall fatty acid content under increasing 8-hydroxyquinoline challenge

Figure 4.13 shows that in the presence of 8-hydroxyquinoline the lysogen and naïve host show a similar initial response to 50 μm 8-hydroxyquinoline. As the cultures shown in Figure 4.7 were sampled at stationary growth as opposed to early growth in Figure 4.13, it suggests initial fatty acid response is sufficient if the drug concentration remains the same. Unlike the lysogen the naïve host shows little adaptation within 18 hours, failing to show any substantial increase in growth after 6 hours (Figure 4.13: plot 3). The fatty acid profile in the naïve host shows a patterned response in increasing levels of fatty acids, where the cell wall fatty acid profile more than triples between 0 and 18 hours. The naïve host shows sharp peaks when 8-hydroxyquinoline concentration surpasses the tolerance of the cell walls adapted fatty acid content, where every peak in fatty acid response is followed by a slight increase in growth. Contrary to the naïve host, the lysogen significantly reduces its cell wall fatty acid content after a greater initial fatty acid spike, a mechanism we hypothesise that supports resistance, shown by exponential growth through the increasing concentrations of 8-hydroxyquinoline. The lysogens fatty acid content slowly rises and peaks again at 15 hours, suggesting the mechanism of resistance was succumb by the increased concentration of 8-hydroxyquinoline. The partial least squares discriminant analysis (PLS-DA) plots in Figure 4.13 (section 2) split the data into 3 time frames to look at the differentiation change between groups over time. Figure 4.13 (section 2) shows an increasing differentiation between the lysogen and naïve host over time, with statistically significant differentiation observed throughout the 18 hours, validated by the R^2 and Q^2 scores in table 1. However the differentiation from the

model of time frame 'a' may be over fitted, as the R2 score is greater than double the Q2 score (Worley & Powers, 2013).

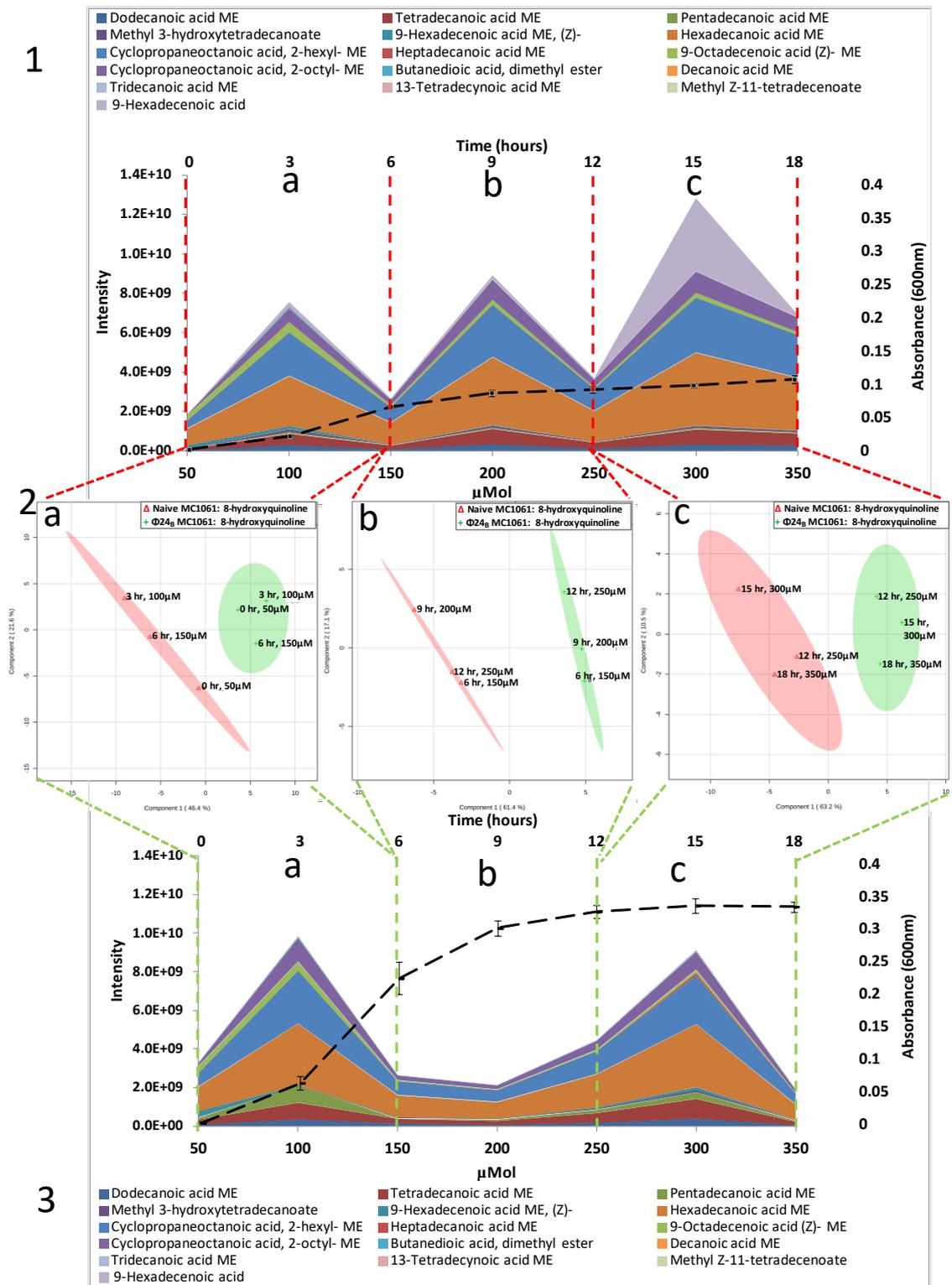


Figure 4.13 Plot representing the fatty acid methyl esters and cell growth of the lysogen and naïve host over 18 hours under increasing concentrations of 8-hydroxyquinoline. PLS-DA is run on each 3rd to classify the points of differentiation between the two cultures. a, b, and c each represent a 3rd of the data divided in order to identify the point at which the greatest differentiation occurs. The stacked fatty acid graphs represent the intensity of fatty acids in the cell wall and the growth of both the lysogen (3) and the naïve host MC1061 (1), grown in the presence of increasing concentrations of 8-hydroxyquinoline over an 18 hour period. Each sample consisted of experimental and technical replicates (n=9). The PLS-DA (2) of each 3rd is identified by the corresponding a, b, or c. PLS-DA is generated from log₁₀ normalised fatty acid intensity of the 16 fatty acids, culture conditions and presence or absence of phage is identified in the plot legend. Each individual mark represents the fatty acid profile of the labelled time point (h), drug concentration (uM), and host (ho) or lysogen (lys). The circles associated to a group are the 95% confident levels, the colour of which denotes the lysogen or host.

4.2.2.4.2 Cell wall fatty acid content under increasing Chloroxylenol challenge

In the presence of chloroxylenol the naïve host grew minimally and demonstrated little cell wall fatty acids response over the 18 hours, suggesting no cell wall fatty acid mechanism of tolerance or resistance was acquired throughout the 18 hours (Figure 4.14). The lysogen has an immediate peak in cell wall fatty acids (4×10^9 an ~ 8 fold increase) in response to the introduction of chloroxylenol. The lysogen fatty acid intensity steadily reduces to original readings of less than 1.0×10^9 over 9 hours, while cell growth increases. This suggests the initial fatty acid response was effective for survival, and that an alternative mechanism is in place after initial fatty acid response, that sustains growth until concentrations of the drug become to high. This is supported by Figure 4.8, which demonstrates no significant fatty acid spike at stationary growth when the low drug concentration remains the same. This also signifies that after the fatty acid spike, whatever the mechanism is, it's sufficient for survival and growth, so long as the concentration isn't increased. At 9 hours the drug concentration is at $200 \mu\text{mol}$, this causes a secondary spike in cell wall fatty acids, ~ 2 fold greater than the previous spike. Although the fatty acid content decreases, it maintains ~ 1.8 fold increase in intensity ($\sim 1.8 \times 10^9$) than previous recoveries ($\sim 1.0 \times 10^9$), suggesting resistance/tolerance acquired is struggling to deal with the increasing concentration of chloroxylenol. Although both the naïve host and lysogen have a similar response to chloroxylenol, the lysogen has a greater rate of growth, suggesting other influencing factors, be it base improvements previously observed or an entirely unique/novel coping mechanism. The PLS-DA plots in Figure 4.14 (section 2), show differentiation between the lysogen and naïve host, however, this could not be statistically validated from the R^2 and Q^2 scores (Table 4.1).

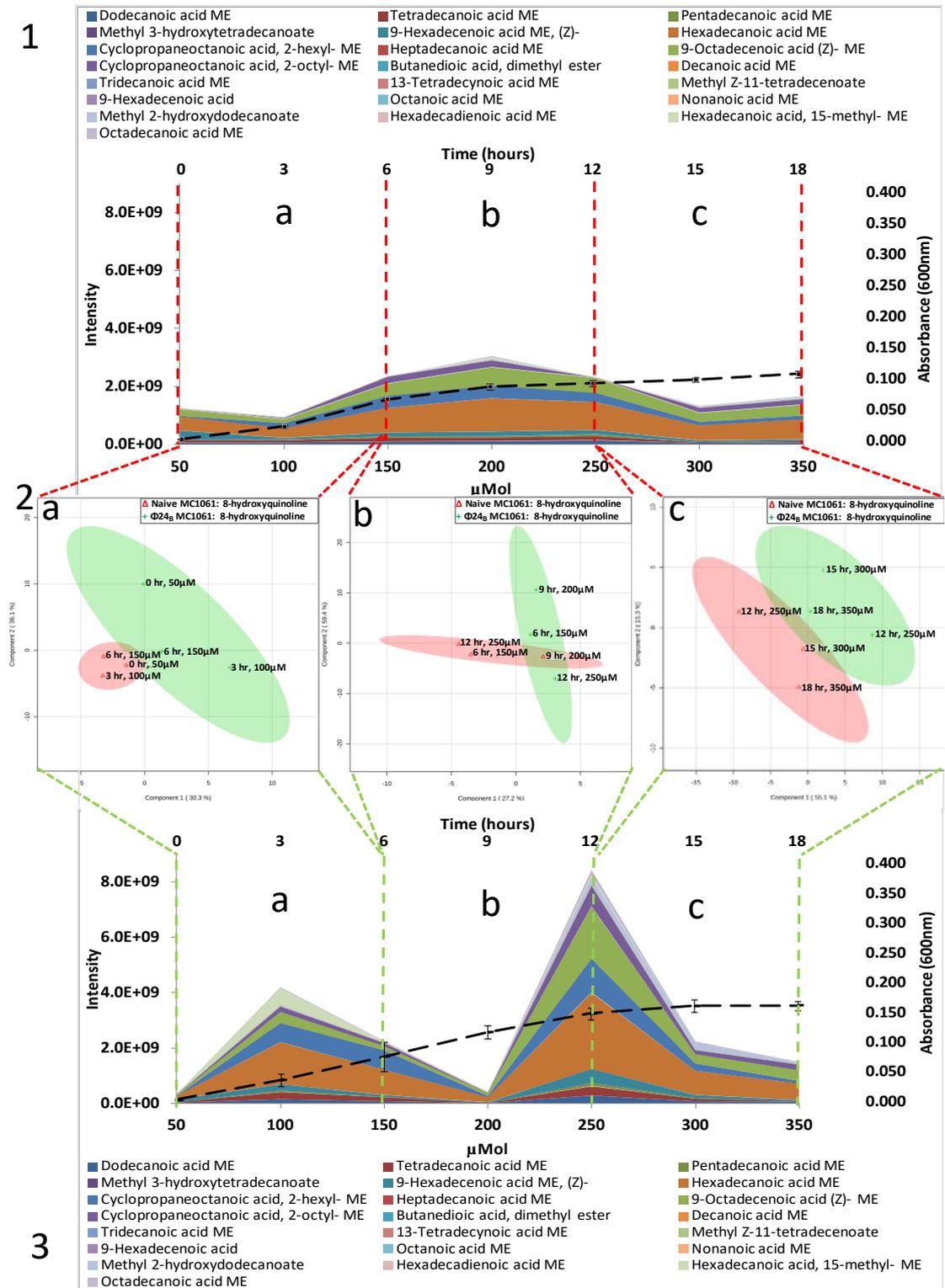


Figure 4.14 Plot representing the fatty acid methyl esters and cell growth of the lysogen and naïve host over 18 hours under increasing concentrations of chloroxylenol. PLS-DA is run on each 3rd to classify the points of differentiation between the two cultures. a, b, and c each represent a 3rd of the data divided in order to identify the point at which the greatest differentiation occurs. The stacked fatty acid graphs represent the intensity of fatty acids in the cell wall and the growth of both the lysogen (3) and the naïve host MC1061 (1), grown in the presence of increasing concentrations of chloroxylenol over an 18 hour period. Each sample consisted of experimental and technical replicates (n=9). The PLS-DA (2) of each 3rd is identified by the corresponding a, b, or c. PLS-DA is generated from log10 normalised fatty acid intensity of the 16 fatty acids, culture conditions and presence or absence of phage is identified in the plot legend. Each individual mark represents the fatty acid profile of the labelled time point (h), drug concentration (uM), and host (ho) or lysogen (lys). The circles associated to a group are the 95% confident levels, the colour of which denotes the lysogen or host.

Table 4.1 R2 and Q2 scores used for validating PLS-DA accuracy in group differentiation. The scores relate to the PLS-DA plots in figure 13 and 14. Scores greater than 0.4 in both R2 and Q2 support the validity of the PLS-DA. The closer the score is to '1' the greater inference can be made from the PLS-DA differentiation. R2 scores significantly higher than Q2 score can mean the model is over fitted

	8-hydroxyquinoline (figure 13)			Chloroxylenol (figure 14)		
	a	b	c	a	b	c
R2	0.98	0.995	0.906	0.85	0.54	0.875
Q2	0.45	0.892	0.755	<0.0	<0.0	<0.0

4.2.2.4.3 Differentiating cell wall fatty acid profiles under increasing antimicrobial challenge

Figure 4.15 VIP plots of fatty acid methyl ester intensities of the lysogen and naïve host over 18 hours under increasing concentrations of 8-hydroxyquinoline, are derived from Figure 4.13 PLS-DA plots. They show which fatty acids are driving the group separation between the lysogen and naïve host at any given time frame (a, b, or c). The figure shows that while up-regulation of methyl Z-11-tetradecanoate is an important differentiating factor in the lysogens cell wall fatty acid content, it becomes less defining as time progresses. The opposite can be said for the lysogens up-regulation of heptadecanoic acid, which increases significantly in importance over the 18 hours. As seen in previous plots the majority of upregulation in fatty acids is observed in the naïve host. The naïve hosts limited adaptation in response to the increasing concentration of 8-hydroxyquinoline is emphasised here, where the majority of its most influential fatty acids remain relatively un-moved over the 18 hours. Cell wall Methyl-3-hydroxytetradecanoate is significantly down-regulated by the naïve host, while cell wall 9-hexadecanoic acid intensity significantly increases over time relative to the lysogen.

VIP plots for the chloroxylenol conditions of the 18 hour study are not included, this is due to the low R2 and Q2 scores of the PLS-DA plots. Group differentiation observed under chloroxylenol conditions could not be trusted without validation, making the VIP plots derived from the PLS-DA's unreliable.

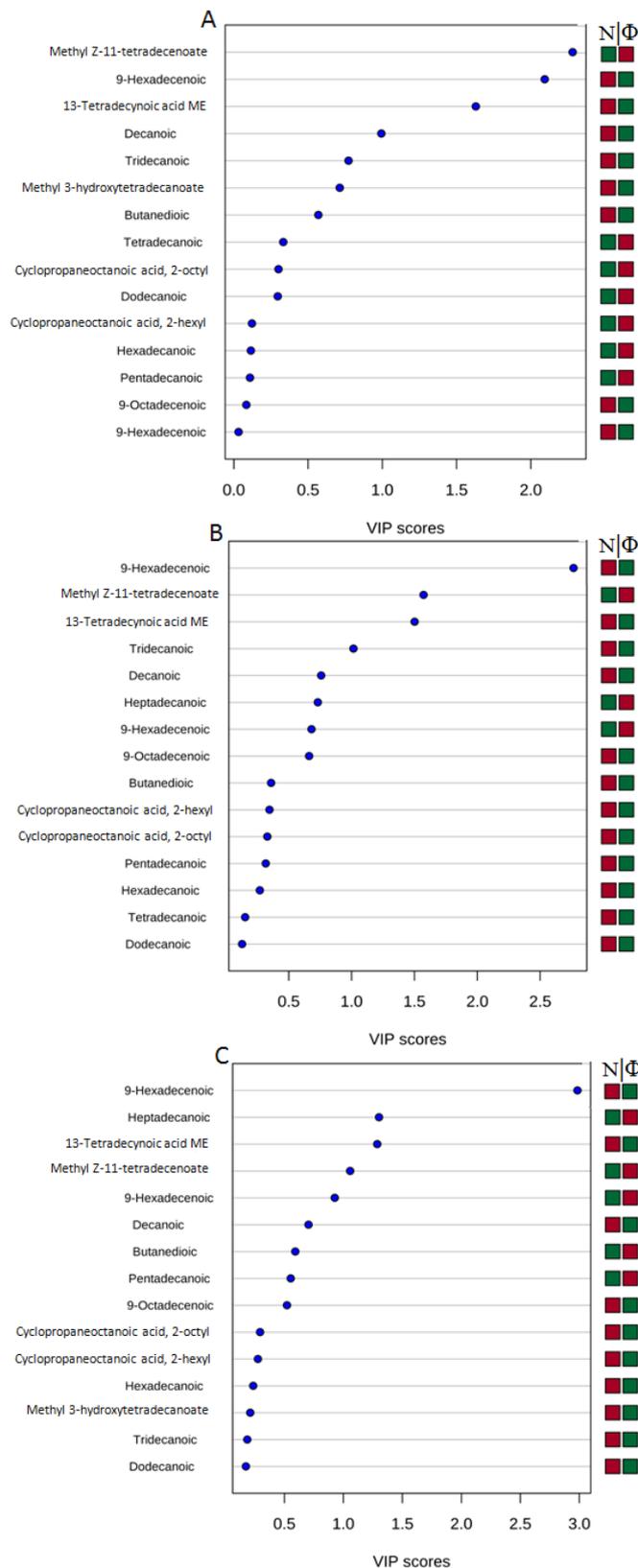


Figure 4.15 VIP plots representing the fatty acid methyl esters of the lysogen and naïve host over 18 hours under increasing concentrations of 8-hydroxyquinoline. The VIP is plotted from the 3 PLS-DA's in Figure 4.13, showing which fatty acids are having the greatest influence on any group differentiation. Plot a, b, and c correspond to to Figure 4.13 PLS-DA plots a, b, and c. The red (high) or green (low) squares show which fatty acids are having the greatest contribution toward differentiation. The 2 columns of squares represent the Naïve host (N) and lysogen (ϕ) respectively.

The complexity of the fatty acid data creates a challenge in observing patterns and variables/factors of significance. PLS-DA, heatmaps, and biplots were used to help interpret the data. PLS-DA were implemented instead of principle component analysis (PCA) due to their predictive power, which helps with complex multivariable datasets. Prior to downstream analysis, normalisation (log10) was carried out to prevent low intensity data of significance being lost/overshadowed (See appendices section 10.3.1).

Fatty acid data over 18 hours under both antimicrobial stress conditions were plotted as a heatmap, where sample types are mapped as a dendrogram using hierarchical clustering (Figure 4.16). Both the lysogen and the naïve host display a fatty acid profile bespoke to either chloroxylenol or 8-hydroxyquinoline tested. However, unique to stress under 8-hydroxyquinoline, the lysogen and naïve host have completely different cell wall fatty acid profiles. This is not comparable to stress under chloroxylenol, suggesting that time and concentration are a more influential factor than the lysogen or naïve host in the presence of chloroxylenol. In the presence of 8-hydroxyquinoline there is a clear upregulation of heptadecanoic acid and methyl Z-11-tetradecanoic acid, and downregulation in decanoic acid, 13-tetradecynoic acid and 9-hexadecanoic acid in the lysogen compared to the naïve host.

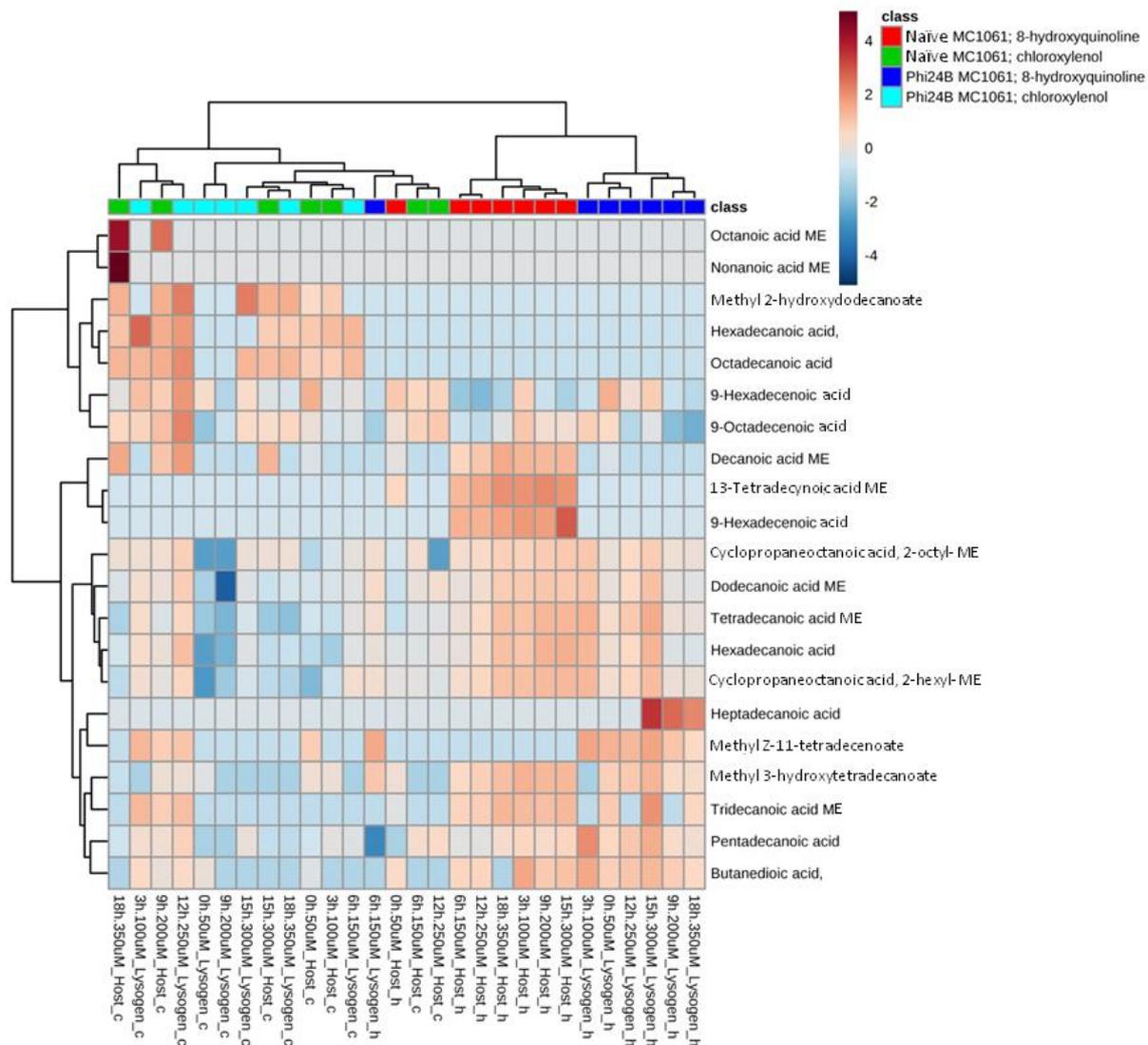


Figure 4.16 Heatmap of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial. Heatmap generated by normalised fatty acid intensity of 21 fatty acids, culture conditions and presence or absence of phage can be found along the horizontal axis (in the following order: timepoint [0-18]h, drug concentration (uM), drug type. Drug type key: h = 8-hydroxyquinoline, c = chloroxylenol). Each individual tile represents a fatty acid. The colour of a given tile denotes higher or lower intensity of the fatty acid. The colour scale key is: dark blue: lowest levels; white: mid-point; dark red: highest level. The gradient between these colours represents variation in the levels of the fatty acid across the colour scale.

The 18 hour fatty acid data used in the heatmap is plotted in a PLS-DA (Figure 4.17), its predictive power can help identify groups in complex datasets. It should be noted that the predictive strength of PLS-DA can also introduce bias/forced grouping, this is why it's important to accompany it with other unsupervised and supervised analysis methods, as well as R2 and Q2 validation.

Figure 4.17 explains 31.2 % and 17.9 % of the data and differentiation validated with R2 and Q2 scores of 0.52 and 0.4 respectively. Validation of the differentiation is weak, this is to be expected due to the plot representing the entirety of data variables, this is addressed in following plots. The figure confirms what's been observed in the heatmap, with the clearest difference between the lysogen and naïve host observed in the presence of 8-hydroxyquinoline. Unlike the heatmap the relationship between the groups is more visible, the PLS-DA shows that the lysogen shares a far similar fatty acid profile under the two antimicrobials than the naïve host. The naïve host under 8-hydroxyquinoline stress is the most unique fatty acid profile, suggesting a specific/specialised response, but as previously shown it had the poorest growth, this could indicate that the specialised response is targeted for an alternative stress, and its implementation highlights the naivety of the host to the stress. The tight grouping of the naïve host in the presence of 8-hydroxyquinoline also suggests there is little adaptation to increases in concentration through the 18 hours of growth, this is supported by figure and detailed in Figure 4.13. As tighter groups suggest less adaptation over time, the figure also suggests that the lysogen, which grew well in the presence of 8-hydroxyquinoline, requires only moderate changes to its initial fatty acid response when acquiring antimicrobial resistance. This is seen in Figure 4.15 with the switch of one fatty acid for another, this being the reduced importance of Methyl-Z-11-tetradecanoate and increased importance of 9-hexadecanoic.

Both the lysogen and the naïve host show a broad and changing fatty acid profile under chloroxylenol duress, suggesting significant adaptation was required to survive. To visualise fatty acid response to antimicrobial presence a biplot was created demonstrating the extent to which any given fatty acid is influencing the differentiation between the lysogen and naïve host (Figure 4.18).

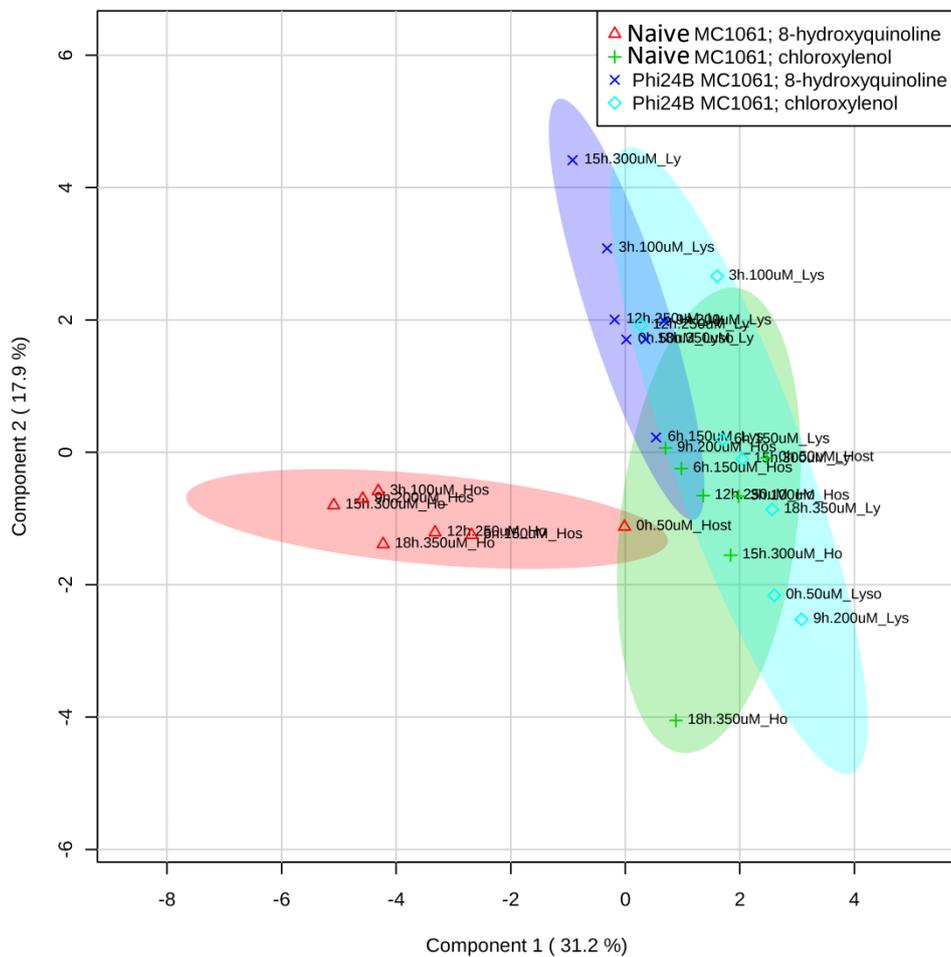


Figure 4.17 PLS-DA of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial. PLS-DA generated by normalised fatty acid intensity of 21 fatty acids, culture conditions and presence or absence of phage is identified in the plot legend. Each individual mark represents the fatty acid profile of the labelled time point (h), drug concentration (uM), and host (ho) or lysogen (lys). The circles associated to a group are the 95% confident levels, the colour of which denotes the lysogen or host.

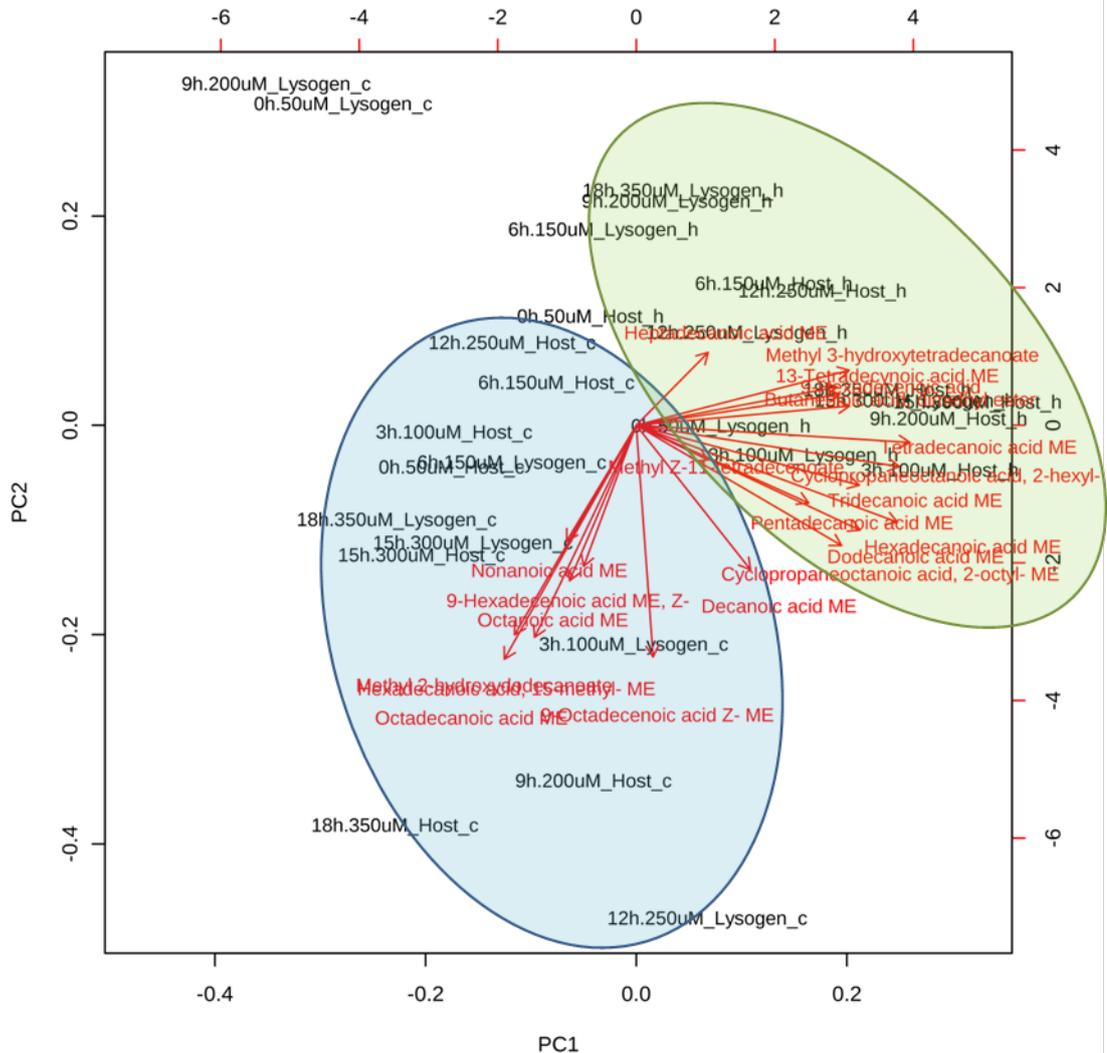


Figure 4.18 Biplot of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial. Biplot generated by normalised fatty acid intensity of 21 fatty acids. Each individual mark in black represents the fatty acid profile of the labelled time point (h), drug concentration (uM), host or lysogen, and drug (chloroxylenol: c, 8-hydroxyquinoline: h). The colour of a circled group also denotes the associated drug (blue: chloroxylenol, green: 8-hydroxyquinoline). The red arrows show which the extent in which a given fatty acid represents a given group.

4.2.2.4.4 Differentiation of lysogen and naïve host cell wall fatty acid profiles under increasing 8-hydroxyquinoline challenge

As discussed above, the fatty acid differentiation between the naïve host and lysogen is defined more under challenge with 8-hydroxyquinoline. As such, greater focus is given to the naïve host and lysogen fatty acid profiles over the 18 hours, under challenge with 8-hydroxyquinoline (see Figure 4.19-4.21). By excluding the chloroxylenol data, the heatmap in Figure 4.19 shows

more clearly that at 0 hours the naïve host has more similarity to the lysogen. There is a clear pattern in the naïve host with upregulation in the majority of cell wall fatty acids, excluding heptadecanoic acid and methyl Z-11-tetradecanoic acid. Methyl Z-11-tetradecanoic acid is strictly upregulated by the lysogen, whether this is a direct resistant mechanism, or a necessary action to allow down-regulation of the majority of other fatty acids is not yet known.

The PLS-DA (Figure 4.20) explains 36.5 % and 30 % of the data respectively, with differentiation validated with R2 and Q2 scores of 0.85 and 0.79 respectively. Separation between the 2 groups is easier illustrated when the dataset is restricted to the 8-hydroxyquinoline condition. By focusing on the 8-hydroxyquinoline condition it demonstrates that conversely to Figure 4.17, the majority of lysogen fatty acid data points group closely to the initial fatty acid response over the incubation period. This supports the hypothesis of an effective resistance strategy from first introduction of the drug.

Figure 4.21 depicts the fatty acid changes deemed most responsible for the lysogen and naïve host differentiation in the presence of 8-hydroxyquinoline. The biplot (Figure 4.21 A) demonstrates the extent in which any of the given fatty acid is influencing the differentiation between the lysogen and naïve host. The VIP plot (Figure 4.21 B) effectively shows the order of importance of those fatty acids. The VIP plot also illustrates that 9-hexadecanoic acid has the greatest influence on fatty acid differentiation between the lysogen and naïve host.

Separate analysis of chloroxylenol challenge over the 18 hours demonstrated no evidence of any consistent lysogen and naïve host differentiation through the 18 hour incubation, and is thus not presented here. This failure in differentiation under chloroxylenol can be seen in Figure 4.16. This suggests that cell wall fatty acid profiles are not a defining feature of resistance under chloroxylenol challenge. This implies that improved lysogen fitness is possibly associated to mutation and rate of growth than any unique or predetermined resistant strategy.

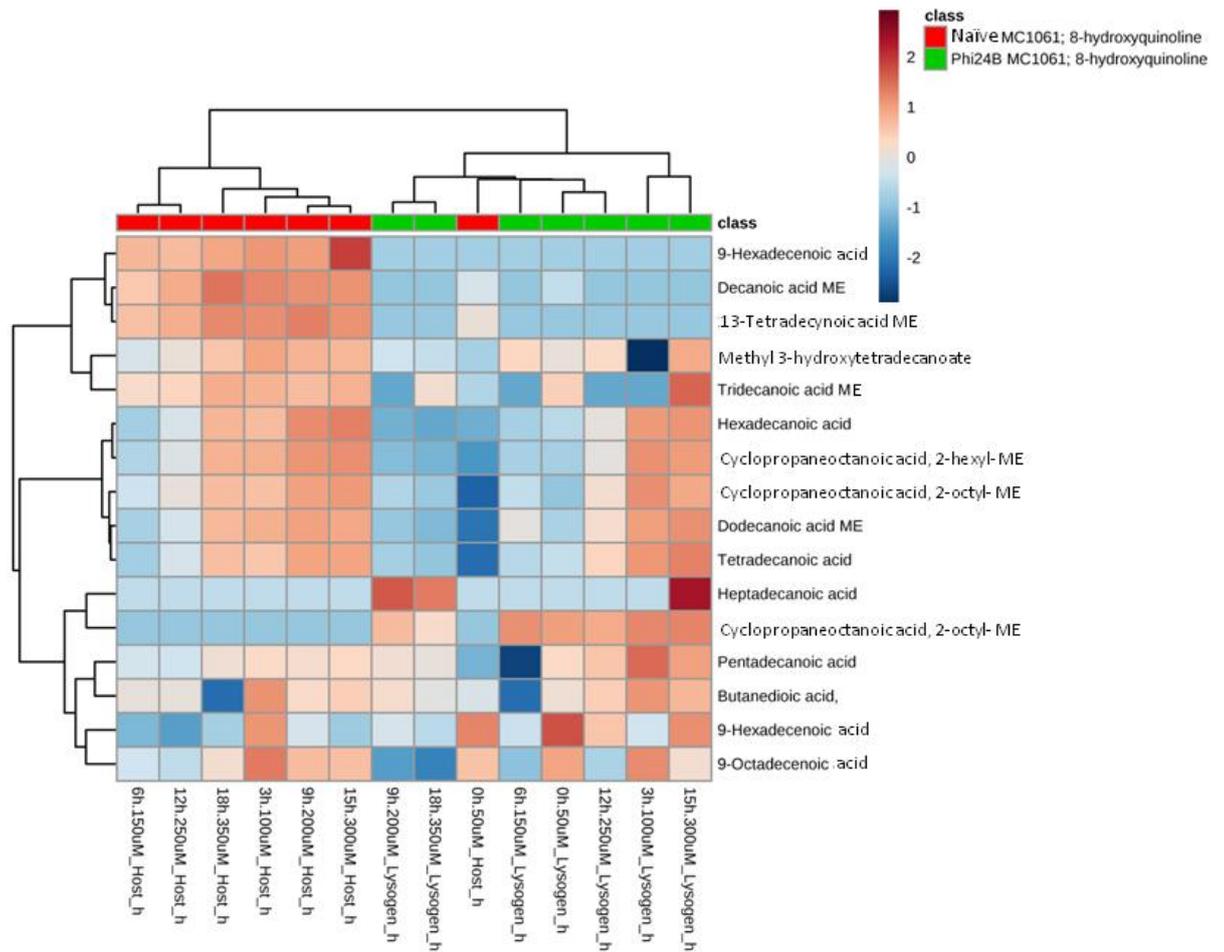


Figure 4.19 Heatmap of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline. Heatmap generated by normalised fatty acid intensity of 16 fatty acids, culture conditions and presence or absence of phage can be found along the horizontal axis (in the following order: timepoint [0-18]h, drug concentration (uM), drug type. Drug type (h = 8-hydroxyquinoline). Each individual tile represents a fatty acid. The colour of a given tile denotes higher or lower intensity of the fatty acid. The colour scale key is: dark blue: lowest levels; white: mid-point; dark red: highest level. The gradient between these colours represents variation in the levels of the fatty acid across the colour scale.

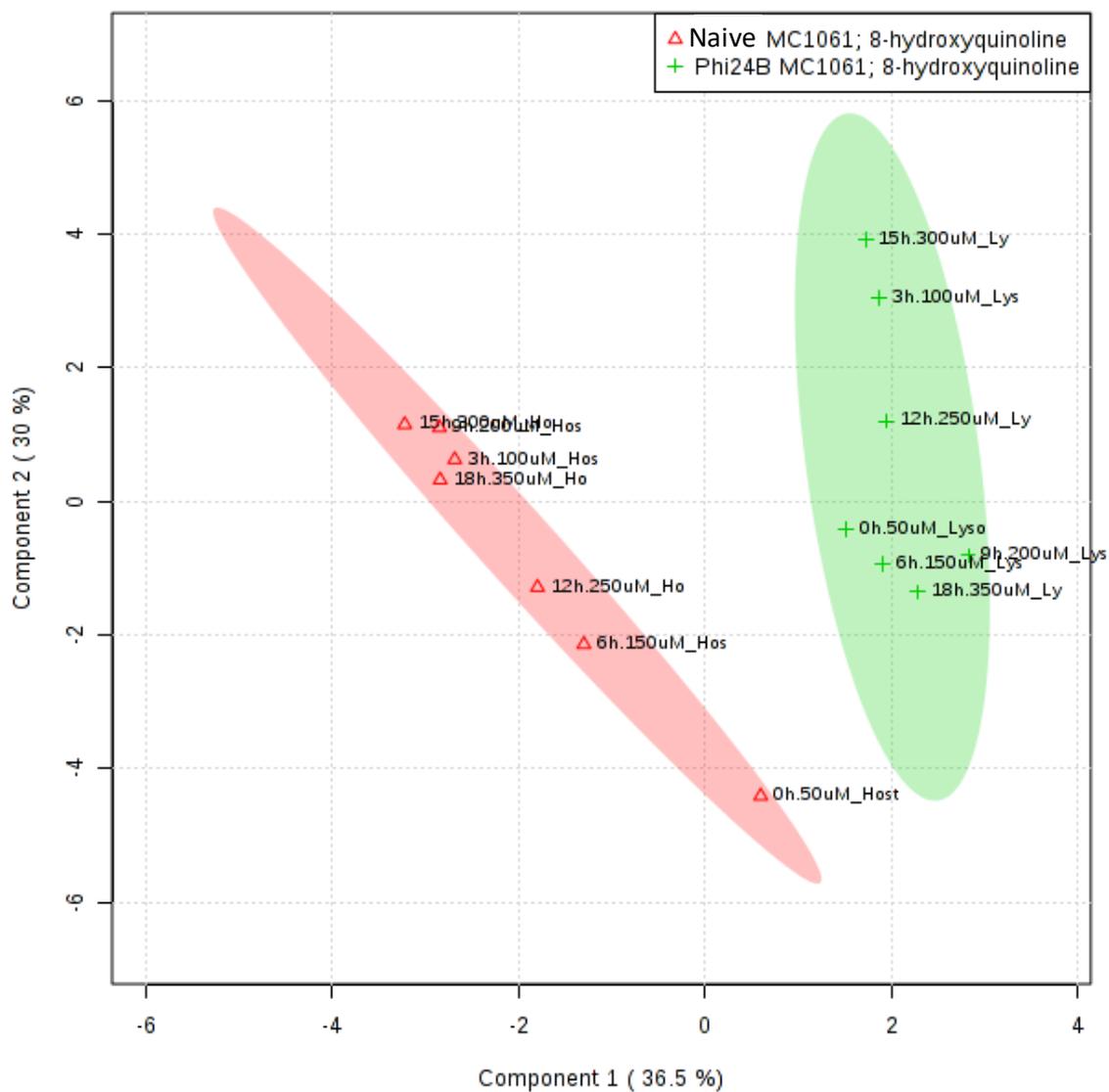


Figure 4.20 PLS-DA of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline. PLS-DA generated by normalised fatty acid intensity of 16 fatty acids, culture conditions and presence or absence of phage is identified in the plot legend. Each individual mark represents the fatty acid profile of the labelled time point (h), drug concentration (uM), and host (ho) or lysogen (lys). The circles associated to a group are the 95% confident levels, the colour of which denotes the lysogen or host.

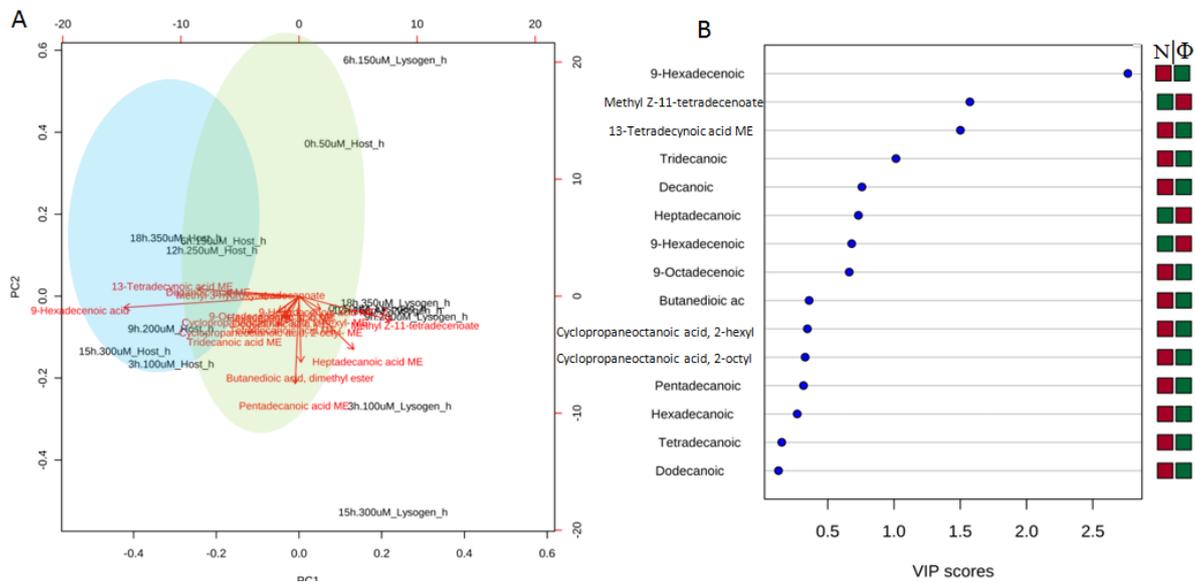


Figure 4.21 Biplot (A) and VIP plot (B) of fatty acid methyl esters identified over 18 hours under increasing concentrations of 8-hydroxyquinoline. Biplot and VIP plot generated by normalised fatty acid intensity of 21 fatty acids. A: Each individual mark in black represents the fatty acid profile of the labelled time point (h), drug concentration (uM), host or lysogen, and drug (8-hydroxyquinoline: h). The colour of a circled group also denotes the associated culture (blue: Naïve host, green: Lysogen). The red arrows show which the extent in which a given fatty acid represents a given group. B: shows which fatty acids are having the greatest influence on any group differentiation, it's an alternative visualisation of the biplot. It improves visual clarity of influential fatty acids, but fails to show its true relation to a group as accurately as biplot. The red (high) or green (low) squares show which fatty acids are having the greatest contribution toward differentiation. The 2 columns of squares represent the Naïve host (N) and lysogen (ϕ) respectively.

4.3 Discussion

4.3.1 Influence of environmental factors on growth

Prophage carriage has been previously associated with improved growth as late back as 1975 (Edlin, Lin et al., 1975b), however the impact of this has since been surprisingly understudied. Continuing from chapter 3, this study identifies the impact that $\phi 24_B$ prophage carriage has on growth under several different environmental stresses. The stress inducing factors are associated to changes in; temperature, oxygen limitation and addition of bile salts. Under these environmental stresses, this study also identifies marked differences in growth in relation to primary and double lysogen. This chapter's investigation into single and double lysogeny in response to environmental factors is a novel approach. Interestingly the frequency of multiple infections and the number of phage related genes has previously been shown greater in pathogenic strains of bacteria, including strains of *E. coli* (Busby, Kristensen et al., 2013, Hayashi et al., 2001, Ohnishi, Terajima et al., 2002, Winstanley, Langille et al., 2009). Characteristics identified in this chapter provide insight into the possible role of single and multiple prophage carriage, in relation to microbial fitness in the environment.

Prophage carriage and the differences observed in growth have likely been largely overlooked in research due to interpreting data based on absorbance or CFU. By plotting growth as the percentage increase with viable cell counts in relation to the naïve host, it was possible to visualise the relative lysogen growth (Figure 4.2 and Figure 4.3). This normalisation allows the impact of lysogeny to be extrapolated, providing particular clarity to early growth differences. This study shows that under all tested conditions (Figure 4.2 and Figure 4.3) the lysogen continues to have significantly greater propagation at early and mid-exponential growth phases. Interestingly the additional advantages provided by double lysogeny during early growth (see chapter 3) are variable under these alternate conditions. Nonetheless, the greater decline in growth displayed by the lysogen in complex media is consistent throughout. The reasons for increased rates of decline in colony counts after early growth have been hypothesised in chapter 3 as cells reaching stationary and death phase more quickly through increased respiration rate.

This chapter's growth data suggests that the lysogen has an overwhelming advantage in establishing itself externally from the animal host, where prophage carriage coincides with increases in growth of > 600 % at 19 °C. Interestingly research has previously shown that *E. coli* has a low rate of survival outside its animal host (Winfield & Groisman, 2003). Therefore Figure 4.2 suggests prophage carriage may significantly influence external environment survival, which in turn may impact environment-host transference of the bacteria. Lysogeny may also improve rates of infection, as there is a marked increase in growth during prophage carriage (~200 %) at temperatures related to core animal hosts, these being 37 and 42°C.

Figure 4.3 identifies that bile salt influences improved growth during prophage carriage. Bovine gut bile salt concentrations (~57 mMol/l) (Doll, Riepl et al., 1999) are far greater than those found in the human gut (~20 mMol/l) (Mallory, Kern et al., 1973, Mallory, Savage et al., 1973). Interestingly this study identifies that in the presence of bile salts, prophage carriage further stimulates growth at temperatures closer to the bovine host (with further increases of 80%). Suggesting the lysogen may regulate growth in relation to animal host background, potentially improving colonisation. There is supporting evidence toward the hypothesis that lysogeny may improve infectivity via the gastrointestinal tract. The findings by Veses-garcia *et al* identify lysogen associated acid resistance, which is linked to the cII gene (Veses-Garcia et al., 2015). The cII gene is essential in maintaining the lysogenic state, suggesting gut related resistance mechanisms are innate upon lysogeny.

Improved infectivity and growth within the gut may have significant impacts on transference and outbreaks of shiga-toxin encoding *E. coli*. The primary reservoir of stx-phages and its *E. coli* host are bovines, the manure of which is used as agricultural fertiliser. The use of manure as fertiliser can result in bacterial leakage into water supplies as well as the transfer of bacteria to food produce. As such, it's important to further investigate possible phage mechanisms in improved bacterial host survival.

4.3.2 Subversion of the Biotin pathway

Chapter 3 identified correlation between increases in cell biomass and biotin presence. Here we confirm using transcriptomic data that $\phi 24_B$ can subvert the biotin synthesis pathway, however we cannot yet confirm that biotin is rate limiting to growth in this instance. Transcriptomics comparing lysogens to their naïve hosts remains a very novel approach. Publications investigating this occurring only in the last 6 years, and make up just 0.9% of the literature associated to bacterial transcriptomics (data attained via PubMed). Mining of transcriptomic data identified that $\phi 24_B$ infection up-regulates the core biotin pathway genes, and can alter regulatory genes associated to both biotin repression and biotin-complex structures (Figure 4.4). These biotin complexes are essential in peptide transport and fatty acid production. Mapping the combined data gathered on biotin thus far to the biotin synthesis pathway map, we see the extent of phage subversion (Figure 4.4). Interestingly the transcriptomic data illuminated to additional gene subversion in the lysogen that is intrinsically linked to the biotin and fatty acid synthesis pathways. A plethora of gene expression differences were observed between the lysogen and naïve host, many of which had implications in growth. Of particular note is the expression of the *aceF* and *aceE* genes. The *aceE* and *aceF* genes encode for the pyruvate dehydrogenase complex and were found to have an approximate 3 fold difference in expression. The multi-enzyme complex is responsible for pyruvate decarboxylation, which is the conversion of pyruvate into acetyl-CoA. Acetyl-CoA is the primary source of energy for the TCA cycle, acetyl-CoA is also the primary source of energy for the fatty acid synthesis pathway via biotin carboxylation (see Figures 4.22-4.23).

4.3.3 Prophage carriage on fatty acids and cell wall fatty acids.

Fatty acid synthesis has many roles in the physiology of bacteria, including the cell wall structure (Zhang & Rock, 2008a), antimicrobial resistance (Di Pasqua, Hoskins et al., 2006) and transportation of compounds (Zhang & Rock, 2008a) (appendices Table 10.10). This study identifies that prophage carriage causes marked differences in cell wall fatty acids, particularly

under antimicrobial challenge. Furthermore we demonstrate phage subversion of the fatty acid synthesis genes.

Figure 4.6 illustrates how the lysogen has a lower cell wall fatty acid average than the naïve host, under optimal growth conditions. Fatty acid synthesis in relation to cell wall maintenance and adaptation have been previously reported (Parsons & Rock, 2013). However, as far as we know, there has been no research that has identified an overall reduction in total cell wall fatty acids after growth in optimal conditions. Down-regulation of fatty acids during optimal growing conditions may be a form of resource re-distribution during low stress conditions. The exact implication of this is unknown, but it may be linked to further increases in growth and/or cost of prophage carriage.

We see an overall increase in fatty acid gene expression in the lysogen compared to the naïve host (Figure 4.9). This suggests a significant increase in available fatty acids in the lysogen, as the cell wall fatty acids are down regulated while fatty acid pathways identified in the transcriptomic data show increased expression. Increased fatty acid expression in the lysogen may be a result of manipulation of biotin pathways for increasing growth rate, however, the purpose and use for this possible increase in free fatty acids available within the cell remains uncertain. Fatty acids play many essential roles in bacteria (appendices Table 10.10), and the lysogen could be re-purposing the resources for any such role, providing any number of additional advantages. For example it is known that changes in cell wall fatty acids and the related recycling of phospholipids is crucial to bilayer stability in dividing cells (Zhang & Rock, 2008a) and this may be reason of re-purposing due to the lysogen increased growth rate.

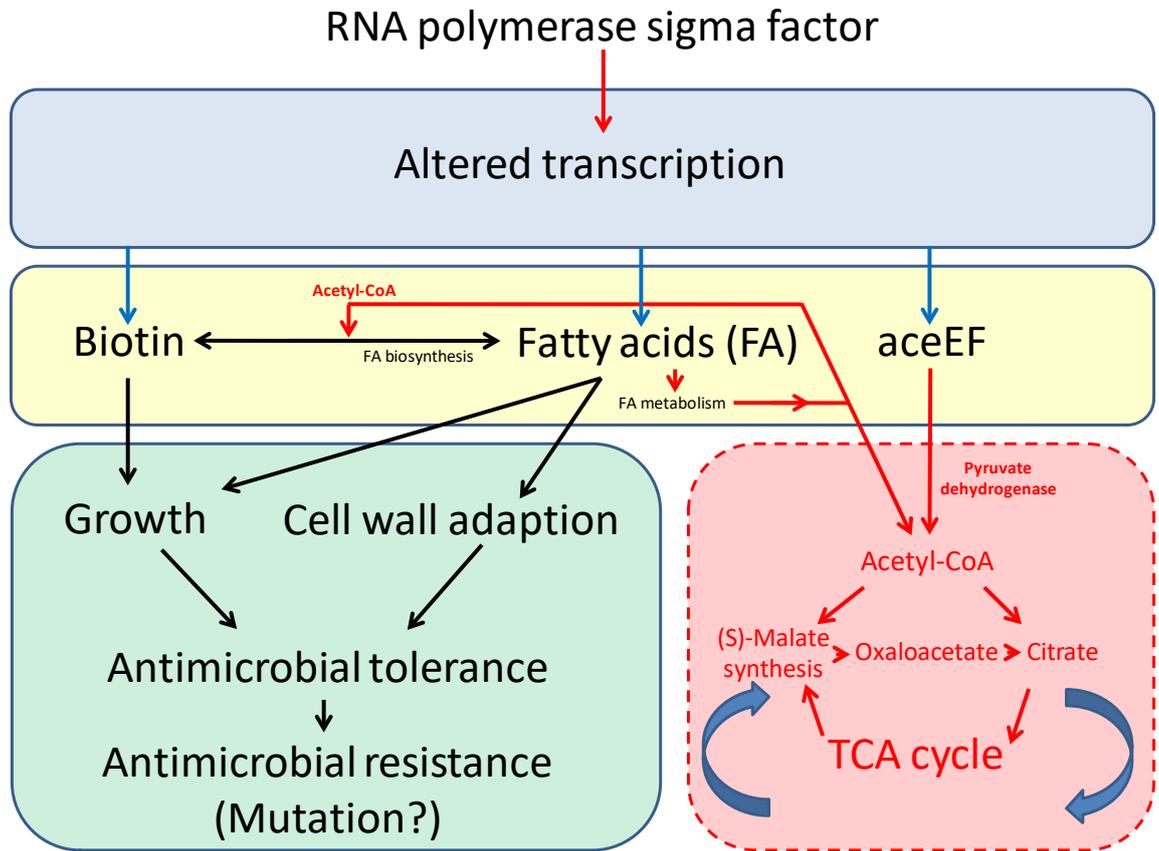


Figure 4.22 An illustration of the working hypothesis based on observed biochemical, transcriptional, and phenotypic changes linked to lysogeny. Factors inside the yellow box are key genes and pathways identified as having a role in the phenotypic changes observed. Factors in the green box are the core observed phenotypic changes associated to antimicrobial resistance. Arrows and their direction relate to factors and co-factors that a given gene or metabolite impacts. Information in red and the red dashed box, are factors that would be expected to be impacted by the changes observed, yet have not been quantified within this study.

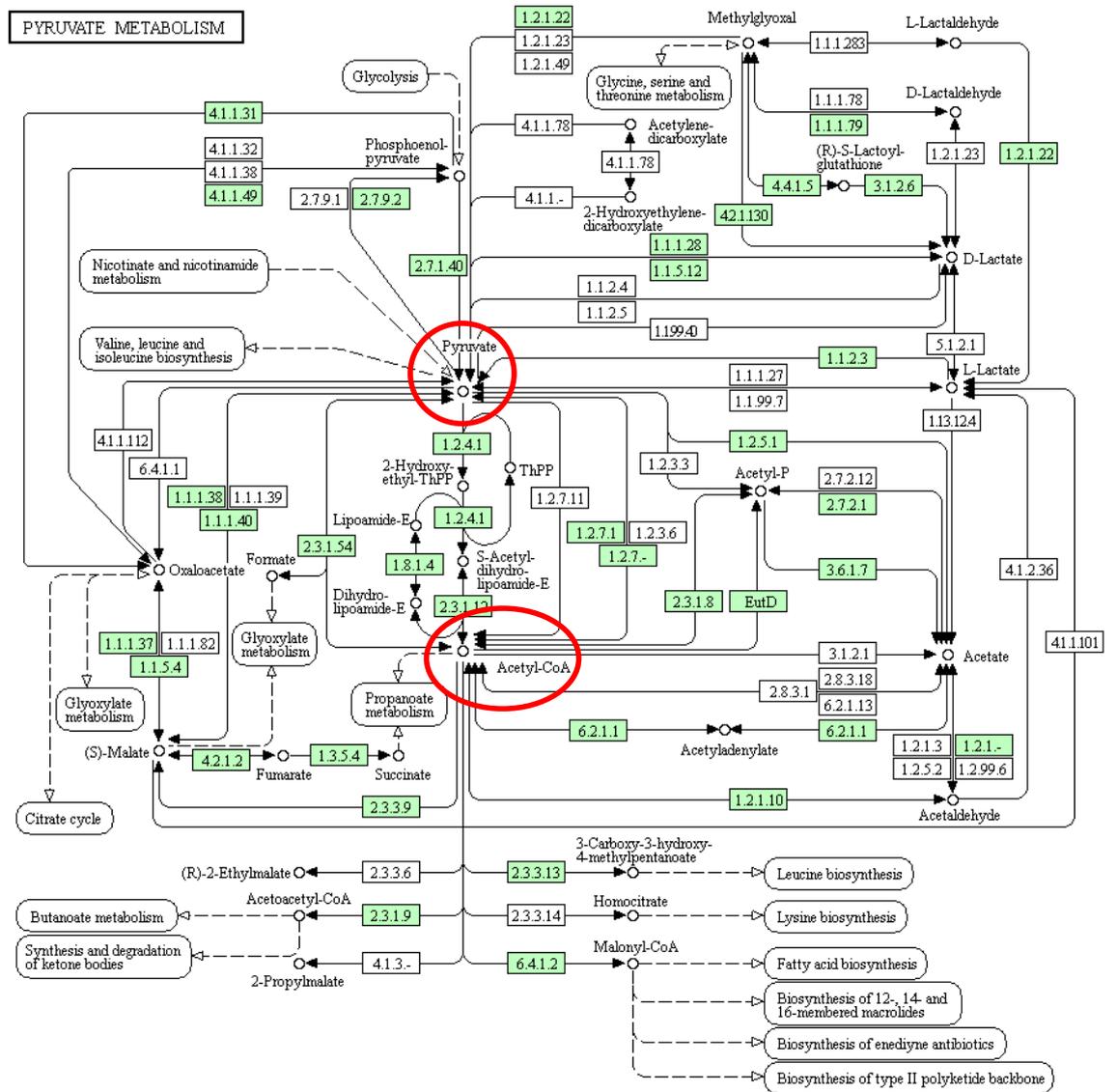


Figure 4.23 Map of pyruvate metabolism, highlighting key factors associated to biotin and fatty acid synthesis pathway, as well as the citric acid cycle. Gene rectangles coloured in green are genes associated to K12 *E. coli* (Kanehisa, Furumichi et al., 2017).

4.3.3.1 Phage driven antimicrobial tolerance and associated cell wall fatty acid profiles

Figures 4.6 and 4.7 show the impact of sub-inhibitory concentrations of antimicrobials on cell wall fatty acid content, the cell wall structures and antimicrobial target regions can be found in Figure 4.24. In the presence of 8-hydroxyquinoline (Figure 4.7), the naïve host and lysogen have opposite reactions, the lysogen significantly ($p < 0.05$) increases average fatty acid content while the naïve host decreases it. This stark contrast in response to the drug likely explains the 2 different tolerance profiles previously observed (chapter 3). Chapter 3 suggested from LCMS metabolomic analysis that lipids were likely an important factor in lysogen tolerance to 8-hydroxyquinoline, the data here supports that hypothesis. The hypothesis being, that due to the lipophilic nature of 8-hydroxyquinoline, an increase in lipids likely interferes with the drugs ability to reach its intracellular targets. Previous data has suggested the lysogen increases its intracellular and extracellular lipids in an immediate response to sub-inhibitory concentrations of 8-hydroxyquinoline (chapter 3), here we show an increase in cell wall fatty acid content, the building blocks of essential cell wall lipids. Increases in cell wall lipids especially core structures such as 'lipid A', could have significant effects on antimicrobial tolerance (Needham & Trent, 2013), furthermore, increases in cell wall lipids may make translocation across the cell wall difficult for a lipophilic drug. Lipid changes under stress may impact other factors such as pathogenicity, as core sub-structures of lipopolysaccharides such as lipid A have been associated in increased pathogenicity (Needham & Trent, 2013).

In the presence of chloroxylenol (Figure 4.8), both the naïve host and the lysogen increase their cell wall fatty acid content, but lose heptadecanoic acid. The disappearance of measurable heptadecanoic acid might be the result of a recycling necessity associated to the cost of other fatty acid increases. Of particular note is that heptadecanoic acid has significant roles in bacterial growth, considering chloroxylenol as a bacteriostatic drug, we may be observing one of the mechanisms behind the drugs bacteriostatic effects. There are several possible reasons for the overall increase in fatty acids in both cultures, however, we can presume that it does not serve as a standalone defensive strategy. This presumption is based in the fact that the greatest increase in fatty acids is observed in the naïve host, while the greatest tolerance to chloroxylenol is observed in

the lysogen. As the lysogen has been shown to have improved growth against chloroxylenol, it suggests that the greater increase in fatty acids by the naïve host is excessive and not an effective strategy against chloroxylenol for long term growth. This is likely due to chloroxylenol targeting the membrane rather than the intracellular target like those of 8-hydroxyquinoline (Figure 4.24).

The lysogen demonstrates that, depending on the antimicrobial, increased cell wall fatty acid content is not always the most effective mechanism of resistance. There are many other core cell wall structures that can play an important role in resistance, such as the peptidoglycan layer and the fatty acid associated core lipid structures, found in Gram negative bacteria like $\phi 24_B$ MC1061 and naïve host MC1061.

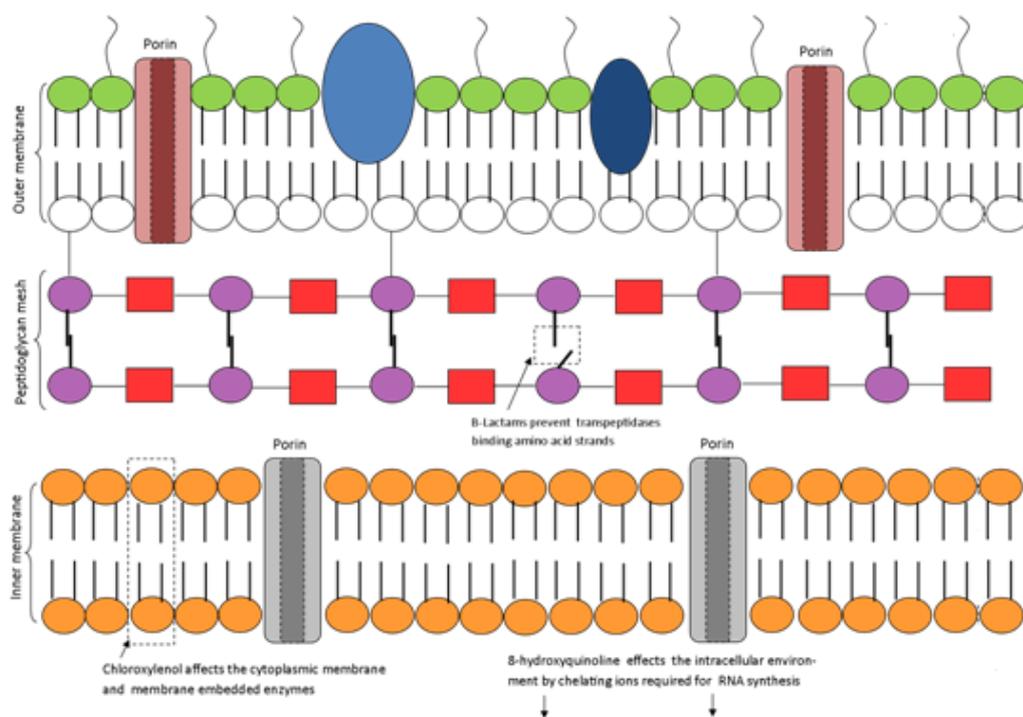


Figure 4.24 Gram negative bacterial cell membrane structure, highlighting the target sites of antimicrobials 8-hydroxyquinoline and chloroxylenol.

4.3.3.2 Phage subversion of the lipid and peptidoglycan pathways

When investigating the extent and mechanisms of antimicrobial resistance acquired by the lysogen, it seemed prudent to understand the potential breadth of lysogen derived manipulation of the cell wall structure. As such, lipid and peptidoglycan gene expression were investigated. A significant increase in both lipid and peptidoglycan gene expression was observed in the lysogen (Figure 4.11 and Figure 4.12), signifying the lysogens ability to subvert these pathways. Lipids have many roles in and out of the bacterial cell, effecting membrane dynamics and regulating cellular functions. Interestingly it's been shown that lipid production in *E. coli* is regulated in a manner that either determines the cell cycle or is dependent on it, this also suggests that biotin may be rate limiting to cell growth (Barak & Muchova, 2013). The Peptidoglycan layer located between the outer and inner membranes (Figure 4.24) is an essential cell wall structure in Gram negative bacteria, managing cell wall flexibility and large molecule diffusion, as well as being intimately involved in cell growth and division (Typas, Banzhaf et al., 2012). As such, it is likely that changes in peptidoglycan synthesis relate to changes in growth (cell division). Here we find that the lysogen can clearly manipulate both the lipid and peptidoglycan pathways, making the possibilities for providing evolutionary advantages extensive. For example crosslinking of the peptidoglycan has been shown as a broad mechanism in the reduction of antimicrobial entry into the inner membrane (Hugonnet, Haddache et al., 2014, Loskill, Pereira et al., 2014, Nikolaidis, Favini-Stabile et al., 2014). Whether changes in peptidoglycan expression observed in the lysogen alters the crosslinking of the peptidoglycan, is a further area of study that should be investigated.

4.3.3.3 Lysogen adaptation of cell wall fatty acids in acquiring antimicrobial resistance over time

After determining that fatty acid cell wall changes at stationary growth phase occurs under sub-inhibitory antimicrobials, we went on to determine the changes under increasing concentrations of antimicrobials. There are environmental parameters associated with the experimental data presented in this chapter, these being; time, infected or non-infected naïve host, drug introduction, drug type, and drug concentration. The complexity of the data, with multiple components, makes

analysis or ability to identify patterns in the data difficult. It also means that any modelling completed needs to be appropriate for the study in hand. The models used in this study were selected as per appendices section 10.3.2.

4.3.3.3.1 Lysogen cell wall fatty acid adaption and acquired resistance under 8-hydroxyquinoline challenge

The stacked graphs (Figure 4.13 and Figure 4.14) display the cell wall fatty acid changes over 18 hours under increasing concentrations of antimicrobial. Differentiation was seen between the lysogen and naïve host over the 18 hour incubation, represented here using PLS-DA plots. For the first time we show the continuing changes in cell wall fatty acids, implemented under antimicrobial challenge, by the lysogen. As the lysogen is shown to significantly out-compete the naïve host under both antimicrobial challenges, it is possible that cell wall fatty acid changes observed in the lysogen are associated to improved survival.

Figure 4.13 identifies the changes to cells treated with 8-hydroxyquinoline over 18 hours, both the naïve host and lysogen have a similar initial response. Their cell walls increase in fatty acid content at the initial concentration of 50 μm at early growth. Interestingly the lysogens fatty acid content increases to a greater extent, along with a correlated improved growth. It can be hypothesised that this initial response of fatty acid increase is sufficient for survival so long as the concentration is not increased. This hypothesis is supported by matching fatty acid changes observed in Figure 4.6, where cells are subjected to the same 50 μm concentration of drug and left to grow to stationary growth phase.

With further increases in 8-hydroxyquinoline we observe unique methods of survival between the naïve host and lysogen. From the results we can hypothesise that when over stressed by 8-hydroxyquinoline both cultures respond with a spike in cell wall fatty acid contents. However the lysogen has a distinctly extended drop in fatty acid content (as well as exponential growth) after its initial response, not spiking again until growth plateaus under the significantly increasing concentrations of drug. This suggests an alternative and more effective coping mechanism

providing true resistance. The source of this resistance is possibly associated to the increased growth (mops up more drug), and previously discovered (chapter 3) intracellular and possibly extracellular lipid production. The naïve host clearly lacks an alternative mechanism for raising its tolerance to a level of true resistance, instead repeating, at an increasing intensity, the spikes in cell wall fatty acids. This differentiation is quantified in the PLS-DA plots within Figure 4.13. The difference is best quantified between the naïve host and lysogen when the 18 hour data under 8-hydroxyquinoline challenge is pooled and analysed using PLS-DA (see Figure 4.20). The patterns in fatty acid profile between lysogen and naïve host can also be observed in the Figure 4.19 heatmap, which highlights Methyl Z-11-tetradecanoic acid as possibly playing a role in the mechanism of acquired antimicrobial resistance by the lysogen. Methyl Z-11-tetradecanoic acid is only upregulated by the lysogen, except at 0 and 18 hours. 0 and 18 hours are the only two time points that growth rate is at its lowest, and the two time points associated to the other fatty acid spikes that resemble the naïve hosts failed attempts at resisting an increasing concentration of 8-hydroxyquinoline.

4.3.3.3.2 Lysogen cell wall fatty acid profile under chloroxylenol challenge

Figure 4.14 demonstrates the changes to chloroxylenol over 18 hours, unlike the lysogen the naïve host appears to make no significant changes in cell wall fatty acid intensity. The lysogens initial response to 50 μm chloroxylenol at early growth phase (lower average cell wall fatty acid intensity to the naïve host) mimicks the stationary growth phase results of Figure 4.8. This suggests the lysogen has an alternate mechanism to cell wall fatty acid changes when tolerating a 50 μm dose. Its possible that the increased growth rate still provides enough of an advantage to out compete the naïve host at these low doses. The lysogens immediate spike in fatty acid intensity at 100 μm is a similar response the naïve host had against 8-hydroxyquinoline when it appeared to have little other strategy of resistance. Significant spikes in the average cell wall fatty acid content seem to be a generic response to recognised stressors where limited other mechanism of resistance are available. These generic responses appear to provide the time required to adapt and develop alternate more effective mechanisms of resistance. After the peak in fatty acid content the lysogen

appears to adapt and implement an alternative mechanism of resistance leading to a steady reduction in fatty acids. 9 hours after the peak at 100 μM , the concentration of chloroxylenol reaches 250 μM at which point the lysogen's cell wall fatty acids spike again, suggesting the concentration had overwhelmed the resistant strategy currently implemented. The lysogen is shown to adapt a second time, which suffices to the end of the incubation period. As the lysogen grows better against the increasing concentration of chloroxylenol, it would suggest that the naïve host has little or no significant mechanism or response for chloroxylenol, due to its lack of generalised response under a recognised stress, and its lack of adaptation to increasing concentrations of chloroxylenol. Furthermore, as the naïve host grows less effectively than the lysogen in the presence of chloroxylenol, it supports the hypothesis that the lysogen's cell wall fatty acid changes play an important role in its observed resistance. Differentiation between the lysogen and the naïve host during spikes in fatty acid content are observed in the PLS-DA plot within Figure 4.14, however due to the complexity of the data this could not be validated (Table 4.1).

4.3.3.3.3 Lysogen and naïve host cell wall fatty acid differentiation under antimicrobial challenge over time

Figure 4.16 and Figure 4.18 (heatmap, PLS-DA, and biplot, respectively) further analyse the 18 hour cell wall fatty acid data under the two antimicrobials tested. The range of plots used were essential for the complexity of the data, and were necessary in identifying trends to form a hypothesis toward resistant strategies against either antimicrobials. Figure 4.16 supports the hypothesis that both the lysogen and the naïve host have completely disparate strategies in cell wall fatty acid content when grown in the presence of 8-hydroxyquinoline. The figure also supports the hypothesis that neither cultures had a cell wall fatty acid strategy against chloroxylenol, although the lysogen did respond in a generalised way with fatty acid changes, it was not significant enough to show complete separation between either cultures. Under chloroxylenol fatty acid changes appear to be influenced more by time and drug concentration than by phage infection.

The PLS-DA (Figure 4.17) elucidates grouping of any and all conditions. It confirms the differentiation between the lysogen and naïve host under 8-hydroxyquinoline, it also identifies

similarities not identifiable in other figures. The lysogens fatty acid responses to both antimicrobials group closely, suggesting a baseline strategy, from which it adapts uniquely depending on the drug. The naïve hosts unique fatty acid profile in Figure 4.13 to 8-hydroxyquinoline suggests a specific response, but its poor growth suggest a failure to recognise the stress type and has employed a mechanism designed for alternative challenges. If this is the case, then phage infection also appears to refine the hosts ability to identify and employ a more appropriate response to antimicrobial challenge. Furthermore, the naïve host does not adapt its strategy after failing to aptly respond to 8-hydroxyquinoline, this is seen in both Figure 4.13 (repeated pattern) and Figure 4.17 (tight grouping). Under chloroxylenol both cultures grouping in the PLS-DA show a lot of cross over, supporting the theory that while fatty acid cell wall structure may play a role in acquiring resistance it is not the mechanism of the resistance observed.

A biplot can help identify the fatty acids most influential in a given group, Figure 4.18 shows which fatty acid changes are associated to either antimicrobial present. Cyclopropaneoctenoic, dodecanoic, decanoic, hexadecanoic and pentadecanoic seem to be relatively important regardless of either antimicrobial. Fatty acids most significantly associated to chloroxylenol are: Nonanoic acid, 9-hexadecanoic acid, octanoic acid, octadecanoic acid, 9-octadecanoic acid, methyl-2-hydroxydodecanoate, hexadecanoic acid - 15 methyl. Fatty acids most significantly associated to 8-hydroxyquinoline are: Heptadecanoic acid ME, Butanedioic acid dimethyl ester, methyl-3-hydroxytetradecanoate, 13-Tetradecynoic acid ME, and Hexadecadienoic acid ME. The importance and role of these fatty acids can be seen in appendices Table 10.10. When looking at Figure 4.15 we can see that the reason for heptadecanoic acid as a differentiating feature in the lysogen is due to its increasing prevalence in the cell wall over time. Interestingly this correlates with the lysogens observed cell growth, as heptadecanoic acid can feed directly into the biotin pathway potentially explaining the reduction in cell wall heptadecanoic acid during early and mid-exponential growth phase. We are potentially seeing the re-purposing of heptadecanoic acid as one method of lysogen driven early growth. The VIP plot (Figure 4.15) identifies 9-hexadecanoic acid as having the greatest influence on differentiation between the lysogen and naïve host against incremental doses of 8-hydroxyquinoline over 18 hours. 9-hexadecanoic acids influence on differentiating between the lysogen and naïve host increases over the 18 hours, with its intensity

continually greater in the naïve host. This may be the result of fatty acid recycling in the lysogen, either for alternative/additional supplementation into the TCA cycle, or for extra/intracellular interference of the lipophilic antimicrobial.

4.3.3.3.4 Lysogen and naïve host cell wall fatty acid differentiation under 8-hydroxyquinoline challenge over time

In the interest of identifying the lysogens specific resistant strategy toward 8-hydroxyquinoline, Figure 4.19-4.21 focused on the changes in cultures under this condition. The condition specific heatmap (Figure 4.19) clarified some unique patterns in cell wall fatty acid changes. Though the naïve hosts response was to up-regulate the majority of its fatty acids, it showed no upregulation of methyl Z-11-tetradecanoic acid or heptadecanoic acid. Furthermore though the lysogen downregulated its fatty acids in gaining true resistance, conversely to the naïve host, it maintained its up-regulation of methyl Z-11-tetradecanoic acid. However it may be the cost of down-regulating the majority of other fatty acids. The role of either methyl Z-11-tetradecanoic acid or heptadecanoic acid within the cell wall structure is unknown. But the data suggests it plays a potentially significant role in initial antimicrobial tolerance.

We further focused on the 8-hydroxyquinoline condition, in the entirety of its 18 hour incubation, using PLS-DA. We show that the lysogens resistant strategy, as far as cell wall fatty acid content was concerned, was relatively consistent, where only sampling at 3 and 15 hours showed any deviation. This suggests that the lysogen adapted quickly to increasing concentrations, and employed an alternate resistant mechanism to that used in the first 3 hours of 8-hydroxyquinoline exposure. The lysogens resistant mechanism appears to be highly effective, as it does not resort back to its initial response for a further 12 hours of increasing doses. The mechanism of resistance, whether directly related to the fatty acid profile or not, is clearly an effective and novel lysogenic mechanism.

A biplot and VIP plot were used to identify the most influential fatty acids associated to differentiation between the lysogen and naïve host, and thus potentially improve our understanding

toward the mechanism of resistance (Figure 4.21). We found that 9-hexadecanoic acid plays the greatest role in the differentiation observed between the lysogen and naïve host. The down regulation of this fatty acid could play a significant role in the lysogens resistance mechanism. 9-hexadecanoic acid has an essential role in the membrane bilayer phospholipids (appendices Table 10.10). Hexadecanoic acid content is reduced in the lysogen when concentrations of 8-hydroxyquinoline increase. It may be that a reduction in hexadecanoic acid adjusts the ratio of the 3 main fatty acids involved in membrane bilayer phospholipids in a manner that improves antimicrobial resistance.

4.3.3.3.5 Lysogen and naïve host cell wall fatty acid differentiation under chloroxylenol challenge over time

Separate analysis of the chloroxylenol condition provided no evidence of similarity in lysogen and naïve host differentiation through the 18 hour experiment, supporting what's shown in Figure 4.16. This suggests that incubation time and chloroxylenol concentration have a greater influence on cell wall fatty acid profiles than phage infection of the naïve host in the presence of chloroxylenol. This implies that improved lysogen fitness is possibly associated to mutation and rate of growth rather than any unique or predetermined resistant strategy, supporting our previous hypothesis (chapter 3). Others research has shown that increased growth and cell stress can affect rate of mutation (Chou, Berthet et al., 2009, Fong, Marciniak et al., 2003, Nishimura, Kurokawa et al., 2017, Poole, 2012, Tenaillon, Denamur et al., 2004), both of which have been observed in the lysogen. We can hypothesise that mutation rates may be a secondary mechanism, providing a number of evolutionary advantages, that would include increased adaptability to antimicrobials in the absence of resistant strategies.

4.3.4 Conclusion

This study illustrates the complex and diverse strategies, in survival and propagation, that *E. coli* MC1061 gains from $\phi 24_B$ infection, where true antimicrobial resistance is only acquired in the lysogen. For the first time we show the specificity and extent of the lysogens increased growth, and confirm up-regulated gene expression of the biotin pathway. Possible similar mechanisms are used to manipulate of the biotin and fatty acid pathways due to their close relationship. Here we identify the lysogens novel subversion of the cell wall fatty acids and potential resistance mechanisms, as well as its increased gene expression in essential cell wall structures. Furthermore, data supports the previous hypothesis (chapter 3) of improved mutational adaptation in the lysogen. The study demonstrates the integral role phage have in antimicrobial resistance and propagation of host bacteria, highlighting the importance in understanding phage in the current struggle with antimicrobial resistance and disease progression.

Chapter 5. Comparative Metabolomics using: Cross Run Analysis for Comparable Compound Data profiling (CRACCD) and GUI interface

5.1 Introduction

5.1.1 Background of Metabolomics as a technique

The use of Mass spectrometry (MS) for quantitative metabolomic analysis of complex mixtures has been in place since the 1960s (Dalglish, Horning et al., 1966), with the term 'metabolic profile' coined by Horning et al. in 1971 (Horning & Horning, 1971). Since the 1970s there has been rapid advancement in both the throughput and sensitivity of MS, leading to its relatively recent increase in practicality for use in microbiological studies. Metabolomics is a fastly growing field of science, and vastly cheaper than other similarly expanding fields such as genomics. The key downside compared to genomics is the lack of free third party software and its availability. There are available tools for analysis, though some at high cost and limited tractable tools for handling data downstream of compound identification and upstream to visualisation and modelling of data. Tens of thousands of potential compounds are often identified per sample, per run, where a given experiment can require several runs and experimental replicates, which increases the difficulty in handling such large, scalable data. For more complicated multivariate studies, and in labs or groups with limited core bioinformatics, this often means laborious and time-consuming work. The common untargeted metabolomic analysis workflow is; Sample extraction or preparation which is usually solvent bases, MS, alignment, compound identification, secondary MS fragment pattern support for compound identification (optional), standards run of identified compounds (optional), compound statistics, and data plotting. The data and work within this chapter is directly associated to the LC-MS/MS technique.

5.2 Metabolomics and liquid chromatography mass spectrometry (LC-MS)

Metabolites are the intermediates and products of an organism's metabolism, including: cell physiology and morphology (Ni, Ghosh et al., 2017, Yao, Davis et al., 2012), growth (Holt, Lodge et al., 2017, Lee, Jenner et al., 2006), quorum sensing and biofilm formation (Koley, Ramsey et al., 2011), and antimicrobials and defence mechanisms (DeVuyst, Callewaert et al., 1996, Petatan-Sagahon, Anducho-Reyes et al., 2011, Vanalphen, Lugtenberg et al., 1979). Identifying metabolites, particularly from a relatively complex profile, can be difficult, with many factors to consider.

5.2.1 LC-MS

Metabolomics incorporates a diverse group of tools in data acquisition that include; Nuclear magnetic resonance (NMR) (Pan & Raftery, 2007), Gas chromatography mass spectrometry (GC-MS) (Koek, Muilwijk et al., 2006), liquid chromatography mass spectrometry (LC-MS) (Zhou, Xiao et al., 2012), capillary electrophoresis mass spectrometry (CE-MS) (Monton & Soga, 2007), mass spectrometry (MS) (Garcia, Baidoo et al., 2008), Fourier transform mass spectrometry (FT-MS) (Southam, Payne et al., 2007), and Matrix-Assisted Laser Desorption Ionization (MALDI) (Fraser, Enfissi et al., 2007).

LC-MS consists of liquid chromatography and mass spectrometry, with an interface between the two to allow the transition from a high pressure to high vacuum environment. There are several current interface methods, which include: electrospray ionisation (ESI) (Maxwell & Chen, 2008), atmospheric pressure chemical ionization (APCI) (Byrdwell, 2001), and atmospheric pressure photo-ionization (APPI) (Raffaelli & Saba, 2003). Liquid chromatography allows the physical separation of liquid components, while mass spectrometry measures the mass to charge ratio (m/z) of charged particles (ions).

The Q-Exactive mass spec unit used in this research consists of 10 major parts (see Figure 5.1), in order of function these are; heated electrospray ionisation unit (HESI), ion transfer tube, S lens, injection flatapole, bent flatapole, quadrupole mass filter, octopole, C-trap, orbitrap, and HCD collision cell.

A given sample is injected into the HPLC unit, in a mobile phase and enters the chosen column. Separation of compounds occurs as they pass through the column alongside mobile phase gradient changes. Compounds in solution (usually an acetonitrile/water mix) enter the interface (HESI unit) between the HPLC and MS unit. The HESI transforms ions in solution into ions in the gas phase by using electrospray ionization (ESI) in combination with heated auxiliary gas. The sample ions pass through the ion transfer tube to the S lens, where applied voltages to the S lens guides the ions via the injection flatapole into the bent flatapole where collision cooling takes place. Ions then enter the quadrupole mass filter, which filters ions based on their mass-to-charge ratio (m/z). The octapole brings the ions into the C-Trap where ions can be stored and directed to either the HCD collision cell or the orbitrap. The orbitrap is the mass analyser, which consists of a small electrostatic device into which ion packets are injected at high energies to orbit around a central, spindle-shaped electrode (Hardman & Makarov, 2003, Scigelova & Makarov, 2006). The frequency signal from the trapped ions is converted into a mass spectrum, which is a graphical representation of the intensity and m/z .

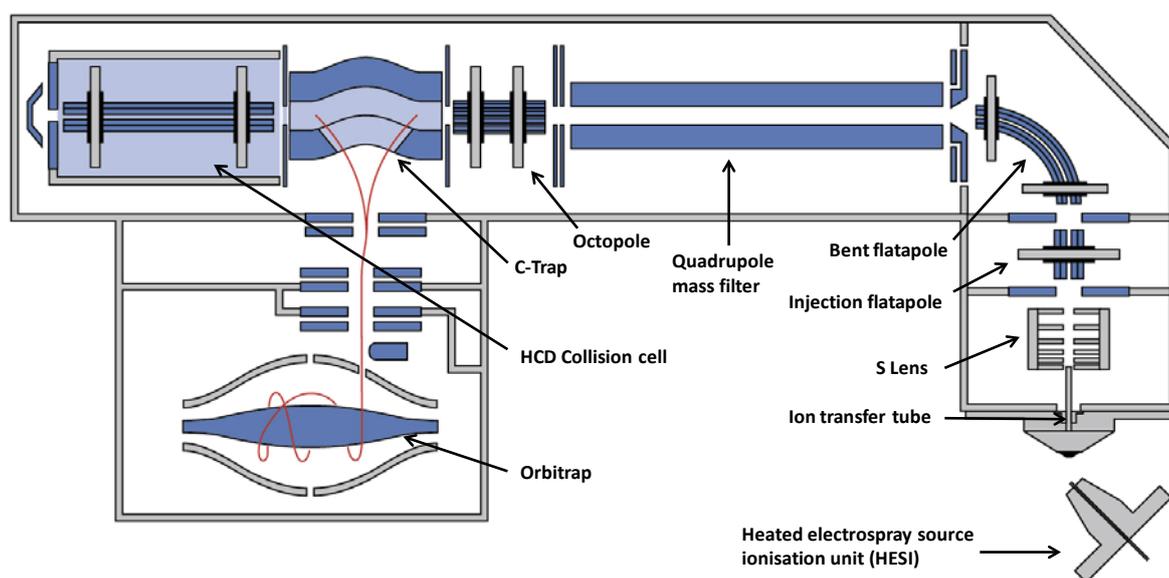


Figure 5.1 Q Exactive LCMS schematic. Illustrating the core structural components of the LCMS system that perform ion transformation, filtering, m/z determination, fragmentation and sorting, and secondary m/z determination. Edited from the Planet Orbitrap documentation(Orbitrap, 2018)

5.2.2 LC-MS/MS

Tandem mass spectrometry (MS/MS) is the process of selecting ions by their mass-to-charge ratio, fragmenting them, and analysing their individual m/z 's (Zhou, Song et al., 2005). This method provides far greater sensitivity to the LC-MS technique (Zhou et al., 2005). Many compounds can have the same m/z , especially when taking into account adducts resulting from the initial ionisation (Kind & Fiehn, 2006), but compound fragmentation patterns are far more unique.

The same initial process occurs as per described in section 5.2.1, the difference being that the orbitrap is used to select specific m/z ions (MS1) which are passed back to the C-Trap. The C-Trap then passes them into the HCD collision cell where ions are fragmented. The fragmented ions are separated and passed back to the Orbitrap via the C-Trap. The Orbitrap then analyses the masses of the fragmented ions (MS2).

5.2.3 Data creation, quality and issues

Metabolite sample preparation is background specific, with every case unique to the sample type, column and machine. The aim of the method is the avoidance of merged peaks and instead obtaining the clean separation chromatography peaks and resolution between compounds.

Data quality in metabolomics drives analysis and in particular, the reproducibility from the instrument, in this case the LC-MS. This is because it relies on the retention time and mass-to-charge ratio (m/z) when identifying the metabolites over several runs. There are numerous aspects that can impact the quality or reproducibility of the data generated by LC-MS, including and not limited to; bacterial culture and cell density, complexity of sample matrix, sample re-suspension solution, injection volume, column type, pressure, mobile phase, gradient, duration of run, separate or simultaneous +/- ion scan, and LC-MS type. It is therefore apparent that an assay is developed for a specific experiment based on these parameters and the depth of data required.

Sample preparation is an important step to refine, particularly when the sample is found in low concentration, even with evaporation/reconstitution steps the background will increase alongside the desired product. Bacterial metabolomics is one such example for low concentration and background noise associated problems. This is often due to both the consumables and large

volumes of culture needed for a weighable cell density and usable concentration. An example can be given in the necessary centrifugation, washes and resuspension of cells, as these steps are generally required to be carried out in plastics and bio-friendly solutions such as PEG and/or PBS. Contaminants such as PEG, PBS, and plastics, are less of a problem when working with quite concentrated samples, as they become background noise that can be negated with a negative control. However even with significant cell numbers, bacterial metabolites extracted from the cells are often low concentration compared to the background associated to their preparation. Where background contaminants are high, it's hard to confirm true identities within a sample, this causes further exponential error when searching for matching compounds from another run within the dataset (false +/- IDs). With proper care taken in sample preparation these problems can be reduced to produce usable and reliable data. Conversely at high concentrations the chromatography becomes swamped, compound retention times bleed into one another, and peak separation becomes impossible, further problems can also occur with sample carry over into the next sample run.

The more compounds present and their similarity to one another can make peak separation difficult, which in turn can lead to less reliable output from the program. A complex sample matrix often requires adaptation and refinement of the type of mobile phase, the mobile phase gradient, the sample run duration and the column type and size. In extreme cases a sample can be split into several groups that undergo different extraction methods prior to injection, reducing the complexity.

Further preparation techniques that can effect compound separation and equally the ability of the program to provide reliable analysis are; sample re-suspension solution used, the injection volume set, and the column used, the latter being the most influential. The re-suspension solution of a sample can affect the interaction with the mobile phase, as well as concentration of metabolites and adducts associated. Injection volume of the sample to pass through the column can affect the concentration and pressure. The most influential variable to retention time is the column, there are numerous column types for use in metabolomics, both normal-phase (NP) and reverse-phase (RP), each with different sizes in width and length, and further differences in pore sizes. Common column types used in metabolomics are RP columns such as C₁₈ and C₈ and NP columns such as

hydrophilic interaction liquid chromatography (HILIC). Each change in column affects the speed and type of compounds that pass through at a given time during a sample run.

Mobile phase is the solution the sample is mixed with and passed through the column (often acetonitrile), The mobile phase chemicals flow continuously through the column under a changing concentration gradient, this change in gradient can alter which compounds pass through at any given time. The run duration can be increased to allow slower changes in gradient, which can further improve compound separation. Differences to either the mobile phase or the gradient will make one run not comparable to another, this leads to datasets not functional for the program, fronting false negatives and positives.

5.2.4 Bioinformatics

There are numerous tools that can be used in the analysis of metabolomic data, such tools include: Progenesis software (<http://www.nonlinear.com/>), SIEVE (<https://www.thermofisher.com/>), MetaboAnalyst (Xia, Sinelnikov et al., 2015), *m/z*mine (Katajamaa, Miettinen et al., 2006), metAlign (Lommen, 2009), BinBase (<http://fiehnlab.ucdavis.edu>), xcms (Smith, Want et al., 2006), LC-MSWARP (Jaitly, Monroe et al., 2006), ChromAlign (Sadygov, Maroto et al., 2006), PETAL (Wang, Tang et al., 2007). Available tools cover core and essential analysis that include: peak alignment, compound identification, biomarker identification, and data plotting. To our knowledge the pooling of numerous separately aligned datasets, fixing of compound identifications and tabulation of compounds of interest spanning several datasets is a novel tool, that's not available with current programs. For those not learned in data handling code, CRACCD prevents the necessity of performing this simple but labour intensive step, particularly when alignment issues occur.

High-throughput data analysis, is often carried out using bespoke pipelines, built from coding languages that include; Bash/sh, Python, Java, html, R, Perl, C++. Languages most common to bioinformatics include Python, R, and Perl. Python, and Perl are widely used high level programming languages, most often used for general purpose programming, but also capable of statistical and graphical computing (Marino-Ramirez, Spouge et al., 2004, McKinney, 2010). The

R language is another high-level programming language, however, unlike Python and Perl, R is specifically designed as a statistical computing and graphics language.

The Bourne-again shell/Bourne shell (Bash/sh) is the oldest shell still in common use. In bioinformatics Bash/sh it is often used for simple pipelining between scripts. This is because it is not a high level programming language as it does not contain easy to use or automated systems that allow strong abstraction. It should be noted that the bourne shell along with standard unix tools such as awk, sed, and grep, actually contribute to the Perl language. However it is often not used as a core coding language in bioinformatics due to it having little support in complex math, performance speed and other high level language features. However, lack of support and difficulty of use should not be confused with an inability to be used for such needs. Building tools/programs in bourne shell code is generally more difficult than high level programming languages. However, it has a significant advantage for simpler installation and use, particularly as it is the generic shell language of linux and unix operating systems. Nonetheless, for the plotting of data it would be excessively impractical to use, as such, final plot/mathematical necessities are ideally outsourced to languages such as R.

5.2.5 Aim

As part of analysing the data in chapter 3 a subsequent aim focussed on the development of a platform that would limit this bioinformatics burden that could be offered in a free and open source manner. Here we demonstrate a novel program with a GUI interface; Cross Run Analysis for Comparable Compound Data profiling (CRACCD). CRACCD is designed to take batched, raw compound data from several separately aligned runs, identify compounds of interest, and cross compare the incidence and richness of these compounds over the entire dataset. This is then tabulated, after which plots of the data can be created from the table, or within CRACCD itself. This tool is particularly useful in metabolomic analysis. It allows the flow and change of compounds that are of interest in a given group to be monitored across several datasets, where data from different conditions were previously too dissimilar for pooling into a single analysis run due to alignment issues.

5.3 Results

Here we present CRACCD, a graphical user interface wrapping an interactive bash scripted pipeline for the acquisition and tabulation of matching compounds, particularly useful in separately aligned datasets. CRACCD was originally built for, and run on, LC-MS metabolomic data from chapter 3, to allow easier tabulation and further analysis. Due to alignment issues caused by sample variation the program was designed to bridge the gap in re-matching and tabulating compounds that have been identified using separate analysis too great for alignment to be possible. Alignment is an essential part of the analysis step, as without it, identical compounds can be missed. Variation in retention of identical compounds is due to a number of factors that include instrument conditions, change of LC-MS system, and underlying composition of a given sample. The m/z of a compound can also vary as a result of instrument noise. The effect this has on data means that two matching profiles can be slightly out of frame with one another, which if not aligned, will lead to identical compounds being missed, providing inaccurate/insensitive analysis. However if two profiles have enough differences, particularly where differences affects peak separation, alignment can become a problem (even though many of the compounds will have the same retention and m/z). When alignment is difficult the data has to be analysed separately, after which multiple tables can be made and painstaking, by eye, cross comparison is carried out to create a table that represents the compounds based on similar mass-to-charge ratio and retention times. In situations where a few hundred compounds are of interest (usually selected through p values, coefficient of variation percentage (CV%) scores, and MSMS confirmation), finding the same compound in a separately aligned and analysed dataset where it was not significant is laborious. This program aims to alleviate this caveat, and allows easy identification and tabulation of such data, based on the user given parameters.

CRACCD is built for linux operating systems, with simple installation, needing only the install file from which a single script is run in the terminal, requiring just 2 commands: `'chmod 755 ~/CRACCD_InstallFile/InstallCRACCD.sh'` (gives permissions to the script) and `'~/CRACCD_InstallFile/InstallCRACCD.sh'` (runs the script). The simplicity of installation is partly due to limited additional installation requirements, with the only other language required being the mathematical and data plotting language 'R' which is installed by CRACCD to reduce

complications and difficulties for the user. The rest of the program runs entirely in bourne shell (bash/sh), using bash as its sole language allows for simplicity in installation and use. Bash is relatively generic and is built across all linux systems, as such, the program can run with little issue or bugs on any linux operating system. This is in contrast to programs built with Python, which has numerous versions, each with enough differences to cause program errors if the wrong version of the language is installed. The code written thus far for the CRACCD program can be found in the appendices section 10.8.

5.3.1 Languages and tools used

Bourne shell was the sole language used for the functionality of the program outside of data plotting. Alongside bourne shell, the scripts were built using the GUI YAD tool and the standard unix tools that include: awk, sed, grep, cut, bc, and ls. The scripts were wrapped in a GUI using the ‘yet another dialog‘ (YAD) language. YAD is a fork of Zenity, which allows a shell script to interact with a GUI user. All high level mathematical and plotting necessities were written in R, with the most frequented packages being: stats, vegan, lattice, gclus, ggplots, and ggbiplot. All scripts that make up CRACCD can be found in the appendices.

5.3.2 Mapping the program interface and data sorting

The GUI interface workflow in Figure 5.2 demonstrates how the program is traversed by the user, it’s built so that input, data investigation, plot building, and output are centralised around the main menu. The buttons for each option are organised left to right in order of recommended program steps, to create a user friendly flow.

CRACCD GUI

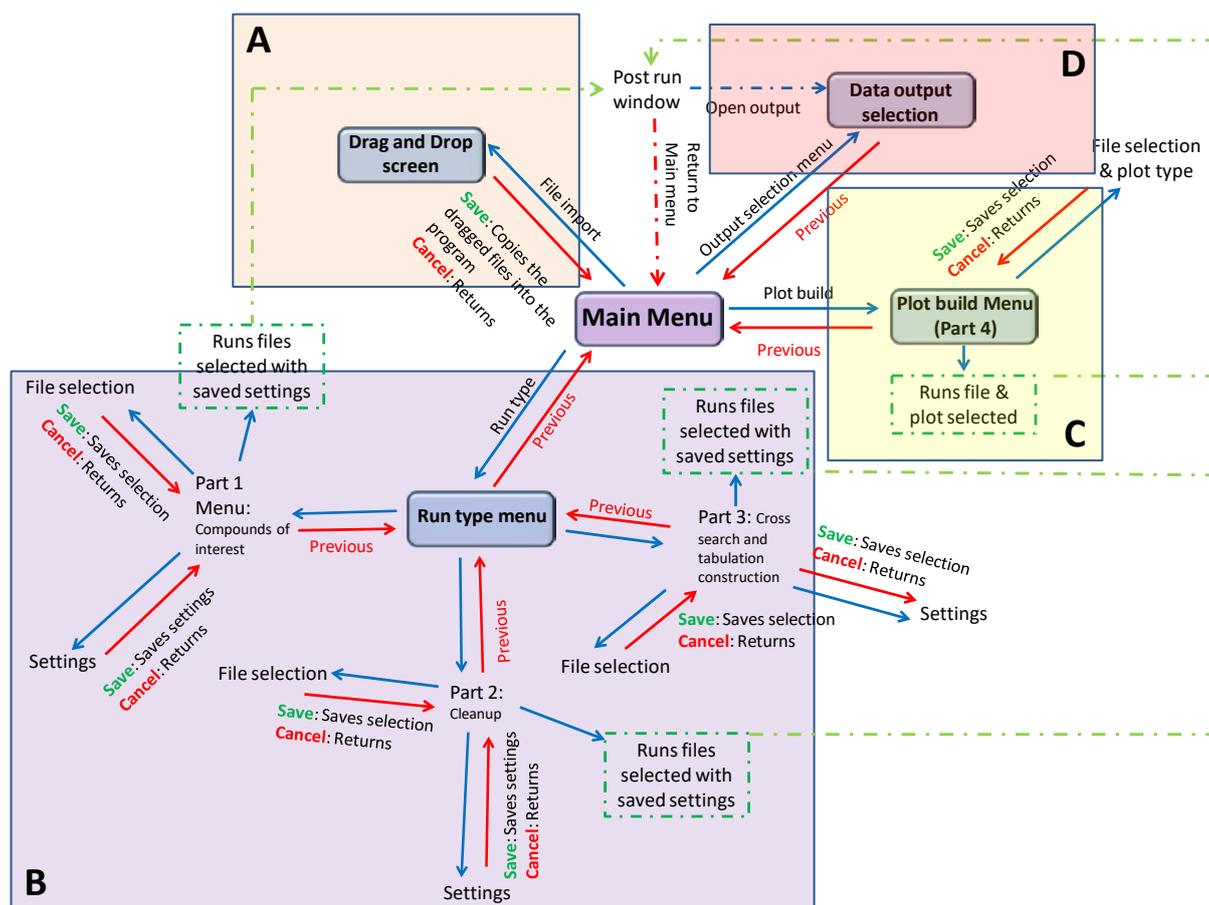


Figure 5.2 Map of the program ‘CRACCD’ GUI. The plot demonstrates the directional flow between menu screens, where blue arrows represent moving toward a window/tool, red arrows represent moving back from a tool/window, and green arrows represent a tool being used to produce an output. Directional workflow is shown A-D, where ‘A’ represents data importation, ‘B’ represents data sorting and tabulation, ‘C’ represents data plotting, and ‘D’ represent the amalgamation of data outputs.

Figure 5.3 demonstrates the generalised script structure and main processes that occur in CRACCD’s preliminary step ‘Step 1’. This step can be used to identify the ‘compounds of interest’ from their significant differences (p value) and their coefficient of variation percentage (CV%). The CV% is the ratio of the standard deviation to the mean = $\frac{SD}{X} * 100$. The CV% is particularly useful for identifying compounds with consistent replication in intensity, increasing the confidence in compounds detection. Other filter steps here include retention time and m/z ranges. A retention time range is often used to filter out known/visible problems with a run, the start of a run can be trimmed due to a number of reasons, the most common being the flushing/run-over of elution from

the previous injection run. The end of a run can be trimmed for a similar reason, where large quantities of mass can flush from the column at the end of the gradient. Compounds identified in either the start or end of a run that has these complications are unreliable. The m/z range can be used when looking for compounds with a certain mass-to-charge ratio, allowing the trimming of large and/or small m/z values that are irrelevant to a given study. The m/z range filter can also allow filtering for very specific compounds you wish to identify. It should be noted that this step is optional as the user can identify the compounds they are most interested in, by using alternative programs or tools, with methods the user deems most appropriate.

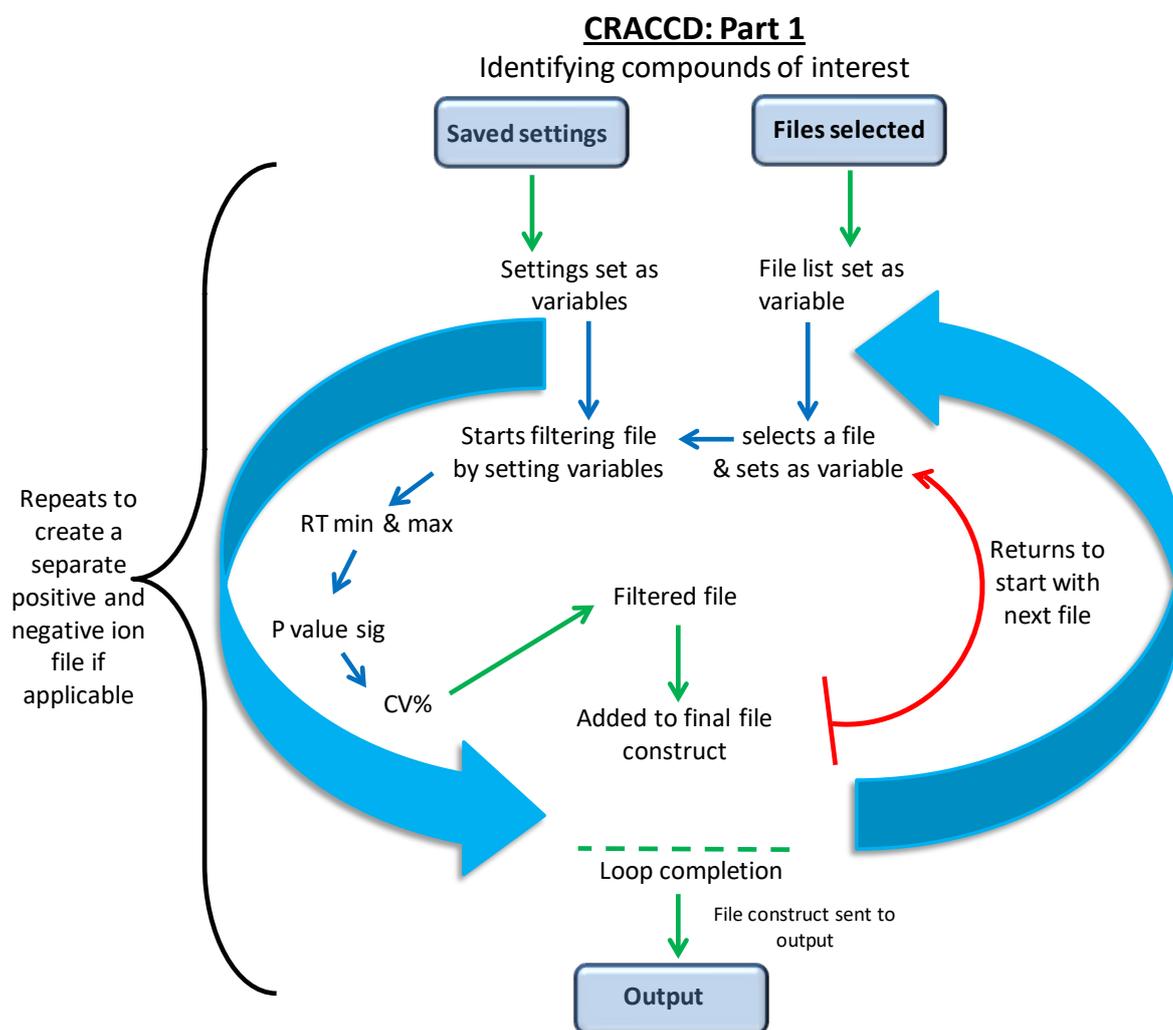


Figure 5.3 Map of the program ‘CRACCD’ method for preliminary compound gathering. The plot demonstrates the scripted order/construct of the code, applying and selecting compounds based on user input. The large blue arrows represent the major loop cycling within the script, the thin blue arrows represent movement of file selection and filtering steps. The red arrow represents the end of one loop cycle and directs back to where the next cycle begins. The black encompassing bracket represents the overarching loop of the entire script, which repeats for separate positive or negative ion file types.

Figure 5.4 shows the map of the script layout/method implemented in 'Step 2' of CRACCD. This map represents a simplification of the script built, displaying the core steps CRACCD takes in name cleaning of identified 'compounds of interest'. This step in CRACCD (Step 2) identifies if any compounds of interest found across different runs and conditions are actually the same compound, altering them to have the same arbitrary name. This clean-up of compound names means that downstream analysis does not treat the compound as a separate interest, but instead as a compound that displays a more constant significance between groups/variables.

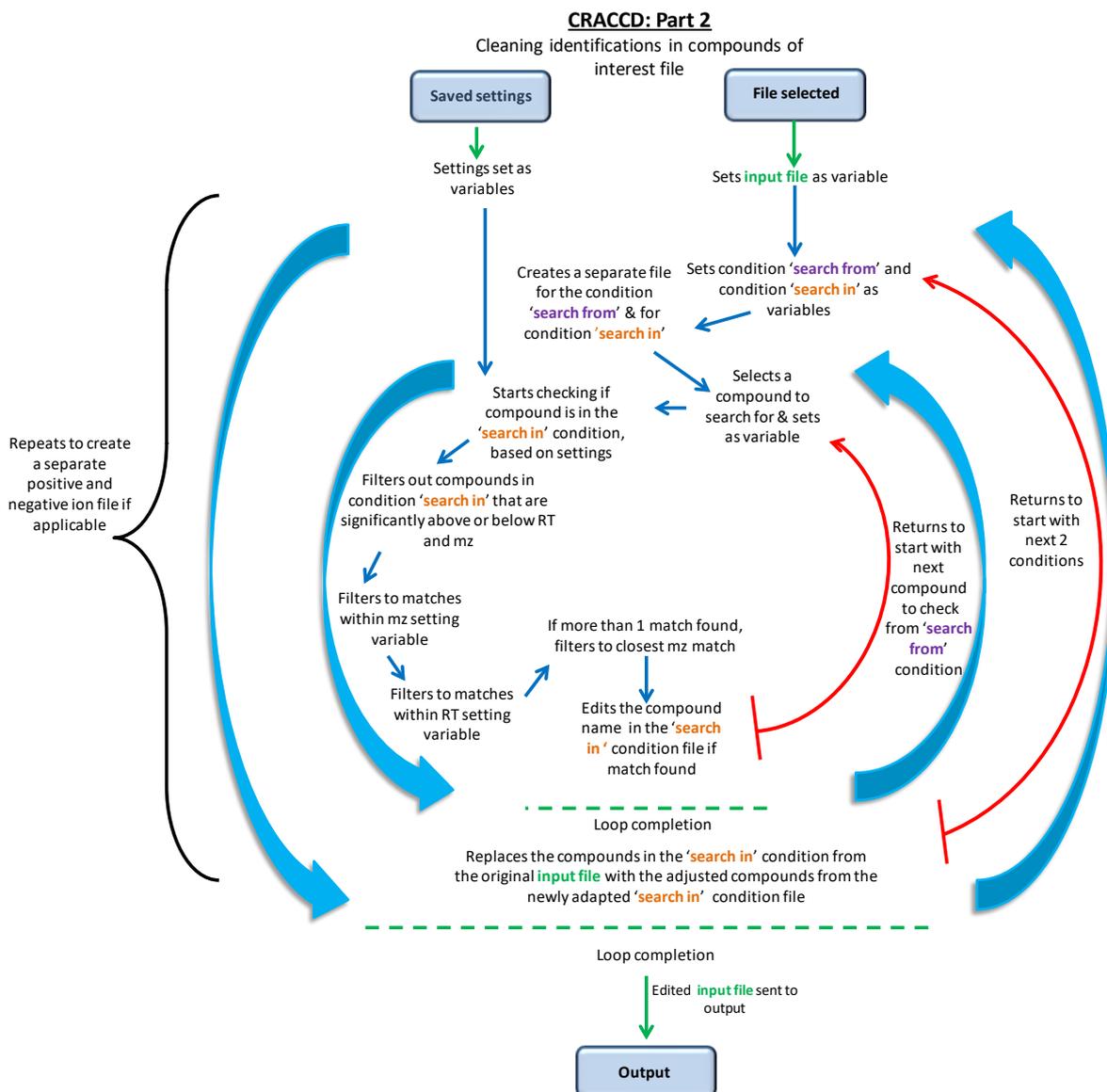


Figure 5.4 Map of the program ‘CRACCD’ method for assessing and altering compound identities when separate identities have been given for the same compound. The plot demonstrates the scripted order/construct of the code, identifying repeat compounds across conditions within the compounds of interest file, adjusting nomenclature appropriately (repeat compounds are given identical names). The large blue arrows represent the major loop cycling within the script. Large blue arrows within large blue arrows represent major loops within major loops. The thin blue arrows represent movement of file selection and filtering steps. The red arrows represents the end of one major loop cycle and directs back to where the next cycle begins. The black encompassing bracket represents the overarching loop of the entire script, which repeats for separate positive or negative ion file types.

Figure 5.5 displays the map of 'Step 3', the map shares many similarities to 'Step 2' as many of the core requirements of the step are the same, such as identifying compounds by m/z and retention. However, step 3 has added complexity, as the script is required to take several large input files into account, often 10s of thousands of compounds, which needs to be achieved at greater speed. Figure 5.5 also shows the final tabulation of the compound data, whereby the averages are calculated to simplify downstream graphical plots. It should be noted that a secondary table is produced that contains the replicates rather than the averages of each sample, allowing further analysis.

CRACCD : Part 3

Identifying compounds of interest throughout multiple datasets

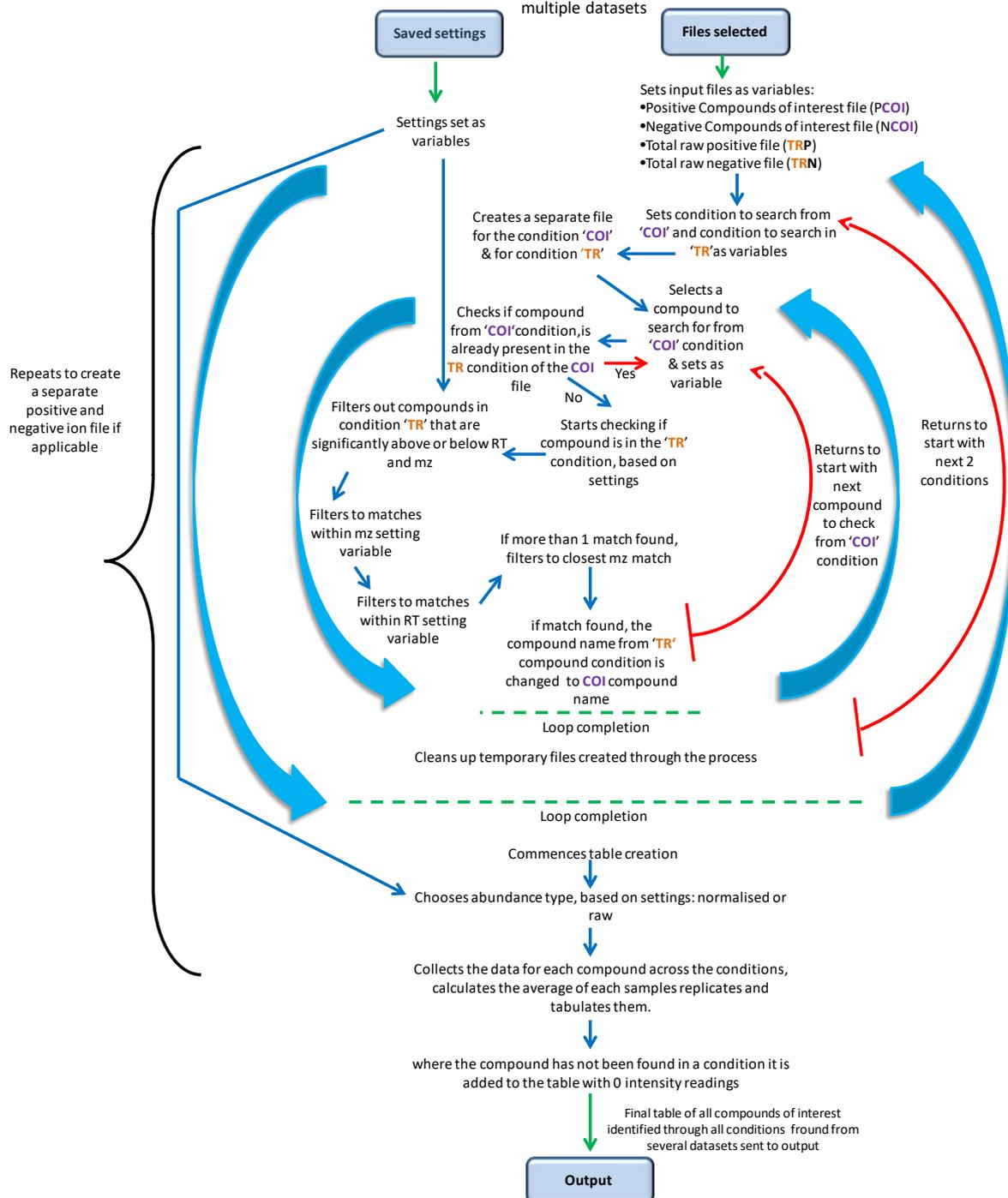


Figure 5.5 Map of the program 'CRACCD' method for identifying compounds of interest across datasets and conditions. The plot demonstrates a minimalised design of the script for identifying compounds of interest throughout all datasets and conditions, renaming them appropriately and tabulating the data. The large blue arrows represent the major loop cycling within the script. Large blue arrows within large blue arrows represent major loops within major loops. The thin blue arrows represent movement of file selection and filtering steps. The red arrows represents the end of one major loop cycle and directs back to where the next cycle begins. The black encompassing bracket represents the overarching loop of the entire script, which repeats for separate positive or negative ion file types.

Step 4 in CRACCD is the data plotting step, this step utilises either a user made table or the table from step 3 as input. The map of the scripts for step 4 can be seen in Figure 5.6, the map also includes the methods used within the R scripts of a given plot type. Some plots can be created from sub sections of a given R script, the map demonstrates the interplay between plot type and script deviation, as well as highlighting bash scripted manipulation to R scripts, to incorporate user specifications prior to being run.

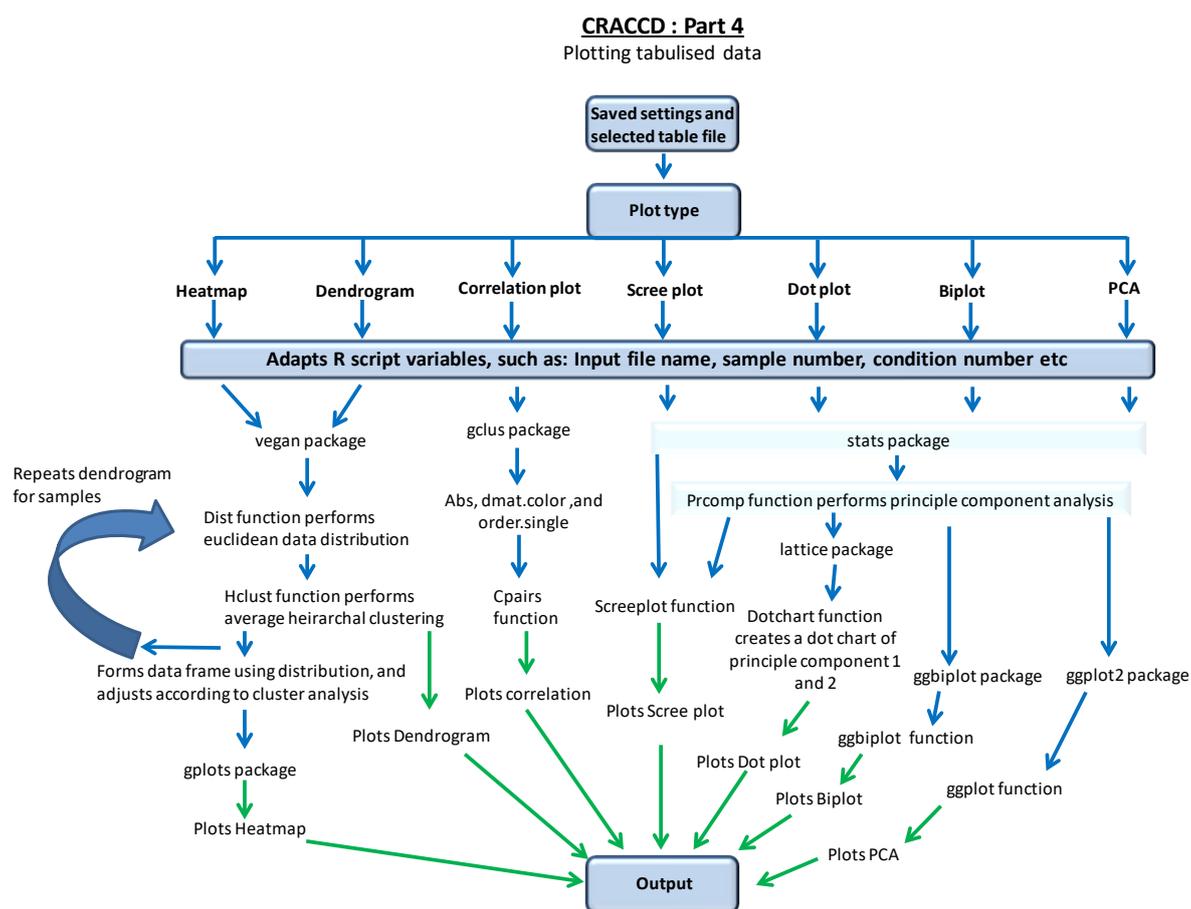


Figure 5.6 Map of the program ‘CRACCD’ method for plotting compound table. The plot demonstrates a minimalised design of the bash and R scripts for plotting the compounds table. Blue arrows represent the movement of plot selection and script selection, while green arrows represent plot creation and output storage.

5.3.3 Example data run through, demonstrating program settings, calculations and output

To show the transformation of data as it passes through each step, provide greater detail in the calculations and methods used in CRACCD, and to demonstrate functionality, an example data set has been created to pass through the CRACCD pipeline. It should be noted that the program does not specifically require data that has only come from previous steps in its pipeline. Data can be taken at any step in the program for separate analysis by the user if preferred, likewise data can be inputted at any step in the program, where the user has performed previous analysis on other platforms or pipelines. Nonetheless any data file used as input must be within the parameters of recognition by the program.

5.3.3.1 Data file formats and examples

CRACCD has been built to be relatively flexible to dataset formats, however a general format is advised. The input file format can be seen in Table 5.1. The example is a reduced dataset to ease explanation. The conditions observed in Table 5.1 can represent entire datasets and the pooling of conditions into a single file can often exceed the limit of simple user data handling tools such as excel. This step allows the filtering of vast datasets, where filter specifications can be provided by the user.

Its advised to keep to this format, however there is flexibility in how many blank lines that can be present prior to any data starting, as well as order or presence of either raw or normalised data. The format does not limit the number of replicates, each sample can have as many replicates as required, but each replicate column must be given the identical name. The position, but not names, of the columns 1: 'Compound ID', 2: 'm/z', 3: 'Retention time (min)', and 4: 'Condition' must remain the same. The columns 'p value' and 'CV%' can be any number of columns along, so long as they are at the end of the dataset. All compound names must be unique, this will be fixed later by the program if a compound happens to be the same but in a different condition. The file type can be either comma or tab delimited, for simplicity to the user it is recommended that it's saved as a .csv file.

Table 5.1 Example of an Input dataset.The table demonstrates the input data format for CRACCD,

this format is the same for both positive and negative ion files.

compound	m/z	Retention	Condition	Raw abundance									p value	CV%
				351	351	351	445	445	445	445	732	732		
187	140.1069	39.63347	Early	12365.92	31215.35	14383.66	11741.29	13143.74	12915.79	18300.74	14119.59	19079.84	0.0012	9.43
192	143.0838	3.854828	Early	145377.8	976421.6	947601.1	632549.6	962555.7	805382.6	877394.9	433697.3	874117.4	0.012	56.44
191	148.0603	13.26365	Early	17147.8	19702.64	15480.96	10211.93	17188.41	18619.29	21580.22	34924.4	26421.66	0.0001	9.43
193	137.0701	10.324	Early	375265.5	823518.3	380180.1	123147.7	808285.2	445906	103609.2	229055.3	334938.5	0.051	20..65
194	131.0565	16.79317	Early	610179	899510.2	452251.2	990424.1	956357.9	369415.9	844585.6	565344.6	364804.6	0.042	36.44
190	183.0802	35.69795	Early	11232.43	16517.53	15594.11	21417.68	19138.59	17314.89	20520.78	17418.33	21173.54	0.044	4.23
195	125.0429	23.26234	Early	412358.7	505684.9	540629.3	187403.7	989372.3	570225.4	771816.1	811312.8	986969	0.076	5.9
196	119.0292	29.73152	Early	808765.6	529150.8	383496.9	199064.6	950197.4	736472.3	597040	414586.6	879739.7	0.061	10.9
188	246.17	32.27078	Early	33235.09	24559.05	34555.08	38211.79	34110.11	37919.8	30850.22	30449.45	33166.17	0.02	3.54
189	334.2946	38.59403	Early	39327.15	33529.81	35486.78	11917.99	18012.18	12113.33	19490.53	12389.98	10257.23	0.001	7.12
197	113.0156	36.20069	Early	295769.6	437229.5	94759.15	913571.4	133880.3	132592.3	838355.1	61320.79	467083.5	0.041	55.41
198	107.0019	32.66986	Early	421796.5	2559.212	400164	10216.62	793526.1	598357.6	868726.4	799832.5	330037.9	0.12	15.32
199	100.9883	36.23069	Mid	298860.8	782944.5	312490.6	829934.4	678021.8	976400.8	222230.5	949950.7	484982.6	0.09	21.15
274	140.1064	39.63347	Mid	959071.1	901712.5	978615.8	42314.2	74010.37	65917.65	83980.27	87351.63	87248.89	0.047	6.44
276	94.97463	33.66986	Mid	674567.5	578226.8	235094.1	424077	180947.3	761655.8	465102.2	341678.7	739042.3	0.101	17.33
277	881.961	31.10903	Mid	846476.5	88216.18	179068	147017.5	567481.1	261456.5	312333.7	809198.2	341129.8	0.07	4.77
275	247.2273	4.225917	Mid	94364.42	89674.58	94400.51	12219.05	74216.98	54516.2	85585.44	86926.54	83691.32	0.007	3.87
278	821.9473	28.54821	Mid	336507.7	380021	254796.8	856856	829903.5	13086.74	686205.3	928873.5	542541.7	0.043	60.12
279	664.6227	7.794	Mid	79839.09	75662.87	74782.98	84515.74	79316.75	71617.68	89845.89	92963.71	84243.92	0.031	7.85
280	174.0874	15.1769	Mid	83025.59	81613.57	89143.54	75281.52	77113.63	78315.28	65556.52	86471.24	77014.9	0.003	8.77
301	676.9337	13.18324	Mid	379050.2	762091.1	586646.8	74954.61	336003.9	37822.87	658386.2	455078.3	768402.4	0.065	14.21
364	477.1758	9.544583	Mid	876122.4	894328.8	844616.7	81119.69	71810.74	84214.38	76318.07	86462.92	87429.31	0.001	7.34
370	194.085	5.3462	Stat	44580.65	41669.71	47493.34	49714.61	45116.67	47910.75	59429.2	59198.17	52564.14	0.0015	2.32
378	770.9201	10.62241	Stat	475479.2	783228.4	561902.6	869485.3	320588.6	838309.3	347767.9	684614.3	57624.69	0.098	31.21
379	264.9064	21.86572	Stat	841946.7	528865.4	267274.3	170165.6	922454.9	549752.3	580878.9	894694.5	80619.05	0.105	12.21
371	218.2114	39.99703	Stat	43487.48	46055.65	43543.86	56119.31	53214.61	47017.6	58698.3	41835.19	50126.39	0.003	3.24
372	152.1342	39.66643	Stat	66334.83	66541.59	69506.58	39318.32	46415.29	40610.6	55315.63	55646.74	53897.24	0.004	8.43
380	458.8928	18.30489	Stat	653579.2	280739.6	642455.5	395990.1	615798.8	881287.1	347626.2	907615.4	954825.6	0.07	12.34
381	152.8791	15.74407	Stat	774955.6	303491.4	110706.4	832876.7	202186.5	935748.3	201982	831969.5	219865.7	0.102	19.32
376	173.0921	1.389517	Stat	63417.14	61674.16	60450.86	47119.57	44512.25	41113.18	46209.86	50438.31	50868.45	0.042	9.11
382	46.86548	27.86572	Stat	816397.3	833244.9	33426.75	729872.1	556418.3	894138.4	379658.3	916493.6	619751.2	0.201	21.12
383	240.8518	5.744067	Stat	662269.9	990060.1	706658.4	229492	251110.8	203674.8	776036.6	292418.8	259414.5	0.092	33.3
384	534.8382	25.98738	Stat	409102.2	988191.4	512536.3	508699.1	326470.4	476586.6	834269.3	908196.3	56366.06	0.101	41.11
385	286.8245	23.42655	Stat	435705	558366.2	493748.2	333379.6	976988.1	808438.2	529437.2	51069.98	650489.6	0.08	18.22
377	140.1069	39.63347	Stat	64522.56	66732.38	65453.14	54213.27	55510.44	61413.47	61089.21	50916.72	60978.64	0.002	1.87
386	122.8109	20.86572	Stat	600702.1	744910.6	3010.589	617671.7	63195.05	404198.8	533568.9	352215.5	540297.7	0.101	31.11

5.3.3.2 Program part 1: Identification of compounds of interest

The programs ‘Step 1’ is used to identify the compounds of interest from a given dataset by filtering to specified p value, CV% scores, retention time range, and m/z range. This step can be carried out by the user or by the program, as small datasets do not require data handling from bespoke code, however for exceedingly large datasets coding is often the only way to manipulate such large file sizes. A brief explanation of each step is provided on the ‘Steps’ main menu, where any of the ‘Steps’ menus can be accessed. To open the ‘Step 1’ menu of CRACCD the user simply selects the associated button visible in Figure 5.7.

Figure 5.8 shows the ‘Step 1’ main menu and settings menu, the settings menu (Figure 5.8B) offers the available options previously described, for our example dataset we filtered by p

value, set at 0.05, and CV%, capped at 10 %. The settings were saved, and 'Run' was selected in the 'Step 1' menu (Figure 5.8A).

The general map of the script for identifying the 'compounds of interest' can be seen in the previously described Figure 5.3. This step is a simple filtering step, predominantly using the tool 'awk'. The script moves from retention and m/z ranges to p value and CV% filtering, using temporary files to store and pass through multiple filters. If the retention or m/z range is selected, the program ultimately uses the tool 'awk' to select any line that contains a number fitting the range given. There are a number of ways to achieve filtering with awk, the method use here is:

```
awk -F '\t' -v h=$MinNumber -v l=$MaxNumber 'BEGIN { OFS = FS } $2 > h && $2 < l {print}'.
```

The output is saved as a temporary file that can be filtered further if necessary. If specified, the compounds in the temporary file are further filtered by significance (p value), selecting lines containing values equal to or less than the specification (in this case 0.05). To achieve this, a similar 'awk' command as above was used and saved to a temporary output file. The filtering approach of p value is used to filter the CV%, using the previous temporary file as input. It should be noted that while this could all be piped from one to the other, problems can occur when data size is extensive, moving each time from a temporary file also allows simpler separation between filters selected.

The output from filter 'Step 1', using the example input data from Table 5.1, can be seen in Table 5.2. The number of compounds is greatly reduced, where compounds not meeting the search criteria are removed. This creates a 'compounds of interest' file, two 'compounds of interest' files are produced sequentially if both negative and positive ion data is inputted, which is ideal for processing positive and negative ion data simultaneously. The method importantly allows the data to remain separate to overcome complications in compound comparisons later in the analysis.

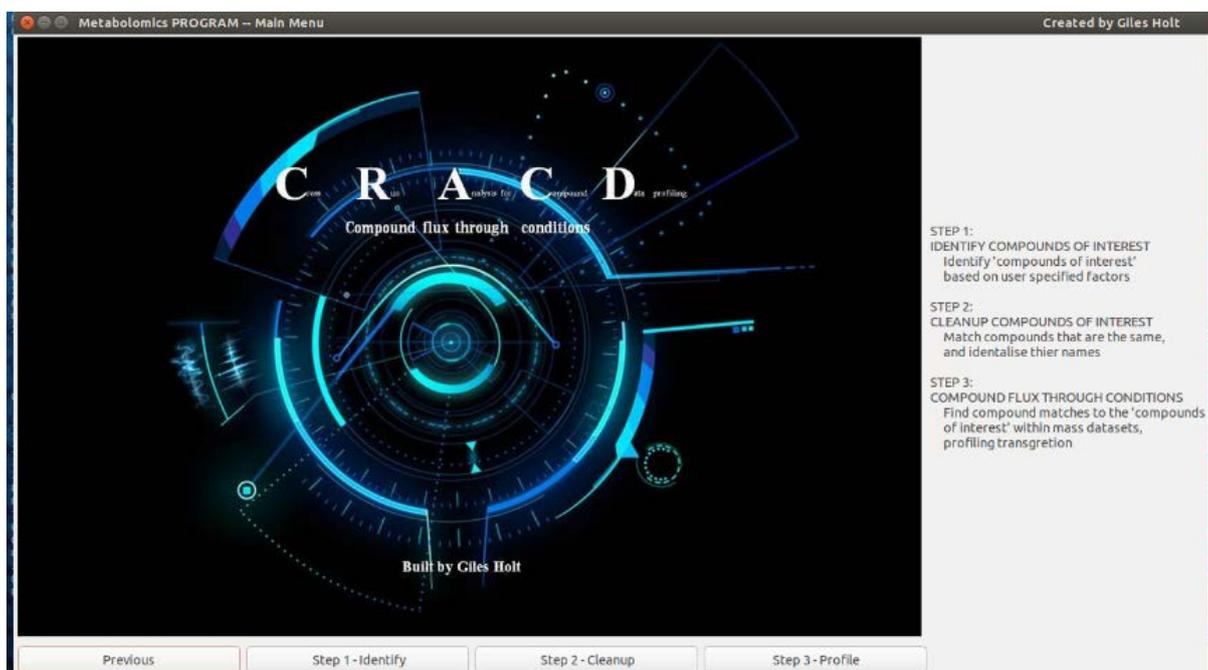


Figure 5.7 Program windows associated to CRACCD identification of compounds of interest. The image demonstrates the user interface of the program and available settings associated to step 1 of the CRACCD program.



Figure 5.8 Program windows associated to CRACCD identification of compounds of interest. The image demonstrates the user interface of the program and available settings associated to step 1 of the CRACCD program.

Table 5.2 Example of compounds of interest file. The table demonstrates the output of the 1st step in the program.

compound ID	m/z	time (min)	Condition	Raw abundance								
				351	351	351	445	445	445	732	732	732
187	140.10694	39.63347	Early	12365.9171	31215.346	14383.6595	11741.2909	13143.7407	12915.7871	18300.7418	14119.5899	19079.8394
188	246.17004	32.27078	Early	33235.0885	24559.0469	34555.0845	38211.7871	34110.1118	37919.8005	30850.2212	30449.4468	33166.1719
189	334.29456	38.59403	Early	39327.1488	33529.8072	35486.783	11917.9904	18012.1796	12113.3348	19490.5346	12389.9819	10257.231
190	183.08022	35.69795	Early	11232.4333	16517.5281	15594.1139	21417.6823	19138.5877	17314.8909	20520.7792	17418.3298	21173.5357
191	148.06027	13.26365	Early	17147.7955	19702.6421	15480.9607	10211.9322	17188.4129	18619.2908	21580.2179	34924.4046	26421.6561
274	140.10636	39.63347	Mid	959071.106	901712.47	978615.85	42314.203	74010.3662	65917.6525	83980.2652	87351.6264	87248.8861
275	247.22732	4.225917	Mid	94364.4219	98674.5759	94400.5114	12219.0531	74216.9783	54516.1961	85585.4416	86926.536	83691.3225
279	664.62268	7.794	Mid	79839.0933	75662.8687	74782.9784	84515.7428	79316.7494	71617.6764	89845.885	92963.7114	84243.9233
280	174.08737	15.1769	Mid	83025.591	81613.575	89143.5388	75281.5223	77113.6337	78315.2774	65556.5218	86471.2385	77014.8958
364	477.17576	9.544583	Mid	876122.393	894328.842	844616.739	81119.6894	71810.7415	76318.0695	86462.924	87429.3115	
370	194.085	5.3462	Stat	44580.6494	41669.7069	47493.3409	49714.612	45116.6735	47910.754	59429.2004	59198.1682	52564.142
371	218.21138	39.99703	Stat	43487.4759	46055.6464	43543.8604	56119.3077	53214.6114	47017.5985	58698.3045	41835.1949	50126.386
372	152.13422	39.66643	Stat	66334.8328	66541.5861	69506.5754	39318.3221	46415.2936	40610.5993	55315.6305	55646.7402	53897.243
376	173.09206	1.389517	Stat	63417.1386	61674.1553	60450.8648	47119.57	44512.2499	41113.1782	46209.8646	50438.3123	50868.4533
377	140.10694	39.63347	Stat	64522.5588	66732.3798	65453.1378	54213.2704	55510.4441	61413.4726	61089.2132	50916.7242	60978.64

5.3.3.3 Program part 2: Clean-up of compounds of interest

The programs ‘Step 2’ is used to clean-up compound names from the ‘compounds of interest’ input file, ensuring any re-occurrence of a compound across several conditions is correctly labelled the same. This step is impractical and time consuming if done by hand, especially when greater numbers of compounds are identified as ‘of interest’, coding is the only practical way to manipulate such large file sizes, which is achieved through the GUI. To open the ‘Step 2’ menu of CRACCD the user selects the associated button visible in Figure 5.7.

Figure 5.9 shows the main menu and settings menu for ‘Step 2’. The settings menu (Figure 5.9B) offers up to 2 input files, to allow for simultaneous positive and negative ion runs, and allows the user to set m/z and retention limits when classifying a compound. For our example dataset we set compound classification thusly: Retention time +/- ‘0.5’ minutes, and m/z match to a minimum of ‘1’ decimal place. The m/z decimal minimum should be set dependent on the extent a user trusts the accuracy of the machine used. In this case, the resolution of the LC-MS is particularly accurate (Thermo Q-exactive), however, the settings saved here err on the side of caution in the analysis. The settings are saved, and ‘Run’ is selected in the ‘Step 2’ menu (Figure 5.9A).

The General map of the script for the correction of ‘compounds of interest’ names can be observed in the previously described Figure 5.4. This step is more complex than the previous step, here conditions within the input file have to be separated and cross-compared to find if a compound of interest is the same as a compound of interest between conditions. The script achieves this based on the simple m/z and retention time rules previously set. The script pipes the data through filtering to check if the compound exists in another condition, it first identifies m/z matches to the decimal place set, and then takes those that fit within the retention time range given. If a single match is found at this point the code adjusts the name appropriately and moves to the next compound, if more than 1 match is identified then it takes into account the additional m/z decimal places and chooses the one with the closest m/z match. The program repeats this in a loop to the number of compounds found within the condition being searched from. At the end of a loop, it re-calculates which conditions to search from and which to search in. Therefore to be able to run an unrestrained and unlimited cross condition analysis the following equation was created: $((N - 1) \times 0.5) \times N$ (where N=the Number of conditions). Temporary files are used from one search to another within this step, this is done to prevent problems occurring with larger data sets, as previously described for ‘Step 1’.

The output from the clean-up ‘Step 2’, where example data from Table 5.2 has been used as input, can be seen in Table 5.3. Here the compound ‘187’ was identified in both the mid and stationary conditions under a different name, as such the program has correctly adjusted the name in both conditions to ‘187’. The program has now created a cleaned ‘compounds of interest’ file, two of these files can be outputted if two input files are provided. This allows for the running of positive and negative data simultaneously. This table can now be used in ‘Step 3’ without the problem of identical compounds being searched for throughout all the raw data, which saves time and reduces incorrect and cluttered output.



Figure 5.9 Program windows associated to CRACCD ‘compounds of interest’ clean-up. The image demonstrates the user interface of the program and available settings associated to step 2 of the CRACCD program. Image A: Step 2 menu, image B: Step 2 settings.

Table 5.3 Example of the cleaned up ‘compounds of interest’ file. The table demonstrates the output of the 2nd step in the program.

compound ID	m/z	Retenti on time (min)	Condition	Raw abundance								
				351	351	351	445	445	445	732	732	732
187	140.10694	39.633	Early	12365.9171	31215.346	14383.6595	11741.2909	13143.7407	12915.7871	18300.7418	14119.5899	19079.8394
188	246.17004	32.271	Early	33235.0885	24559.0469	34555.0845	38211.7871	34110.1118	37919.8005	30850.2212	30449.4468	33166.1719
189	334.29456	38.594	Early	39327.1488	33529.8072	35486.783	11917.9904	18012.1796	12113.3348	19490.5346	12389.9819	10257.231
190	183.08022	35.698	Early	11232.4333	16517.5281	15594.1139	21417.6823	19138.5877	17314.8909	20520.7792	17418.3298	21173.5357
191	148.06027	13.264	Early	17147.7955	19702.6421	15480.9607	10211.9322	17188.4129	18619.2908	21580.2179	34924.4046	26421.6561
187	140.10636	39.633	Mid	959071.106	901712.47	978615.85	42314.203	74010.3662	65917.6525	83980.2652	87351.6264	87248.8861
275	247.22732	4.2259	Mid	94364.4219	98674.5759	94400.5114	12219.0531	74216.9783	54516.1961	85585.4416	86926.536	83691.3225
279	664.62268	7.794	Mid	79839.0933	75662.8687	74782.9784	84515.7428	79316.7494	71617.6764	89845.885	92963.7114	84243.9233
280	174.08737	15.177	Mid	83025.591	81613.575	89143.5388	75281.5223	77113.6337	78315.2774	65556.5218	86471.2385	77014.8958
364	477.17576	9.5446	Mid	876122.393	894328.842	844616.739	81119.6894	71810.7415	84214.378	76318.0695	86462.924	87429.3115
370	194.085	5.3462	Stat	44580.6494	41669.7069	47493.3409	49714.612	45116.6735	47910.754	59429.2004	59198.1682	52564.142
371	218.21138	39.997	Stat	43487.4759	46055.6464	43543.8604	56119.3077	53214.6114	47017.5985	58698.3045	41835.1949	50126.386
372	152.13422	39.666	Stat	66334.8328	66541.5861	69506.5754	39318.3221	46415.2936	40610.5993	55315.6305	55646.7402	53897.243
376	173.09206	1.3895	Stat	63417.1386	61674.1553	60450.8648	47119.57	44512.2499	41113.1782	46209.8646	50438.3123	50868.4533
187	140.10694	39.633	Stat	64522.5588	66732.3798	65453.1378	54213.2704	55510.4441	61413.4726	61089.2132	50916.7242	60978.64

5.3.3.4 Program part 3: Finding and tabulating compounds of interest throughout all data sets provided.

The programs ‘Step 3’ is used to search a raw data set for compounds from a ‘compounds of interest’ input file, and produce a table representing the compound flux through conditions. To open the ‘Step 3’ menu of CRACCD the user simply selects the associated button visible in Figure 5.7.

Figure 5.10 shows the main menu and settings menu for ‘Step 3’. The settings menu offers up to two ‘compounds of interest’ input files and 2 dataset input files in which to search, this allows for simultaneous positive and negative ion runs. The settings menu demonstrated in Figure 5.10B allow the user to set the *m/z* and retention limits for identifying a compound of interest within the dataset. The final stages of ‘Step 3’ calculate the average of compound replicates to produce a table for further analysis. Where some input data may have both raw and normalised data, the program settings offer a choice of which data type to average and tabulate. For our example dataset we set compound classification: Retention time +/- ‘0.5’ minutes, and *m/z* match to a minimum of ‘1’ decimal place. The settings are saved here, and ‘Run’ is selected in the ‘Step 3’ menu (Figure 5.10A).

The General map of the script for tabulating the fluctuation in the levels of the ‘compounds of interest’ through conditions can be seen in Figure 5.5. This step is more complex than the previous step, as here conditions within the input file and additional dataset file have to be separated and cross compared to identify the appearance of the ‘compounds of interest’ across the entire dataset. The script determines each compound incidence through the dataset based on the user settings. Similar to ‘Step 2’ the script shifts the data through filtering techniques to check if the compound exists in another condition, the method for this is adapted for the larger scale of the search. To improve speed, this larger search removes m/z and retention times outside the range of compounds from the condition searched from, prior to more specific search/filtering. The program repeats in a loop to the number of compounds found within the condition being searched from. At the end of a loop it re-calculates which condition to search from and which condition to search in, an edited version of the ‘Step 2’ equation was made here to take into account the added complexity: $(NT - 1) \times N$ (where NT=Number of conditions in the total dataset and N=the number of conditions in the ‘compounds of interest’ dataset). Again temporary files are used from one search to another within this step, for reasons given previously.

The output from the compound intensities seen between conditions ‘Step 3’, where example data from Table 5.3 has been used as the ‘compounds of interest’ input, can be seen in Table 5.5. Two tables are output to allow the user to choose either averaged or non-averaged data for downstream applications. The dataset in which compounds were searched for, can be seen in Table 5.4. Here we see each compound of interest has been found across the majority of conditions within the Table 5.4 dataset, with conditions where compounds haven’t been found still tabulated but with ‘0’ readings for intensity, as seen for compounds ‘191’ and ‘280’. All compound names have been repeated and edited to account for the conditions. The program here has created a single tabulated file, 1 table is always output regardless of whether 1 or 2 input files are provided. 1 output file is created by adjusting the compound names from the associated positive or negative ion input to include a ‘P’ or ‘N’ respectively, this allows for the plotting of positive and negative data as one. The averaged table can be plotted in ‘Step 4’, or it can be plotted using other methods external to the program with relative ease, due to its simple layout and generic file type.

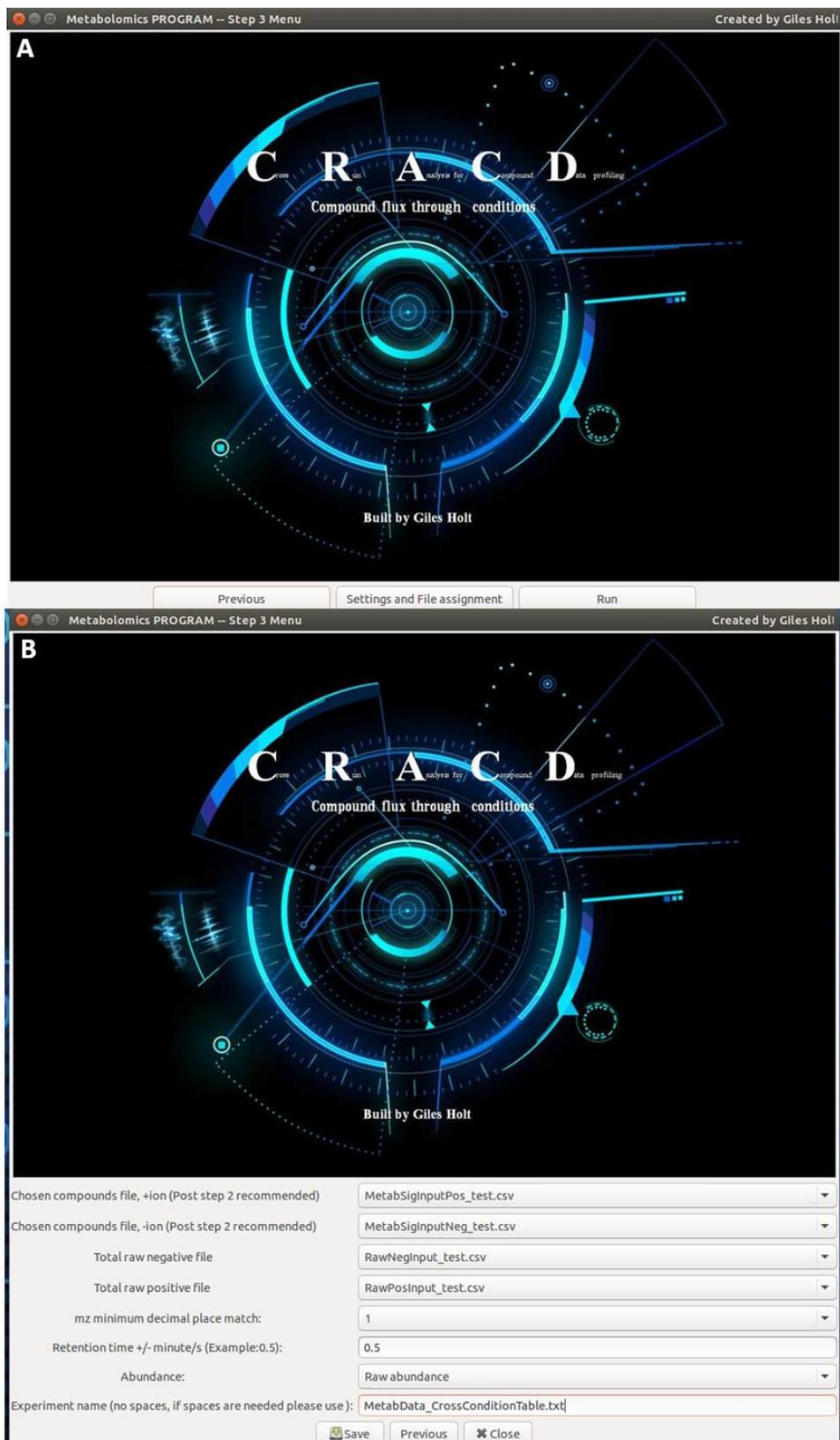


Figure 5.10 Program windows associated to CRACCD ‘compounds of interest’ profile tabulation of all datasets. The image demonstrates the user interface of the program and available settings associated to step 3 of the CRACCD program.

Table 5.4 Example total data set file containing all of the conditions. The table demonstrates a restricted mass dataset in which the compounds of interest are searched.

compound ID	m/z	Retention time (min)	Condition	Raw abundance								
				351			445			732		
				351	351	351	445	445	445	732	732	732
1093	194.08464	5.3462	Early	31440.0928	21208.0325	22113.0503	28713.8213	25717.8115	29114.6551	11979.1198	11936.6518	11560.3001
1094	234.08476	8.147133	Early	6855.06352	5619.41909	3932.67305	16858.5943	63016.5727	90412.67	2797.31328	1385.75978	5650.55748
1095	218.21198	39.99703	Early	25273.3917	24901.8077	24859.1309	14914.5705	22818.9082	17851.9727	13462.3747	13298.6863	12522.8481
1096	234.07613	8.704467	Early	20390.589	23265.8194	26707.4647	261851.42	288095.448	325661.809	204811.954	204645.869	209685.275
1097	477.17511	9.544583	Early	31257.4018	33894.9576	34850.8246	10118.2615	21615.8116	19919.6264	25604.5146	22263.4071	21166.2207
1098	152.1341	39.66643	Early	24234.5668	24312.3841	25335.5837	39317.334	39416.3673	38917.528	22395.8192	29087.2309	27897.2827
1099	228.1339	7.4613	Early	21589.4079	26253.1711	18737.7255	50244.3307	18756.0523	20205.7226	11275.3385	10881.3723	30046.19
1100	196.09677	7.4613	Early	528.231882	688.104283	0	62611.2369	19215.0533	0	2795.46248	0	492.78691
1101	346.11724	10.71923	Early	88594.7711	70767.907	80895.6424	117153.679	167987.348	213236.901	102150.04	72621.0852	52719.3447
1102	120.10692	39.63347	Early	47661.9632	21507.1154	49115.7825	30804.0128	51312.9677	45408.579	47517.1605	66341.1382	26701.1464
1103	247.22739	4.225917	Early	38039.7803	39775.0544	31306.9688	16511.8433	22814.803	16216.4475	27476.7443	26567.9228	28159.859
1104	118.08644	12.0164	Early	43567.8023	10390.0299	38756.634	98676192.3	45534365.1	43680048.9	36042330.5	27018053.7	67628261.9
1105	664.62268	7.794	Early	38451.1296	32640.8347	34898.7591	10912.2423	10912.9272	13510.1521	21092.2423	21984.1324	25324.6066
1829	173.09212	1.389517	Early	37322.7115	33280.5115	32341.3234	38412.7615	39619.1399	32219.2887	32288.5345	42124.031	22750.0793
9562	194.08497	5.3462	Mid	71075.8596	74578.9811	78118.2571	96811.1248	92313.5599	95216.3201	73900.5961	73902.3222	78697.5678
9563	328.22218	7.02865	Mid	103.291996	1548.46094	0	990.875166	921.525143	16914.4826	0	283.760415	623.740799
9564	218.21233	39.99703	Mid	23421.7384	28425.5692	31729.156	55318.4682	51911.4839	57314.0455	62804.9841	62621.0724	63556.6564
9565	246.17017	32.27078	Mid	90190.4952	91175.4192	91154.1716	70712.4358	75417.597	72016.4246	81528.4802	81943.4851	82083.8823
9566	138.09124	6.91155	Mid	55.5102234	539.879891	963.283643	11213.1759	0	14918.1058	0	322.696421	1001.65832
9567	334.29494	38.59403	Mid	96759.4587	96082.2016	90339.472	70814.7569	84418.4796	86710.0265	82029.8964	82595.5763	84280.9027
9568	152.13422	39.66643	Mid	80316.4161	79356.7955	76101.2537	88817.0277	89518.7349	81719.0998	97815.7544	81008.4782	91599.2113
9569	312.19483	6.8651	Mid	0	15805.4116	0	22650.8114	64817.0158	0	0	0	0
9570	209.99712	6.832533	Mid	2334.03461	1720.01846	2333.33359	18612.0841	19714.651	18910.3668	2557.4446	2481.58088	2440.61325
9571	183.08031	35.69795	Mid	96811.4379	97292.2796	97860.2266	76513.9334	79513.754	74918.4361	87844.7716	89641.4285	87274.291
9572	210.06281	6.7654	Mid	5619.12681	6606.6798	995.513942	299.602714	89.2369623	68611.0953	7346.5472	3727.76775	852.94445
9573	652.40244	6.7654	Mid	43929.1471	17550.2213	72003.2953	86285.2695	154144.524	102591.16	42764.6462	77264.9208	87513.1703
9574	116.07071	6.5869	Mid	266.789774	278.559113	243.593994	248.564818	273.720736	195.575519	235.577409	496.686627	281.444096
9575	173.09206	1.389517	Mid	74390.3579	74521.577	73362.2182	88617.3199	73212.9739	72715.3562	66477.5333	75003.8275	94755.3
10274	223.09614	0.5402	Mid	18291.7539	21765.9492	18992.165	15136.286	18786.0816	18534.9109	19206.3175	21834.9435	18226.6525
10275	100.07592	0.522217	Mid	1553.58859	1344.60699	1296.29413	10318.9826	11519.6604	14110.4811	1381.88742	1716.09099	1218.38429
10412	246.17003	32.27078	Stat	66157.3609	67583.637	64526.7886	40210.4717	48317.117	40910.8936	66493.0437	61183.5251	55028.5524
10413	664.62212	7.794	Stat	65302.9037	62964.8268	60762.97	45110.2308	43715.937	40214.0276	32793.6227	38914.957	34578.2886
10414	140.10634	39.63347	Stat	5513.67706	10408.1954	57355.8108	48233.3407	17596.664	72427.233	52687.3856	58202.6719	47501.1169
10415	251.15025	7.514583	Stat	13401.3428	36416.713	18950.3302	300536.321	163097.695	253573.291	195306.187	338160.518	173655.383
10416	387.06401	15.05228	Stat	41084.4628	30539.9373	47468.78	495003.467	310465.736	334507.443	299202.871	213733.045	388690.769
10417	334.29456	38.59403	Stat	63744.3568	64262.9084	60983.9553	40018.2713	50917.4445	41011.686	54231.8314	51242.6101	59098.3159
10418	741.4782	4.155883	Stat	6745.64286	5965.9646	2167.13554	22814.5071	0	1056787.74	0	14030.6447	3319.04616
10419	477.17578	9.544583	Stat	76451.015	71351.022	71421.011	53315.0599	51115.0332	53215.0123	64361.0499	64219.0244	63311.0491
10420	653.42672	4.170783	Stat	1909.53295	12512.5226	27992.4301	0	0	1813689.65	0	0	0
10421	195.12319	7.484717	Stat	375917.086	28061.8295	40373.7537	4241455.29	4529822.21	2526839.99	2874939.11	2836619.49	4223602.55
10422	609.39944	4.126033	Stat	0	13047.6525	20297.4307	0	0	1372430.14	1073.60687	0	0
10423	247.22731	4.225917	Stat	65383.7505	66040.2962	65610.8461	46618.5615	48516.8358	46212.5358	52923.3829	54466.3631	57526.6451
10424	183.08011	35.69795	Stat	64068.1832	66071.0595	66748.8521	41314.54	40814.8902	40712.6116	55260.5111	51966.6164	52952.8217

Table 5.5 Example of the final averaged tabulated data that represents the changes and appearances in the compounds of interest from all of the conditions. The table demonstrates the output of the 3rd step in the program.

compound IDCondition	351	445	732
P_187Early	19321.641	12600.27	17166.72
P_187Mid	946466.48	60747.41	86193.59
P_187Stat	65569.359	57045.73	57661.53
P_188Early	30783.073	36747.23	31488.61
P_188Mid	90840.029	72715.49	81851.95
P_188Stat	66089.262	43146.16	60901.71
P_189Early	36114.58	14014.5	14045.92
P_189Mid	94393.711	80647.75	82968.79
P_189Stat	62997.073	43982.47	54857.59
P_190Early	14448.025	19290.39	19704.21
P_190Mid	97321.315	76982.04	88253.5
P_190Stat	65629.365	40947.35	53393.32
P_191Early	17443.799	15339.88	27642.09
P_191Mid	0	0	0
P_191Stat	0	0	0
P_275Early	36373.935	18514.36	27401.51
P_275Mid	95813.17	46984.08	85401.1
P_275Stat	65678.298	47115.98	54972.13
P_279Early	35330.241	11778.44	22800.33
P_279Mid	76761.647	78483.39	89017.84
P_279Stat	63010.233	43013.4	35428.96
P_280Early	0	0	0
P_280Mid	84594.235	76903.48	76347.55
P_280Stat	0	0	0
P_364Early	33334.395	17217.9	23011.38
P_364Mid	871689.32	79048.27	83403.44
P_364Stat	73074.349	52548.37	63963.71
P_370Early	24920.392	27848.76	11825.36
P_370Mid	74591.033	94780.33	75500.16
P_370Stat	44581.232	47580.68	57063.84
P_371Early	25011.443	18528.48	13094.64
P_371Mid	27858.821	54848	62994.24
P_371Stat	44362.328	52117.17	50219.96
P_372Early	24627.512	39217.08	26460.11
P_372Mid	78591.488	86684.95	90141.15
P_372Stat	67460.998	42114.74	54953.2
P_376Early	34314.849	36750.4	32387.55
P_376Mid	74091.384	78181.88	78745.55
P_376Stat	61847.386	44248.33	49172.21

5.3.3.5 Program part 4: graphing the tabulated data

The program graph types for plotting tables of data that are formatted in the manner shown in Table 5.5 can be found in Table 5.6 and associated R packages are shown in Figure 5.6. ‘Step 4’ main menu is accessed separately from the CRACCD main menu. The ‘Step 4’ main menu can be seen in Figure 5.11A, to open the ‘Step 4’ settings (Figure 5.11B) of CRACCD the user simply selects the associated button visible in Figure 5.11A. The settings menu offers seven different plot types (Table 5.6) that can be built from a single selectable tabulated input file. The settings are saved here, and ‘Run’ is selected in the ‘Step 4’ menu (Figure 5.11A).

The output from the ‘Step 4’ data plotting can be seen in appendices section 10.4. The previous example data was simplified to show the transitional states between steps, as such it’s not suitable for use in representing the plots. The data used to demonstrate the plot types is based from a portion of the fatty acid study previously presented in chapter 4. The dendrogram plot is useful for visualising the dissimilarity between nodes/variables. CRACCD uses the ‘vegan’ package in R for this. The script reads in the table provided and performs euclidean data distribution and hierarchical clustering, using vegans ‘dist’ and ‘hclust’ functions respectively. The heatmap plot is useful for manually visualising patterns as well as supporting any discrimination observed in a PCA. CRACCD uses the same steps required to produce the dendrogram of compounds (rows). To create the heatmap it goes on to use the ‘dist’ and ‘hclust’ functions from the ‘vegan’ package to form the distribution and hierarchical clustering for the dendrogram of samples (columns). Both dendrogram information is then fed into the ‘heatmap.2’ function from the gplot package to produce the heatmap. The correlation plot is used to investigate the dependence between multiple variables at the same time and to highlight the most correlated variables. Variables with higher correlations are closer to the principal diagonal. The table is read into R and using the gclus package, the functions dmat.color and order.single were used to calculate the color matrix based off of the dissimilarity and the ordering of objects so that similar object pairs are adjacent, respectively. This was fed into the ‘cpairs’ function, which was used for drawing the scatter plot matrix. The programs scree plot is useful for showing the fraction of total variance in the data, which allows the identification of which components to use in explaining the data. The scree plot

shares the same initial analytical methods as the biplot, dot plot and PCA, in that the principle components need to be ascertained. Here it plots the variances against the number of the principal components. The table is read into R and using the function 'prcomp' the principle components are calculated. Using the principle component data the scree plot is created using the 'screeplot' function. The programs dot plots are useful for explaining the extent of each variable's effect on either principle component. To create the dot plots the program uses the same initial step to calculate the principle components as previous plots, it uses the principle component data as input for the 'dotchart' function in the 'lattice' package. The program PCA biplot is useful for visualising multivariate data, as it combines the variables, subjects, and principle components, showing both the separation between groups and the extent a given variable is influencing said separation. The biplot shares the same initial principle component calculations as the scree plot and PCA. To create the biplot the script uses the principle component data calculated to build the biplot using the ggbiplot package and function. The program 2D PCA plot is useful for visualising 2 principle components of a multivariate dataset, often 2 components are enough to explain the majority of a dataset, identifying which two components can be done using alternative plots such as a scree plot. The PCA plot uses the same principle component data as previously calculated. The principle component data is used as input for the 'ggplot' function in the 'ggplot2' package, where it plots the principle component 1 and principle component 2 as x and y axis respectively.



Figure 5.11 Program windows associated to CRACCD plotting tabulated compound data. The image demonstrates the user interface of the program and available settings associated to step 4 of the CRACCD program.

Table 5.6**List of plots that can be graphed in CRACCD**

Plot Type	Brief description
Dendrogram	Each compound is arranged along the bottom of the dendrogram, where similarities between these nodes is viewed as clusters, with each clusters similarity to another represented by the distance/branch separation between them.
Heatmap	The sample type is labelled along the bottom of the heatmap, the compounds are labelled along the right of the heatmap, with corresponding dendrograms opposite the given variable type. The intensity of a given compound is represented with its associated gradient along a color scale, where blue, white and red represent low, middle and high intensity respectively.
Correlation plot	Variables with higher correlations are closer to the principal diagonal, with color dictating the size of the correlation
Scree plot	The y axis shows the eigenvalues and the x axis shows the number of factors
PCA Dot plot	The x axis represents the loading values for a given variable, the variables are shown along the y axis
PCA biplot	The x and y axis represent the 1 st and 2 nd principle components respectively, the circle is the unit circle and the arrows represent the influence of a given variable upon a sample/group.
PCA	Plotted on x,y,-x,-y axes

5.4 Discussion

This research offers a novel program for handling and comparing metabolomic data. It importantly tight binding of DNA to intracellular proteins may cause translocation via a ‘Brownian ratchet mechanism’ of compounds, which can reduce total detailed searching by tens of thousands. This removal of outliers can be performed with greater simplicity and speed than allowing the search for very specific m/z and rt values throughout the entire dataset. This step significantly improved the speed of analysis.

There are numerous tools available in the analysis of metabolomic data, each provides either pipelined or individual specialist tools. Available pipeline tools can take a user with relatively straight forward data (similar and alignable) from raw or partly processed to plotted data (see section 5.2.4). Specific tools offer core individual jobs such as alignment, compound identification, and $msms$ fragmentation (see section 5.2.4.). None of these tools are completely comparable to CRACCD as they are all core function for metabolomics analysis. CRACCD is designed for input specified searching and filtering by m/z and retention on a mass scale using several input files, this is a relatively simple task that’s exceptionally labour intensive when searching this data for the first time or infrequently.

To aid use of the program we deemed a graphical interface would aid community uptake. For this reason CRACCD is wrapped within a GUI using the ‘Yet another dialog’ (YAD) tool (Figure 5.2 and Figures 5.7-5.11), this tool interfaces directly with the bourne shell and thus the linux terminal. Simple installation is another factor taken into consideration, which is achieved through scripting almost entirely in the bourne shell language, outsourcing only the final plotting steps to ‘R’.

CRACCD’s greatest use is in datasets that require splitting into several different datasets due to alignment issues. CRACCD can piece data back together and tabulate it after individual datasets have been separately analysed and compounds of interest separately found. Pooling and sifting through datasets manually can be an impossible task in utilities such as excel when the data files are of a significant enough size, as tools such as excel have a limited number of rows/columns that they can accommodate for visual filtering. As such, in particularly large and complex studies,

coding and alternative tools are the only way to achieve analysis. When scientists lack the resources to analyse their data quickly and effectively, it costs time that could be used more appropriately in their field of study. Gaps in free analytical/streamlining tools lead to necessities in learning code to achieve goals, this is impractical as coding is a skill set in itself with its own field.

Currently CRACCD is not a standalone tool for metabolomics, as it requires data preparation in other programs. With time, future additions to CRACCD will include data alignment, calculations of p values and CV%, and an automated system for prior pooling of an unlimited number of datasets.

It should be noted that CRACCD cannot be reliably used across multiple datasets originating from alternate protocols. It relies on the repetition of an identical protocol, as this ensures that alignment issues are due to sample compound differences, meaning compound similarities will still be identifiable at the same m/z and very similar retention times. Its due to the rigour required in the exact replication in data acquisition, that this program can function accurately. It should be noted that, in principle, the program would work on any data using mass and retention time. However thus far it has only been tested and run on LC-MS data.

This study illustrates a novel and basic tool in metabolomic data analysis, reducing the manual computational time required in more complex studies. Separation at each step allows the user to place input or take output at any stage in the process, providing flexibility in use with other analytical methods. CRACCD was originally built for, and essential in, the metabolomic analysis within chapter 3 of this thesis, but its speed and user friendly interface was added to make it more transferable to other researchers. Every tool built to replace manual computational analysis increases the time and focus that can be given to the biology of a given investigation.

Chapter 6. Graphical User Interface (GUI) for Genomic analysis

using Open Source Software (GGOSS)

6.1 Introduction

The “nuclein” now termed DNA, was first discovered in 1869 by Friedrich Miescher (Dahm, 2008). This led to further developmental research by Rosalind Franklin (Maddox, 2003) that led to the discovery and publication of the structure of double stranded helical model by Watson and Crick in 1953 (Watson & Crick, 1953). DNA research has since been an integral part of most biological fields, and after the completion of the human genome project in 2003 (Collins, Morgan et al., 2003), genomics has developed rapidly. Moving from Sanger sequencing (Sanger & Coulson, 1975, Sanger, Nicklen et al., 1977) through to massive parallel sequencing (Rogers & Venter, 2005) and more recently long read, real time, single molecule sequencing (Berlin, Koren et al., 2015, Chaisson, Huddleston et al., 2015, Clarke, Wu et al., 2009).

6.1.1 Genetic repositories and databases

The bioinformatics industry is at an all-time high with a proposed £4.39 billion turnover globally in 2016, and a projected rise of over 250% by 2021 (Markets, 2016). The data output generated from genome sequencing techniques is increasing at an exponential rate, with a projection of 2-40 exabyte/year from 1 zetabases/year by 2025 (Stephens, Lee et al., 2015). The cost per megabase of data has been increasingly more affordable, with recent costs just 0.003% of what they were a decade ago, this rate of advancement is far greater than the rate of computational advancement see Figure 6.1 (KA, 2017). As such, the necessity for mass scale and affordable genomic analysis programs, and storage of data has increased (Muir, Li et al., 2016). Trends in sequencing, data output and publications can be seen in Figure 6.2, where even individual projects (admittedly large scale) are shown to produce tens to thousands of terabytes of data. A significant amount of genomic data is made collectively accessible by the National Centre for Biotechnology Information (NCBI) via the Entrez database (Schuler, Epstein et al., 1996), which contains 39 databases made up of over 1.7 billion records, the genomic databases and their number of records can be seen in appendices Table 10.11.

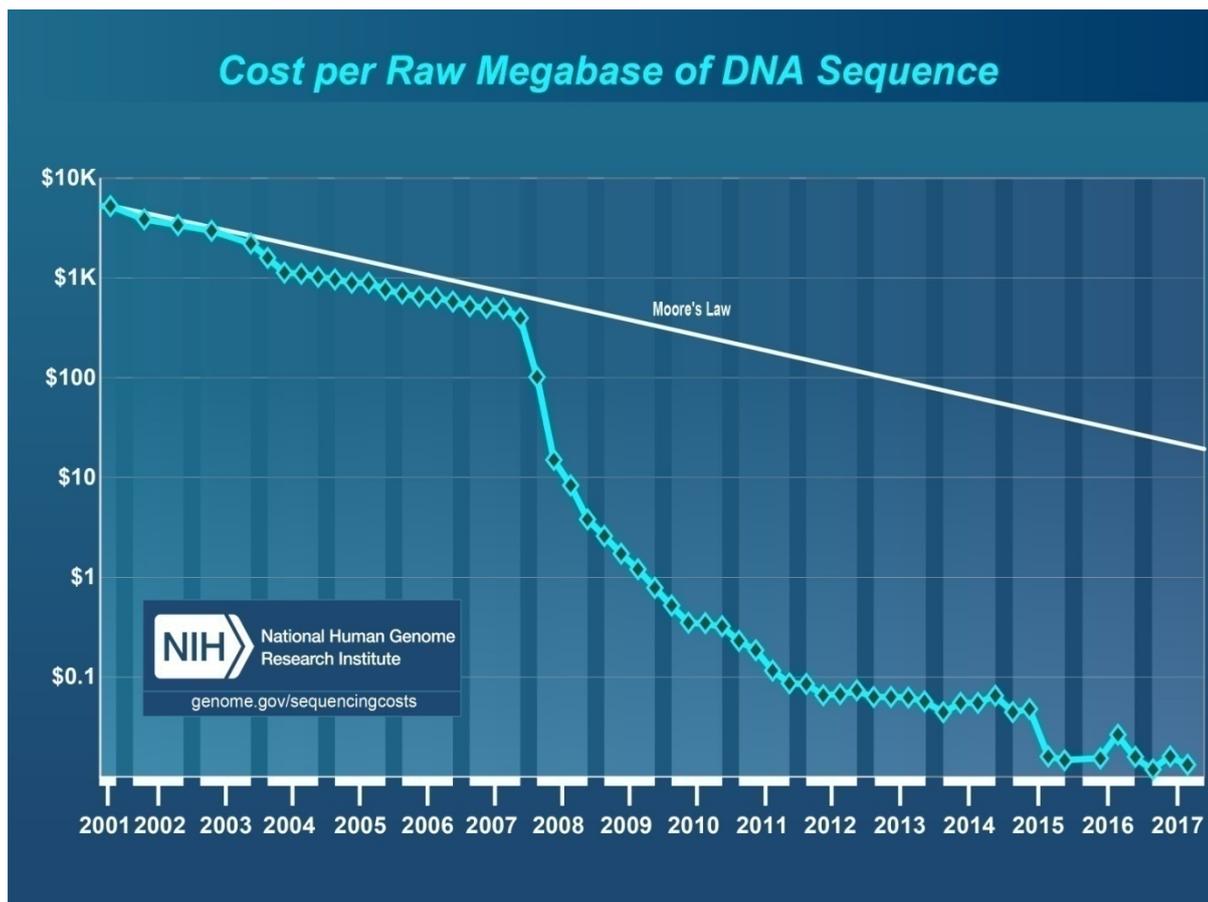


Figure 6.1 Cost per raw Megabase of DNA sequence compared to computer hardware related **Moore's law calculations.** The white line plotted on the graph is hypothetical data reflecting Moore's Law, which describes a long-term trend in the computer hardware industry that involves the doubling of 'compute power' every two years. Technology improvements that 'keep up' with Moore's Law are widely regarded to be doing exceedingly well. The sudden drop in cost after 2007 represents the transition from Sanger-based (dideoxy chain termination sequencing) to 'second generation' (or 'next-generation') DNA sequencing technologies (KA, 2017).

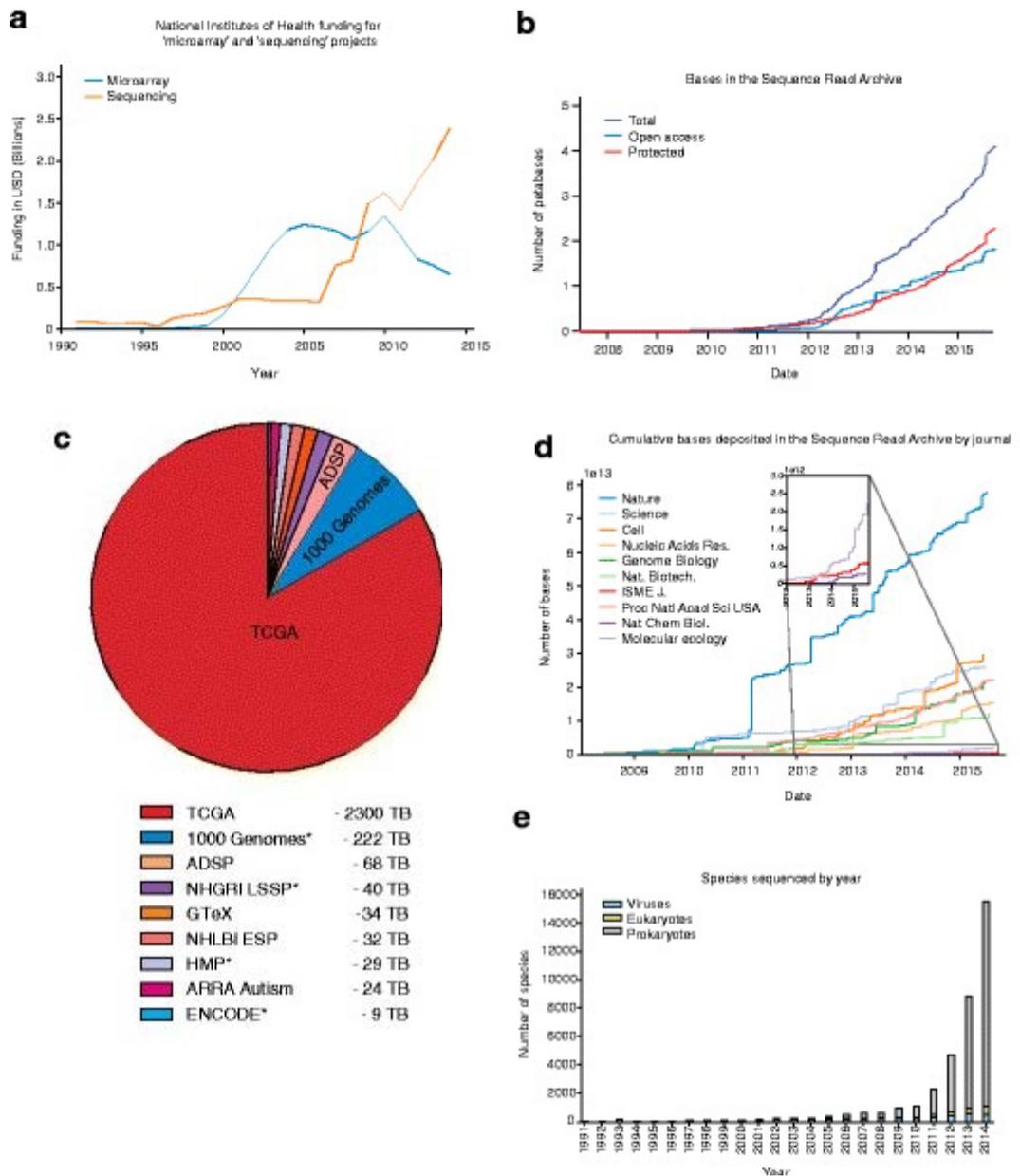


Figure 6.2 The Increase in the rate and amount of sequencing (Muir et al., 2016). **a:** graph of National Institutes of Health (NIH) funding related to the keywords “microarray” and “genome sequencing”. **b** a plot demonstrating the size and growth rate of the Sequence Read Archive (SRA). **c** Indicates the size distribution in data output of sequencing projects. **d** Plot depicting the cumulative number of bases deposited in the SRA and linked to papers appearing in different journals provide a proxy for sequencing adoption. **e** Plot depicting the number of species of each kingdom sequenced by year. Data was obtained from GenBank (Muir et al., 2016)

6.1.2 Genomic bioinformatics

Comprehensive analysis can cost almost as much as the sequencing itself, which has led to many attempting analysis themselves. This can again be at cost, through the purchasing and updating of expensive platforms such as CLC workbench (CLC Bio), DNAnexus (DNAnexus, Inc., Mountain View, USA, <http://www.dnanexus.com>), Geneious (Kearse, Moir et al., 2012), DNASTAR (DNASTAR inc.) and basespace (Illumina, inc., <https://illumina.com>). The majority of universities within the UK and many universities globally, still struggle with the costs and tools for genomic sequencing and analysis. This slows progress and future research. Recently progress has been made in addressing core issues such as computational infrastructure, via the set up of a Cloud Infrastructure for Microbial Bioinformatics, known as the 'CLIMB' project (<https://www.climb.ac.uk/>).

Currently, the alternative to costly graphical user interfaces is the command line terminal to access open source tools, such as; SPAdes, Velvet, Khmer, PRICE, mothur, MUMmer, and prokka (Bankevich, Nurk et al., 2012, Crusoe, Alameldin et al., 2015, Delcher, Kasif et al., 1999, Delcher, Phillippy et al., 2002, Ruby, Bellare et al., 2013, Schloss et al., 2009, Seemann, 2014, Zerbino & Birney, 2008), which are often obtuse and difficult for those with no experience in terminal use. Though there are several html alternatives such as MAKER, DNA Duster (<https://users.soe.ucsc.edu/~kent/dnaDust/dnadust.html>), KAAS, Phred, EGAssembler, Galaxy, etc (Cantarel, Korf et al., 2008, Ewing & Green, 1998, Ewing, Hillier et al., 1998, Giardine, Riemer et al., 2005, Masoudi-Nejad, Tonomura et al., 2006, Moriya, Itoh et al., 2007), html alternatives are significantly affected by the speed of the server and site traffic. Other effects on analysis speed include user upload and download speeds, as such html tools can often be much slower than running the same analysis in-house. html, like purchasable genomic analysis software, often has relatively strict pipelines, providing use of only some of the most common methods for basic analysis. The lack of diversity means no cross comparisons between various tools can be made to find the best analytical approach. This is not ideal as some tools are better for different sample/data types. Difficulties increase significantly in this form of analysis with large experiments, many often resorting to painstaking sample-by-sample analysis. Furthermore, publication often requires several analytical tools be investigated to prove the best tools have been used for your analysis, increasing

the strain for those unable to streamline the workload efficiently through scripts. This often means laborious and time consuming work, for this reason we built a free program to alleviate this strain when analysing genomic data. The basic functions and scripts within the program were built for, and used in, the analysis of chapter 7 when investigating the guts bacterial, fungal, and viral interactions in neonates.

6.1.3 Difficulties in genomic analysis

There is no standardised method for analysing genomic data, and each approach is often different depending on the quality of data and the sequencing method, for example massive parallel sequencing versus long read, SMRT sequencing. With the Illumina SBS method whole genomes need to be assembled, how you assemble a genome changes depending on the sample, and the type of assembler used can affect the quality of assembly, where some assemblers are better for certain sample types. Many errors can occur in genome assembly, particularly de-nova assembly, the type of problems that cause misassembly include; violation of mate-pair constraints, mis-classified singletons, and incorrect polymorphism classification. The annotation of an assembled genome designates the location of genes and other features. There are a few open source software tools available for annotation, see section 6.3.1. Automated annotation is the only practical method of annotation, particularly with the high throughput of next generation sequencing. Annotation is an error-prone process (Kyrpides & Ouzounis, 1999), and the automation of annotation is perhaps more prone to error (Richardson & Watson, 2013). Errors that are the product of automated annotation include; inconsistent annotation, same gene names and different product names, and many hypothetical/uncharacterised proteins likely artefacts of gene prediction process (Richardson & Watson, 2013). The problems in producing accurate genomic analysis highlighted thus far are just a sample of considerations when analysing a single type of sequencing data. The errors and problems that occur can frequently vary, emphasising the complexity and difficulties in comprehending and installing the tools required to accurately analyse data, and the problem of strict pipelines. These difficulties are stressed further when needing to streamline large sample numbers. GGOSS has a number of pre-set pipelines, but custom pipelines and settings can be made to adjust for data types and address individual error.

6.1.4 Programming languages

There are many coding languages used in building genomic tools, some of the core languages used are Python, Perl, C, C++, Java, and html, previously discussed in chapter 5. GGOSS (excluding external tools within GGOSS) uses a single coding language, the bourne shell (sh)/bourne again shell (bash). Bash/sh is the linux shell scripting and command language, is optimal for low level programming jobs and pipelining tools, as it is fast, low maintenance, and simpler for installation. Although bash/sh is capable of the same high level programming of python and perl etc, its more time consuming and difficult to achieve the same result. Languages like Python however are high maintenance due to it being a fast evolving language, and tools/programs that are not kept updated often become difficult to use. Open source software tools/programs are often not updated, requiring a user to maintain several python versions on a single operating system. Managing multiple python version on an operating system can be extremely challenging and troublesome.

Yet another dialog (YAD) (<https://github.com/EsMaSol/yad-dialog>) was used to wrap all the genomic tools and GGOSS program in a GUI. YAD, a fork of zenity, is ideal for the task, as it is designed to allow the building of simple user friendly interfaces that interact directly with the bash/sh shell and thus the linux terminal.

6.1.5 Aim

The aim of this chapter was to construct user friendly pipelines with analytical function to facilitate DNA sequencing data analysis through an easy to use interface that has no associated cost unlike similar pipelines like CLC genomics workbench (Qiagen). This process has been a natural expansion of research detailed in chapter 7, and future expansions are possible with the availability of new software. The aim of a user friendly design would enable researchers with little or no programming knowledge to carry out their analysis through an intuitive GUI. It should be noted that GGOSS (beta version) is currently only designed for illumina input files, this can and will be adjusted to accept additional file types prior to release.

6.2 Results

To our knowledge there has been no user-friendly open-access software package available for home install that implements many of the known OSS's into a high-throughput graphical user interface (GUI). Demonstrated here is a novel program with GUI interface; GUI for Genomic analysis using Open Source Software (GGOSS). GGOSS is designed to take raw DNA data of any number of samples through to finalised files and plots, for whole genome, bacterial 16S rRNA, fungal internal transcribed spacer (ITS), and viral community analysis. GGOSS provides several tools for every analytical step to give the user more flexibility in analysis rather than a strict pipeline, but with the ability to save default custom pipelines to further streamline their future experiments. Most importantly, GGOSS is designed to provide the identical functions and settings of the terminal tools, whilst adding additional analytical and transitional tools for greater interrogative power. The code written thus far for the GGOSS program can be found in the appendices section 10.9.

6.2.1 GGOSS installation

GGOSS is built for running on the linux operating system, as the program infrastructure and unique features run entirely in bourne/bourne again shell (sh/bash). This simplifies the installation, requiring just 2 commands: `‘chmod 755 ~/GGOSS_InstallFile/InstallGGOSS.sh’` (gives permissions to the script) and `‘~/GGOSS_InstallFile/InstallGGOSS.sh’` (runs the script). The only additional install for GGOSS (excluding external genomic OSS) requires only one other language, the statistical language ‘R’ which is installed by GGOSS for the user. Greater explanation of languages used and those types used by OSS can be found in chapter 5. Complication and length of installation is increased by the genomic open source software (OSS) installed by GGOSS for the user. The OSS have been constructed in languages such as Python, and often have other pre-installation requirements. Manual installation of OSS often involves much frustration, particularly with scientists unfamiliar with the terminal. GGOSS provides a simpler installation for these OSS, with 3 methods of installation; basic, careful, and complete. ‘Basic’ installs just the GGOSS program and none of the OSS tools within it, ‘careful’ installs only the OSS that are missing from the system, ‘complete’ installs all of the latest OSS used by GGOSS regardless of their presence on the system.

6.2.2 GUI map and file importation

The GGOSS GUI layout is built so that all categories and tools are directly accessible from the main menu. The map shows the simplicity of movement between menus, settings and running tools. As seen in Figure 6.3, the tabs for each category are organised left to right in a relatively natural order of analysis, to create a user-friendly flow. Further simplicity has been used for data input, to maintain a user friendly interaction with the program, where a ‘drag and drop’ method has been implemented.



Figure 6.3 ‘GGOSS’ GUI main menu (tab 1, 2, 3, and 4). The image demonstrates the categories separated by tabs between menu screens, A: Clean-up and QC, B: Assembly, alignment, mapping, and annotation, C: Post annotation analysis (Currently non functional), D. Community analysis

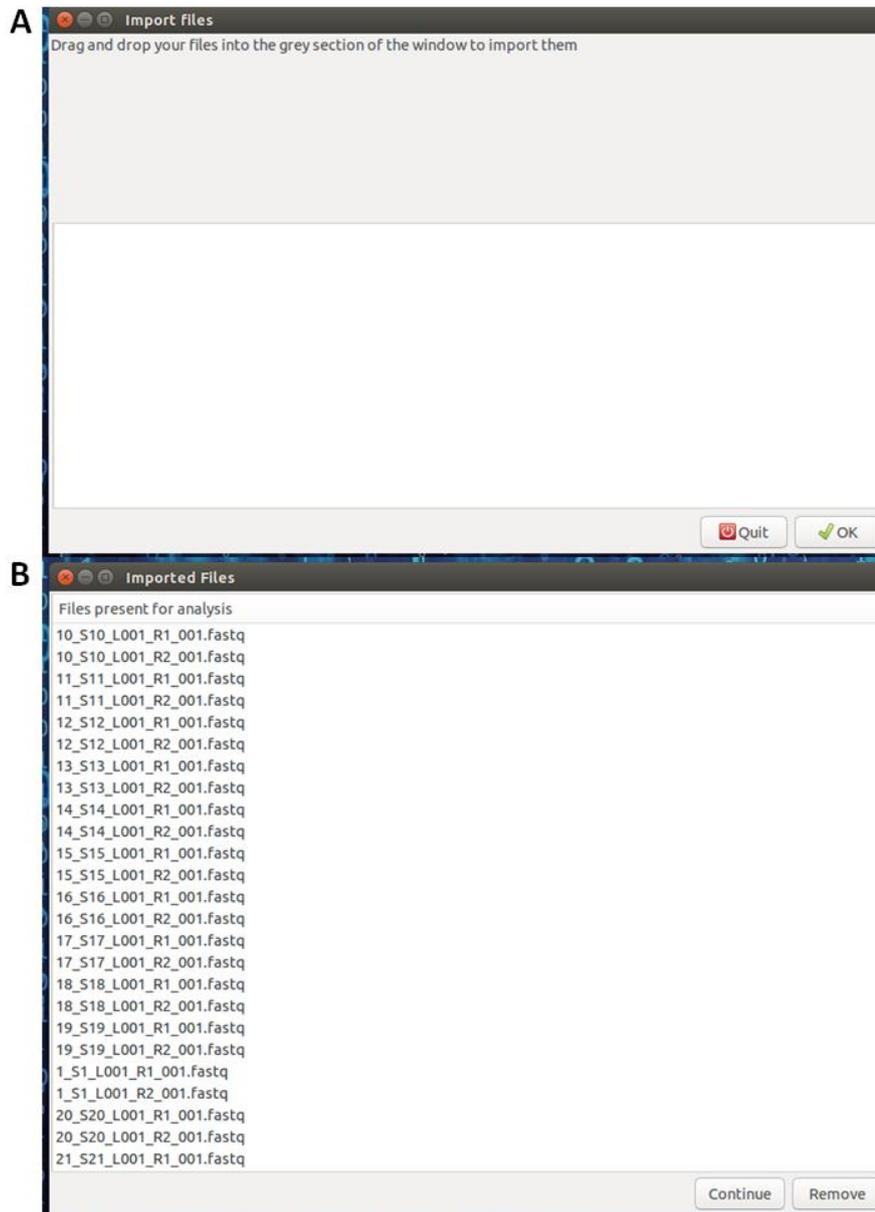


Figure 6.4 ‘GGOSS’ GUI ‘Drag and Drop’. The plot demonstrates the ‘drag and drop’ window for importing files into GGOSS (A), and the window confirming importation and removal option (B). In A the window explains the need to drag the files into the grey section of the window, the imported files will appear in the window below. The B window appears after confirming importation in window A.

6.2.3 Setting menus

To allow the same control and options available in OSS tools, a range of setting menus were built to accommodate all of the command flags of each tool. Within a given settings menu, options are more self-explanatory than the associated terminal command, however all flag names are also included to allow smooth transition for researchers that are use to the flags. Furthermore, settings selected are remembered by the program, and are only reset if the same settings menu is re-opened.

6.2.3.1 Clean-up and quality check tool settings

The Cutadapt tool is created by Marcel Martin, Department of Computer Science, TU Dortmund, Germany. The tool is designed to find and remove adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing files. The settings menu built for the Cutadapt tool can be seen in Figure 6.5.

Figure 6.7 demonstrates the GGOSS settings menu for the sickle tool. Sickle (<https://github.com/najoshi/sickle>) is used for improving the quality of sequence data by discarding reads that have deteriorating quality towards the 3'-end or 5'-end. Incorrect base calling in either region negatively impacts downstream bioinformatics analyses.

Figure 6.8 shows the GGOSS settings menu for Khmer (Crusoe et al., 2015), Khmer is designed for pre-processing short read Illumina data sets prior to de-novo sequence assembly. As per the settings menu the options within khmer are: digital normalisation, k-mer counting and read trimming, and partitioning reads into disconnected assembly graphs.

It's important to assess the quality of files before and after using genomic analysis tools. The quality of fastq files can be checked using FastQC (<https://github.com/bsib/ncs/fastqc>), no setting changes are required for FastQC and the GGOSS menu for it can be seen in Figure 6.9. QUAST is another quality checking tool incorporated into GGOSS, QUAST provides information on the quality of assembly, the GGOSS settings menu for QUAST can be observed in Figure 6.10.

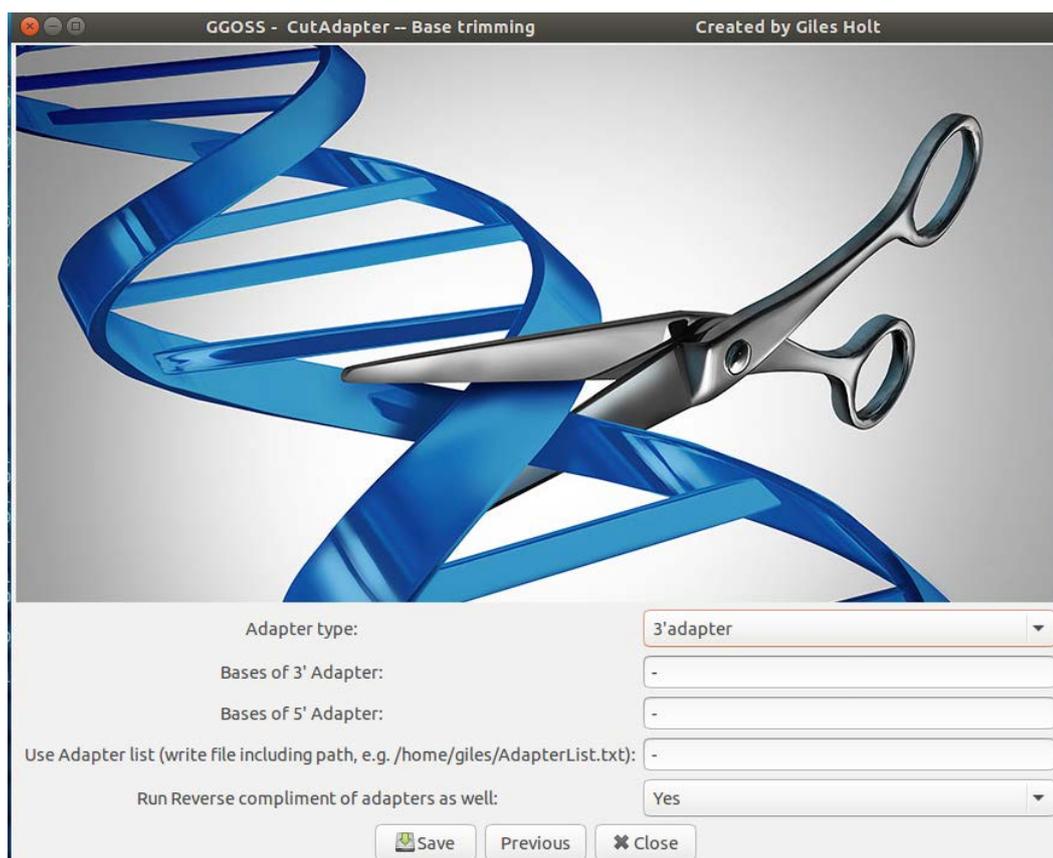


Figure 6.5 ‘GGOSS’ GUI Cutadapt settings menu. The image demonstrates the Cutadapt settings window, it provides full control and function of the Cutadapt OSS. Settings provide the choice of adapter types, input for adapter bases, adapter list option, and additional check and removal of reverse compliments. Settings can be edited and saved upon selection of the save button.

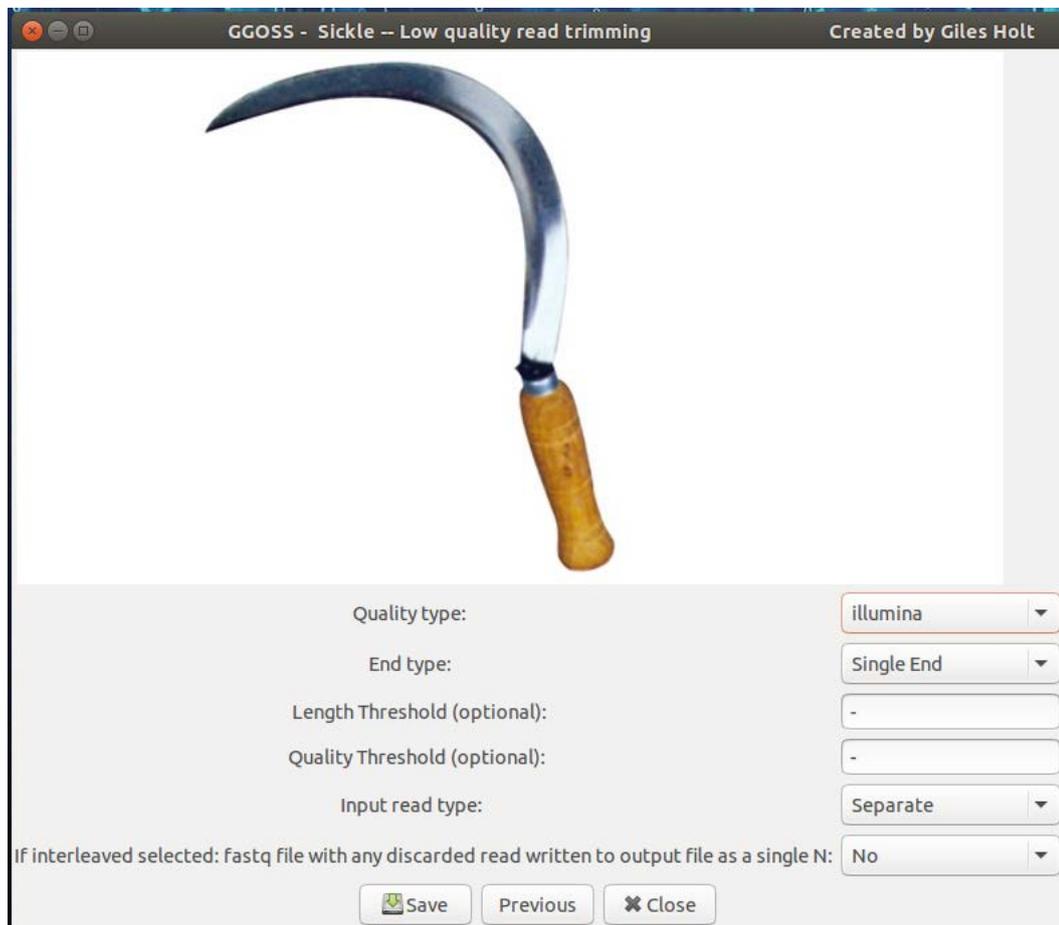


Figure 6.6 ‘GGOSS’ GUI Sickle settings menu. The image demonstrates the sickle settings window, it provides full control and function of the Sickle OSS. Settings provide the Quality type, end type, Length threshold, quality threshold, input read type, and interleaved options. Settings can be edited and saved upon selection of the save button.

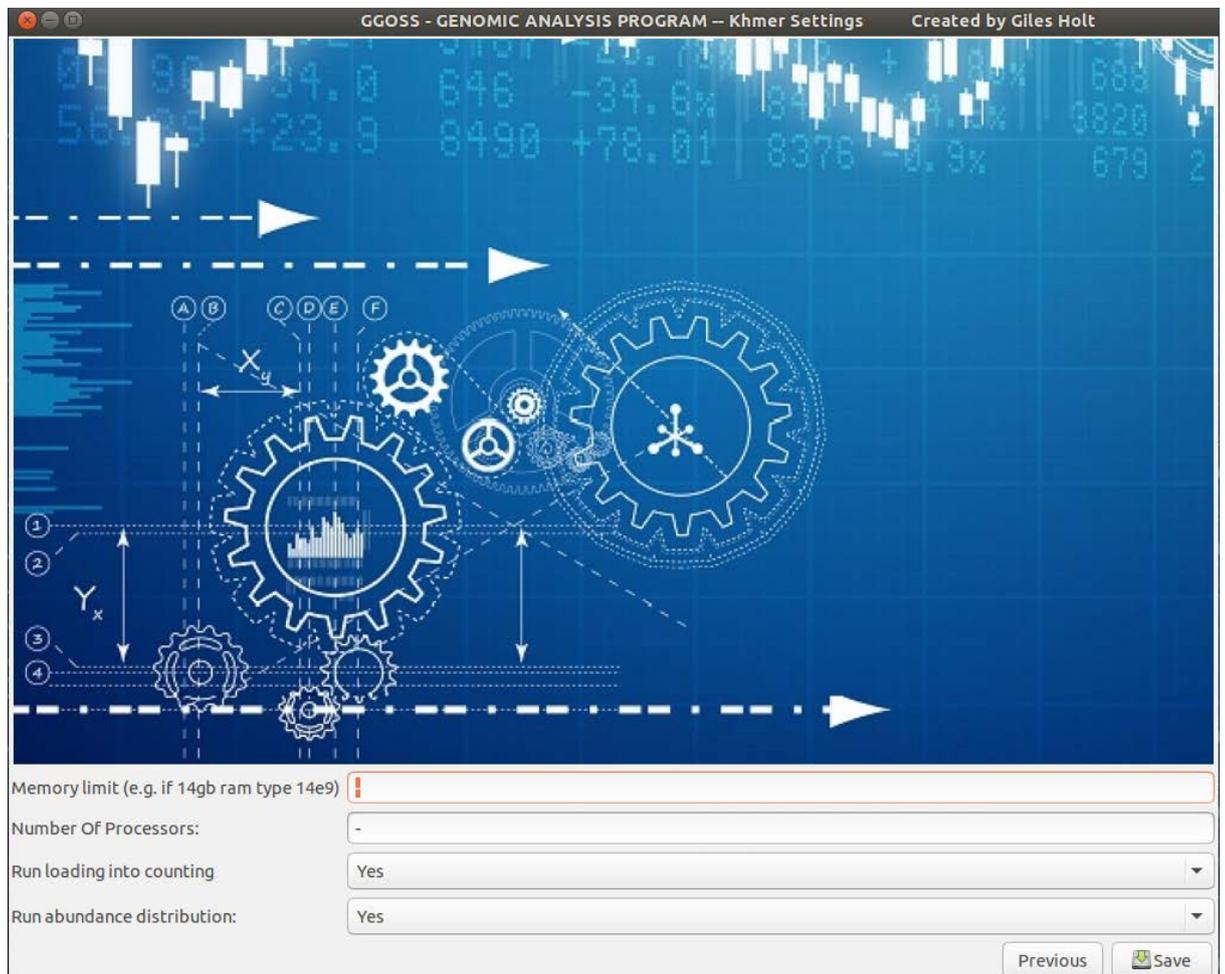


Figure 6.7 ‘GGOSS’ GUI Khmer settings menu. The image demonstrates the window for the khmer settings currently built in, it provides some control and function of the khmer OSS. Settings provide the ram limit, number processors, script run types (‘loading into counting’ and ‘abundance distribution’). Settings can be edited and saved using the save button.



Figure 6.8 ‘GGOSS’ GUI FastQC menu. The image demonstrates the FastQC menu window. Settings are not required and sequence files are simply selected and run through FastQC.

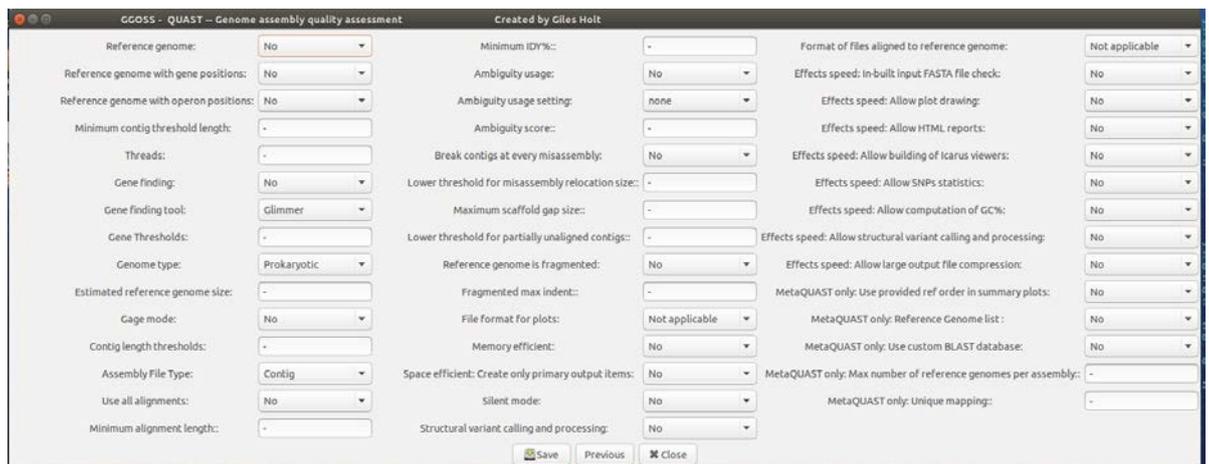


Figure 6.9 ‘GGOSS’ GUI QUAST settings. The image demonstrates the QUAST settings window, it provides full control and function of the QUAST OSS, where settings can be edited and saved upon selection of the save button.

6.2.3.2 Tool settings for sequence assembly, mapping, alignment, and translation

This beta GGOSS currently has two tools capable of genome assembly; SPAdes and PRICETI. The Velvet OSS assembler is also partially included into the program, though only for the function of sequence file shuffling, it's not currently a functional assembler in GGOSS.

Figure 6.10 demonstrates the GGOSS settings menu for St. Petersburg genome assembler (SPAdes) (Bankevich et al., 2012), SPAdes is an assembly toolkit containing various assembly pipelines. Figure 6.11 shows the GGOSS menu for the various GGOSS setting menus for Paired-Read Iterative Contig Extension (PRICE), several menus are created due to the complexity of the tool. Unlike other assemblers, PRICE is designed with metagenomic subcomponents of interest in mind, as it iteratively increases the size of existing contigs, individual reads from a subset of the paired-read dataset, or non-paired reads from sequencing technologies that provide non-paired data. PRICE can also be used for normal assembly. Figure 6.12-Figure 6.15 show the settings of each settings group seen in Figure 6.11, the setting groups being: Input/output, parameter, filter read, and filter contig.

BLAST finds regions of similarity between biological sequences, comparing nucleotide or protein sequences to sequence databases and calculating the statistical significance. This is a core genomic tool, and commonly used in genetic analysis. The settings for this tool can be seen in Figure 6.15.

Mapping sequence reads to a reference genome has its advantages and disadvantages to de-novo assembly, though as it requires a reference genome it's not always possible. Disadvantages of mapping sequences include: bias towards a reference genome, not normally as good for large/medium scale differences. In many methods reads that do not map are not used in the final sequence, and new completely different sequences are lost. Advantages to the sequence mapping include: Less contigs, and single nucleotide polymorphisms (SNP's) and structural variants (SV's) are more easily positioned and compared among groups. Because of the advantages and disadvantages, sometimes a combination of de-novo and sequence mapping is used. Sequence mapping tools and tools incorporating sequence mapping that are built into GGOSS are; Ragout,

BWA, PRICE, and MUMmer. The setting menus for these can be seen in Figure 6.12 and Figure 6.17-6.19.

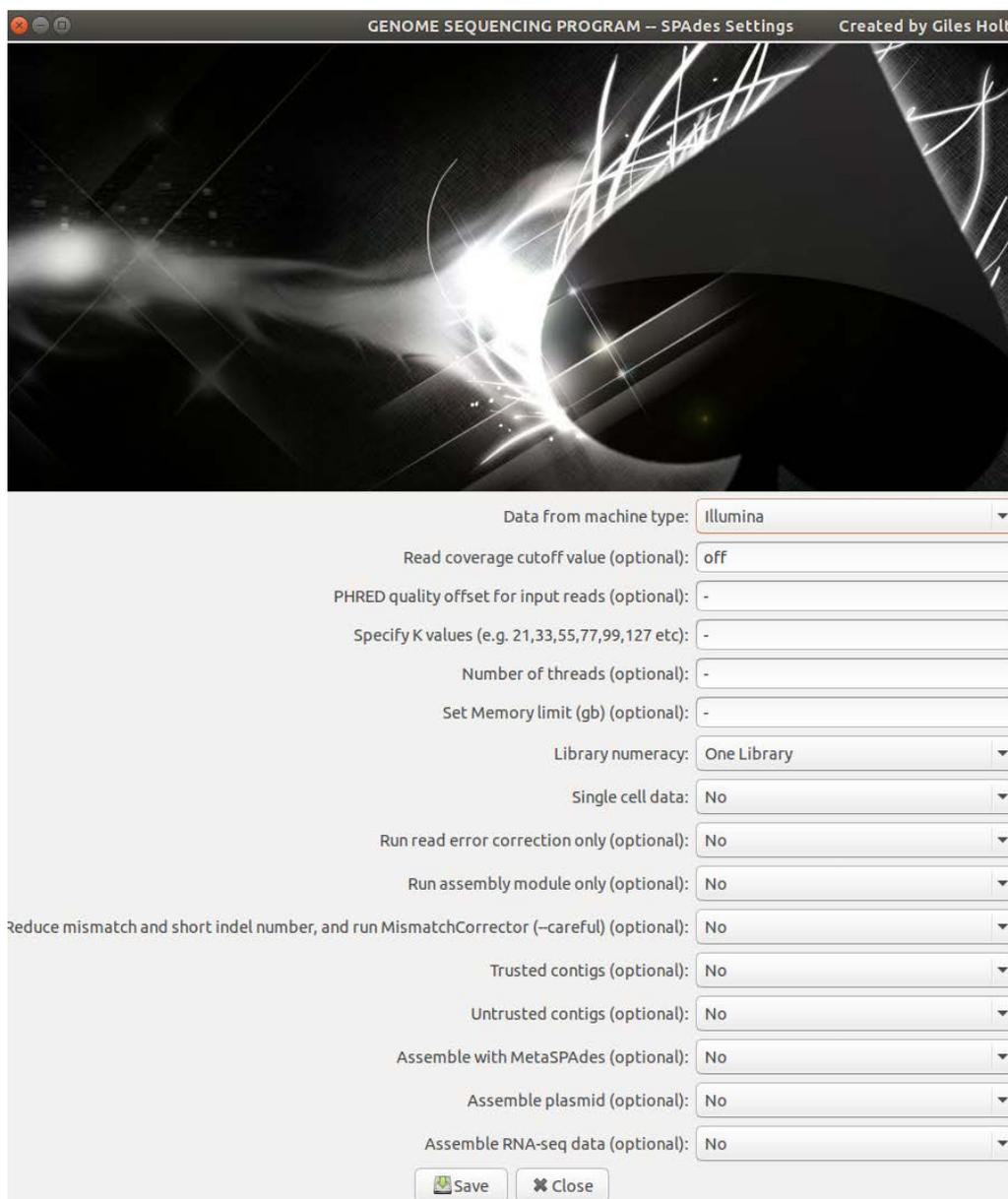


Figure 6.10 ‘GGOSS’ GUI SPAdes settings menu. The image demonstrates the SPAdes settings window, it provides full control and function of the SPAdes OSS, where settings can be edited and saved.



Figure 6.11 ‘GGOSS’ GUI PRICE settings menu. The image demonstrates the PRICE settings window, it contains the selection of setting categories, ensuring full control and function of the PRICE OSS. Setting categories shown left to right are Input/Output settings, parameter settings, filter read settings, filter contig settings.

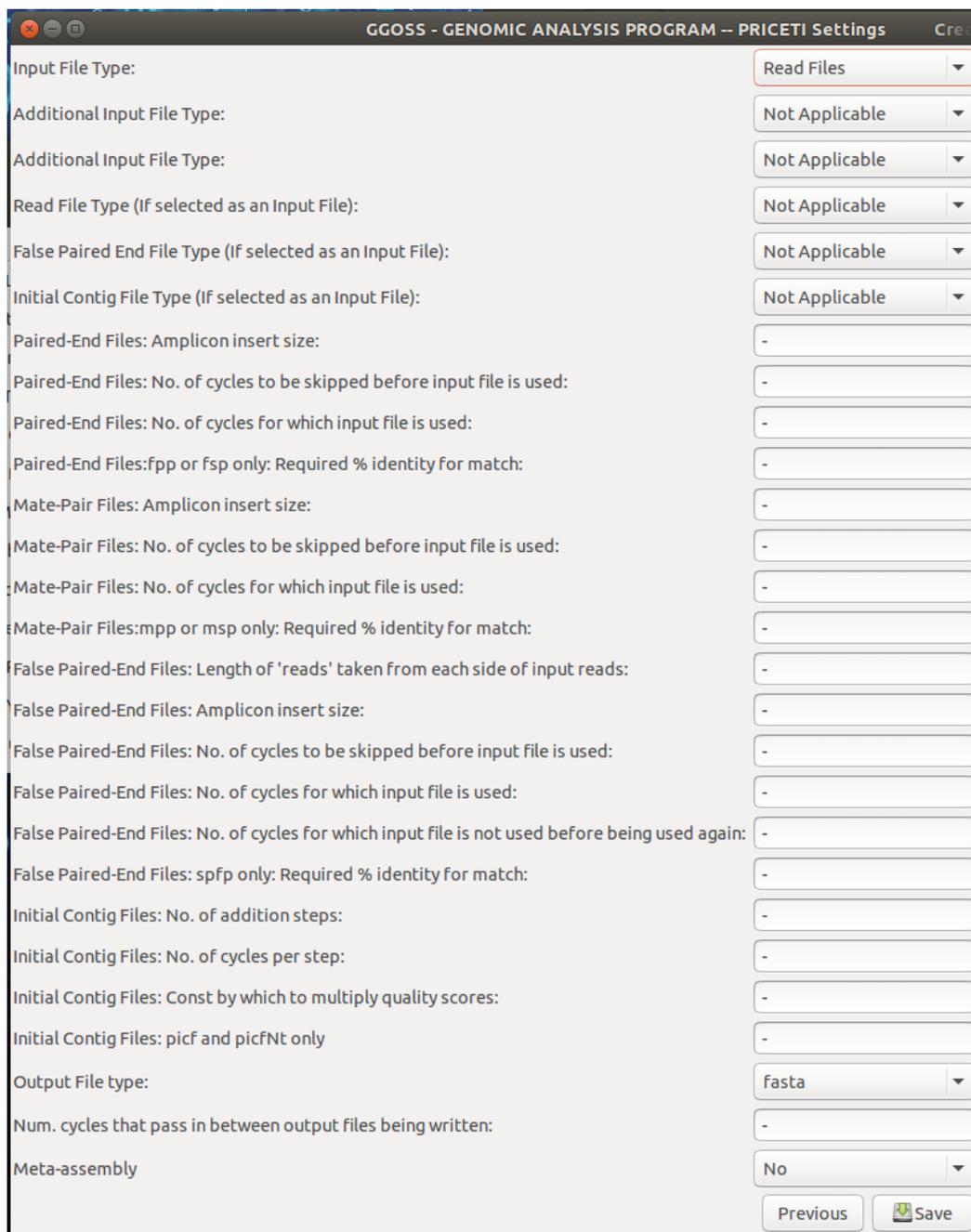


Figure 6.12 ‘GGOSS’ GUI PRICE input/output settings. The image demonstrates the PRICE input/output settings window, where settings can be edited and saved.

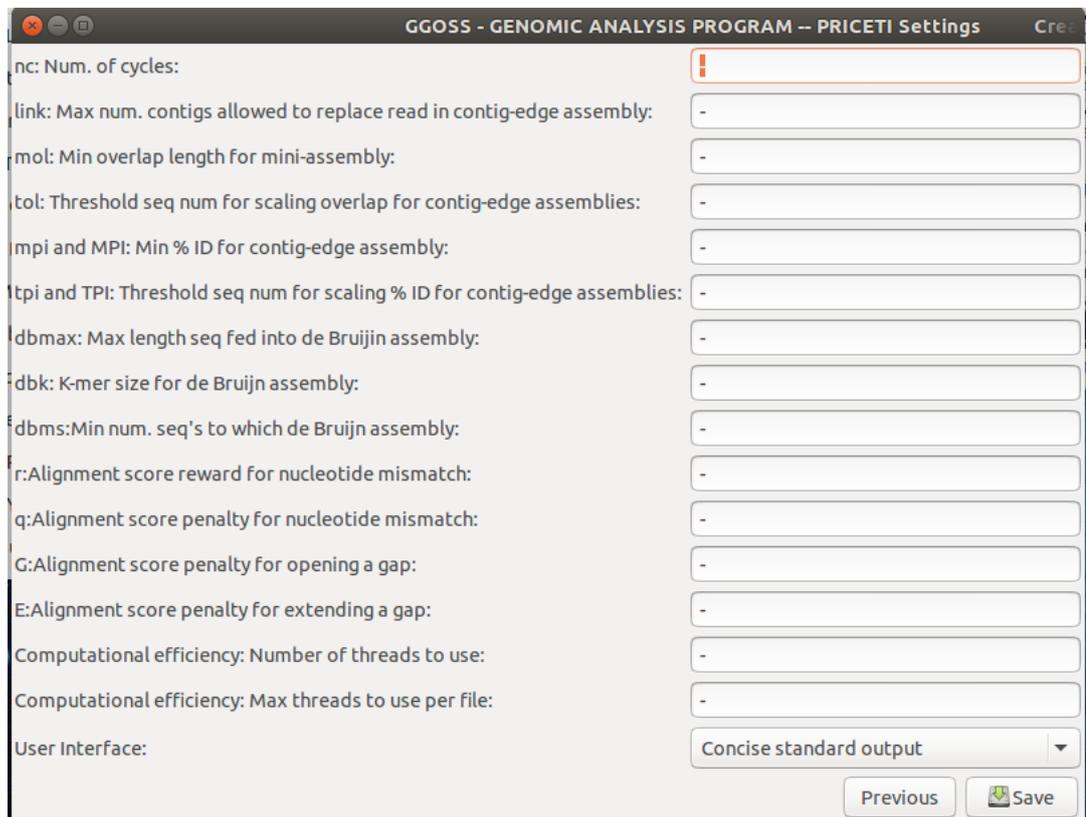


Figure 6.13 ‘GGOSS’ GUI PRICE parameter settings. The image demonstrates the PRICE parameter settings window, where settings can be edited and saved.

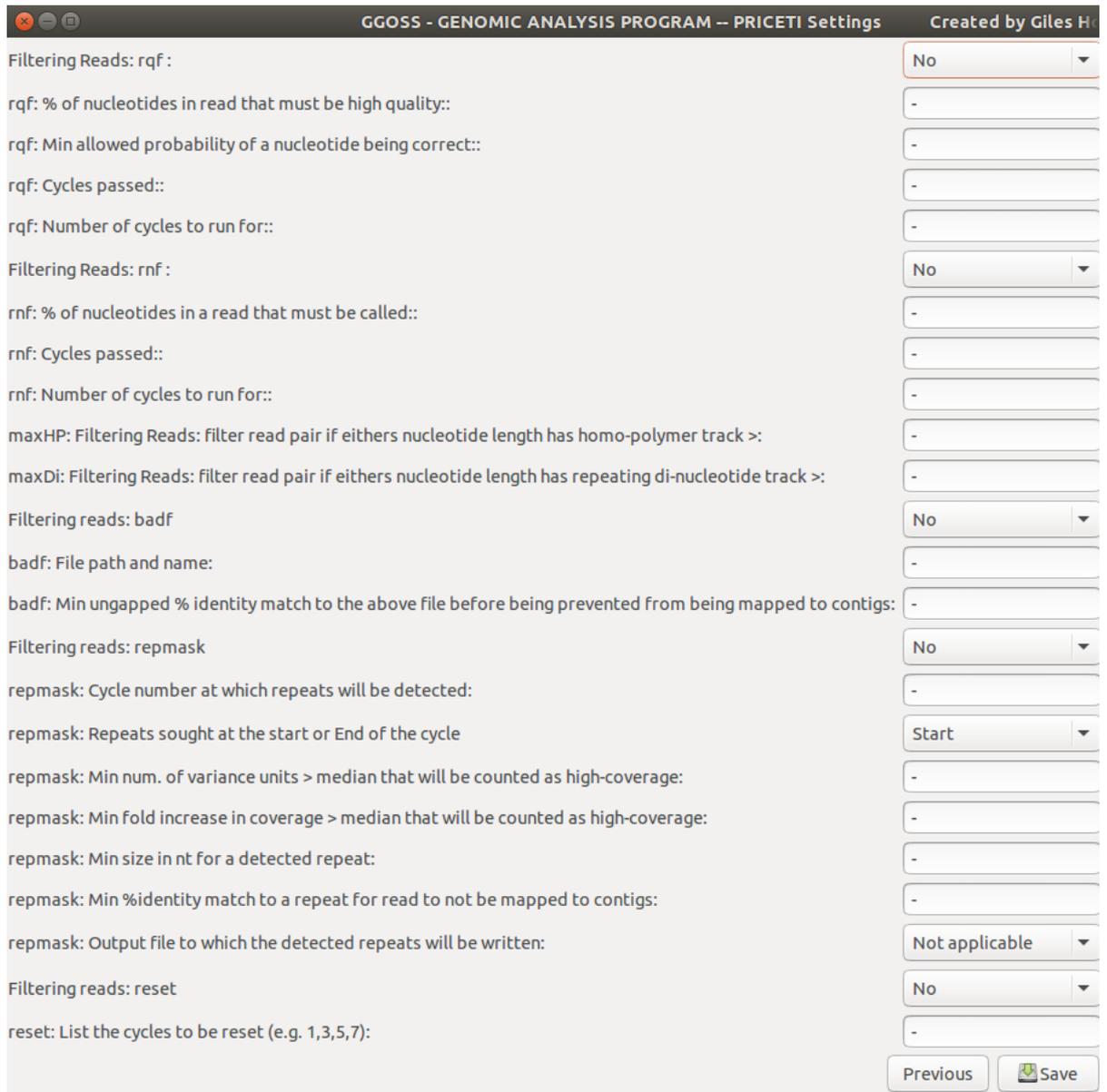


Figure 6.14 ‘GGOSS’ GUI PRICE filter read settings. The image demonstrates the PRICE filter read settings window, where settings can be edited and saved.

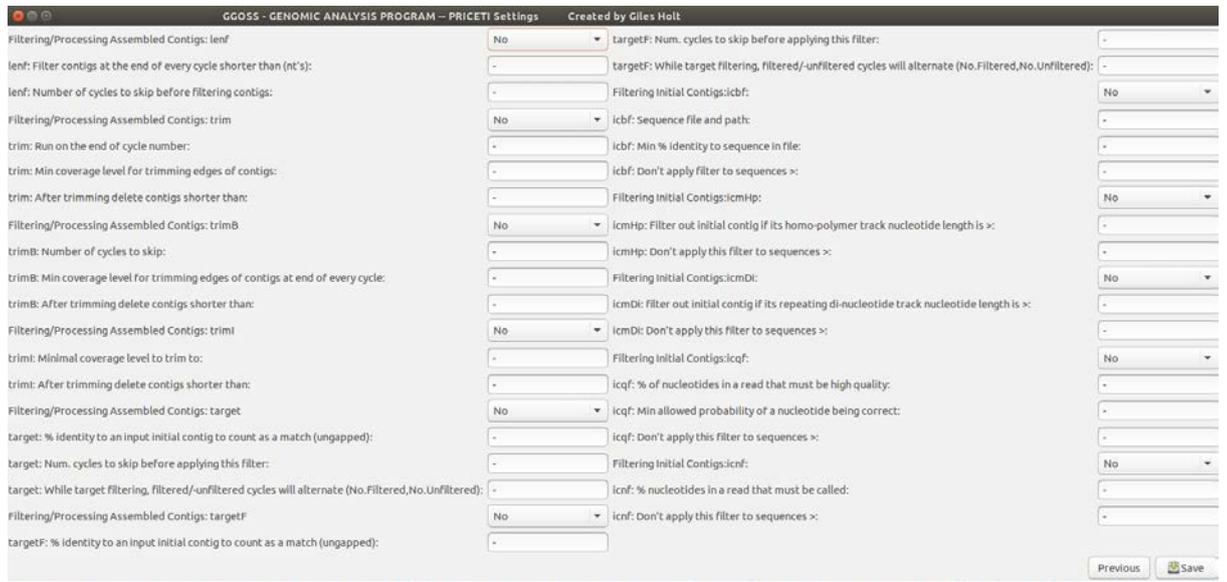


Figure 6.15 ‘GGOSS’ GUI PRICE filter contig settings. The image demonstrates the PRICE filter contig settings window, where settings can be edited and saved.

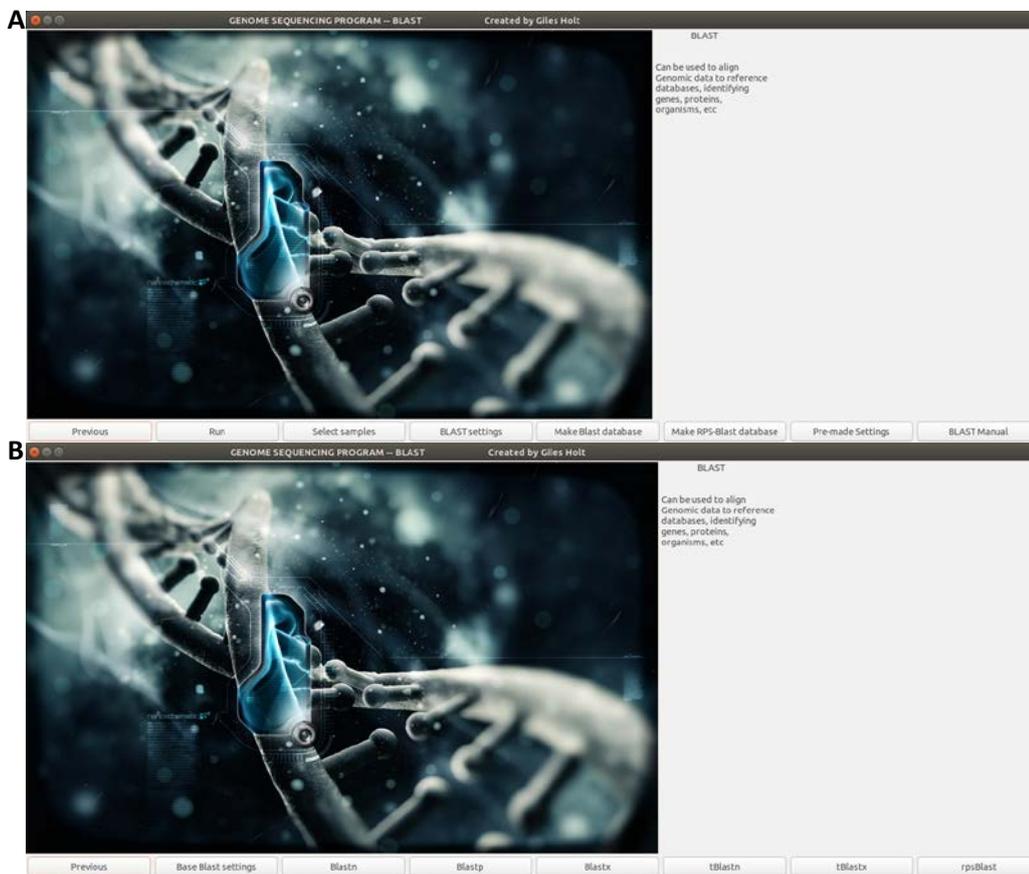


Figure 6.16 ‘GGOSS’ GUI BLAST main menu and settings. The image demonstrates the BLAST main menu window (A) and main settings window (B). The image in section B contains the categories of blast run types, each button links to its own settings selection window where settings can be edited and saved.

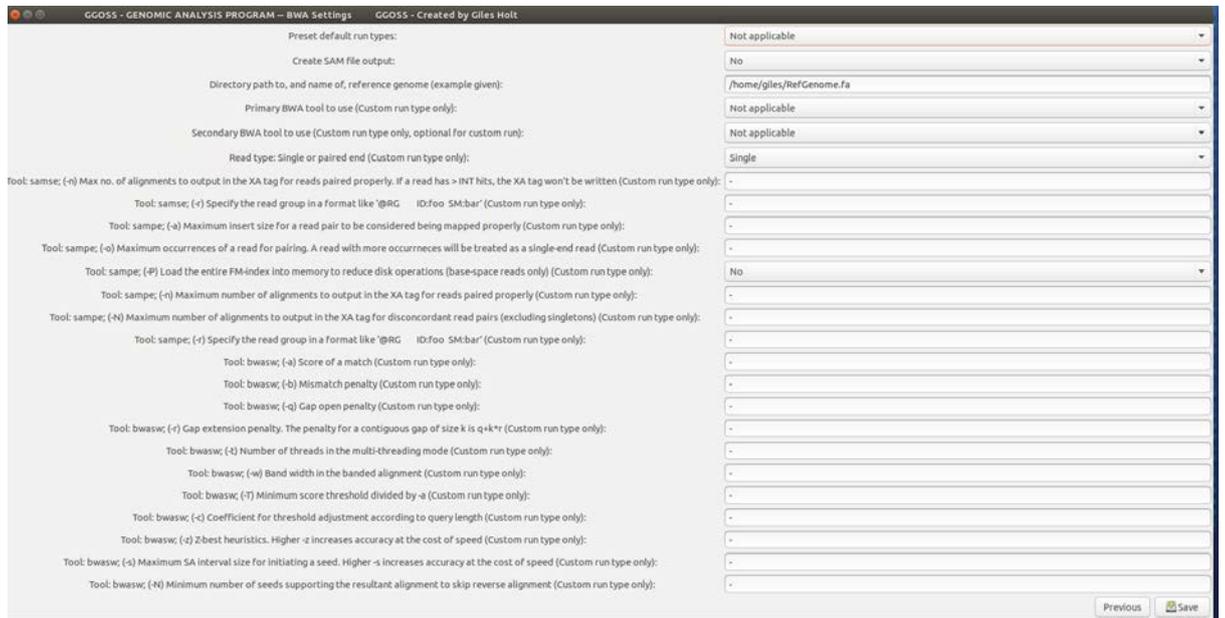


Figure 6.17 ‘GGOSS’ GUI BWA settings. The image demonstrates the window for the BWA settings currently built in, it provides full control and function of the BWA OSS. Settings can be edited and saved using the save button.

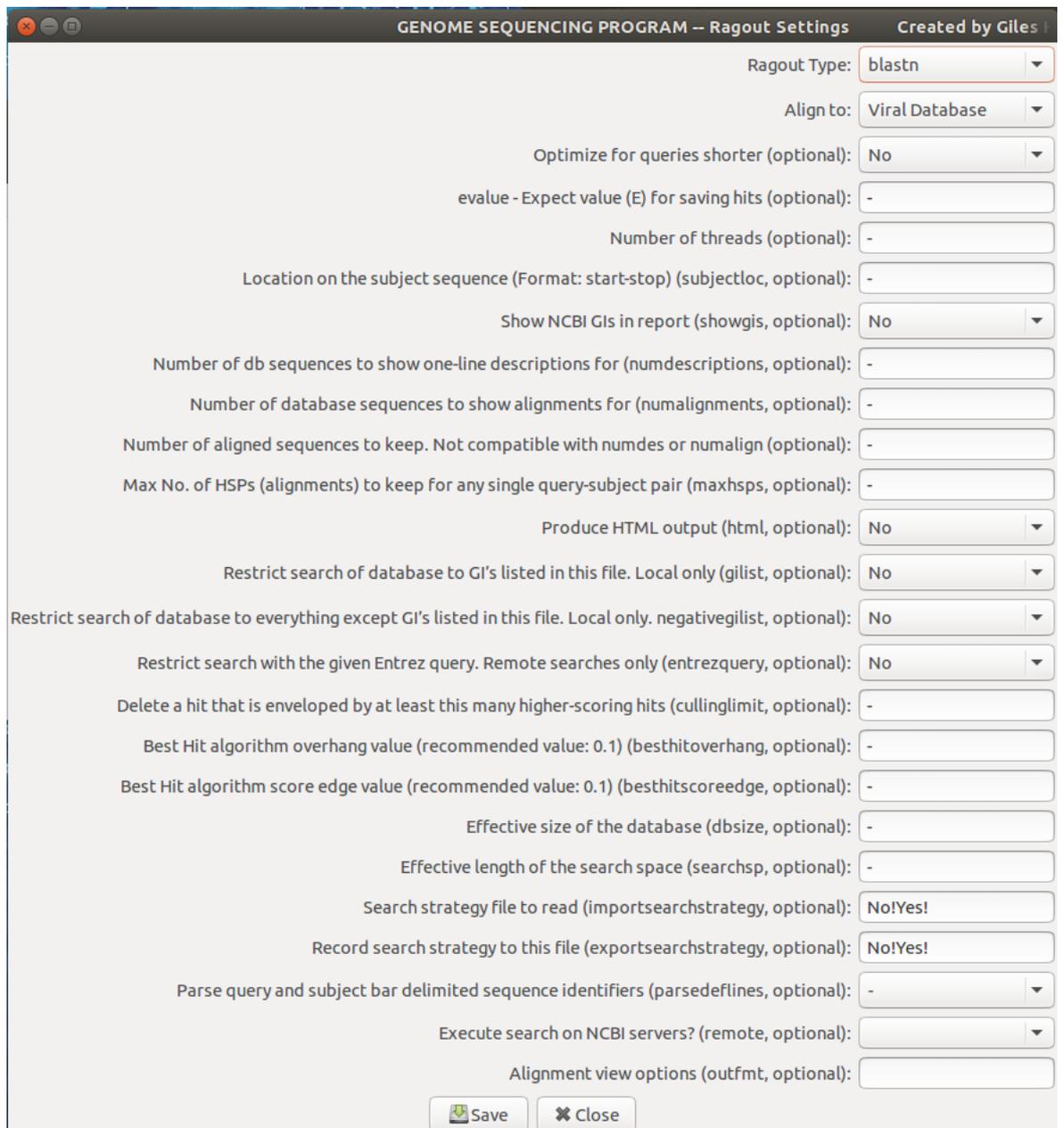


Figure 6.18 ‘GGOSS’ GUI Ragout settings. The image demonstrates the window for the Ragout settings currently built in, it provides full control and function of the Ragout OSS. Settings can be edited and saved using the save button.

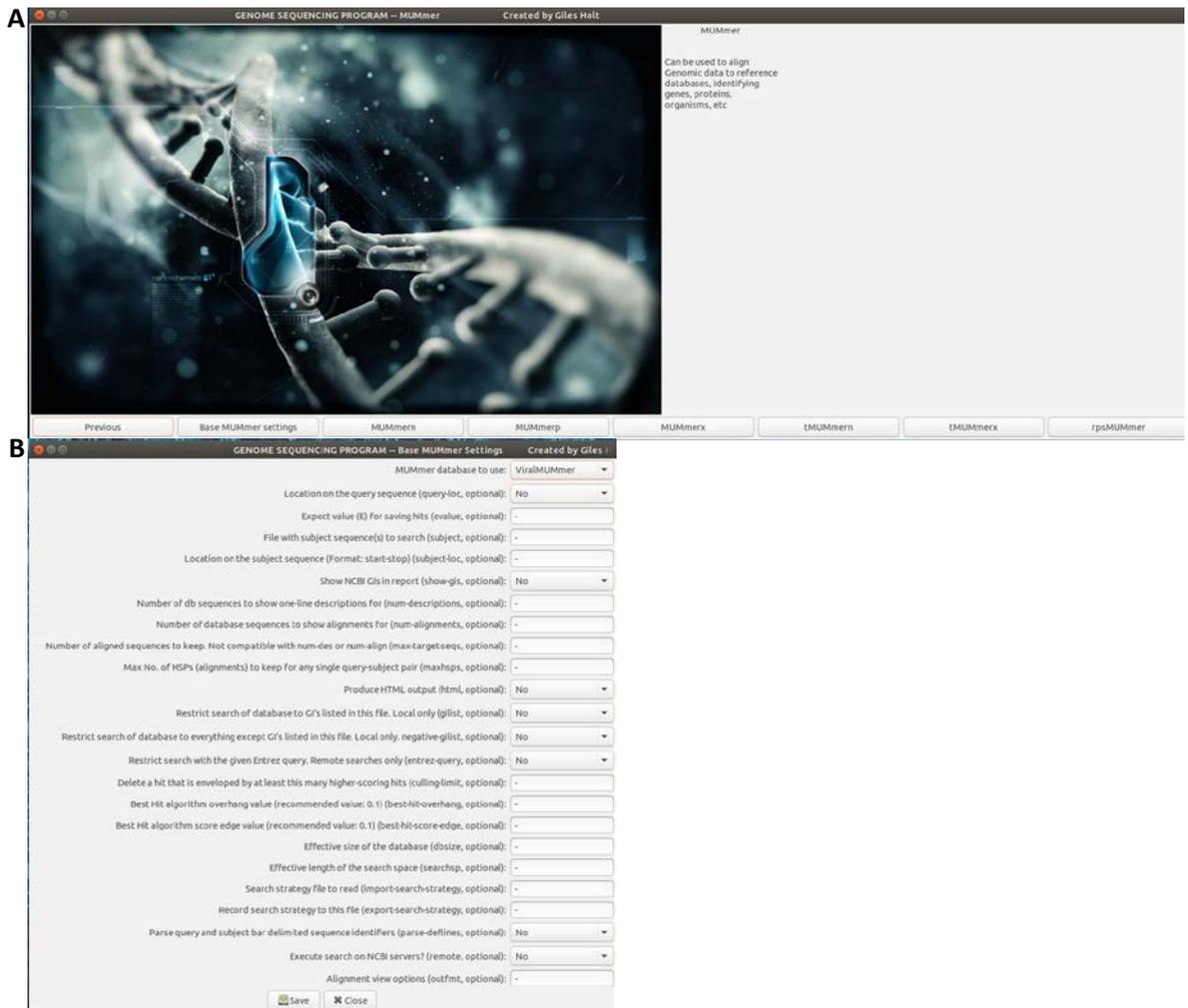


Figure 6.19 ‘GGOSS’ GUI MUMmer settings. The image demonstrates the window for the MUMmer setting menus (A) and the functioning setting menu ‘Base MUMmer settings’ (B). It provides control and function of the base functions of MUMmer OSS. Settings can be edited and saved using the save button.

6.2.3.3 Amplicon sequencing and community analysis pipelines

Community analysis of a given environments micro-organisms has become relatively cheap and common practice in science, with many associations found in community structure linked to disease and disease progression. As community structure can play an essential role in our understanding and identification of disease and its progression, it's important to identify as much of the community as possible. Here we show the menu settings of 3 OSS tools built into GGOSS; mothur (Figure 6.20 and Figure 6.21), PIPITS (Figure 6.22) and MetaPhlAn (Figure 6.23). The OSS 'mothur' is primarily used for 16S rRNA gene amplicon sequencing analysis, though it can also be used for any given community target based sequencing, for example, fungal community analysis, this method was used in chapter 7. As demonstrated in Figure 6.20 every mothur command is selectable, additional mothur parameters built into GGOSS can be seen in Figure 6.21, which include the reference database type. PIPITS is designed for fungal analysis. Its pipeline is relatively inflexible as seen in the GGOSS settings menu (Figure 6.22), PIPITS is also strict to ITS regions 1 and 2. MetaPhlAn is used for community analysis of whole genome sequencing data, which is particularly useful for viral community analysis.

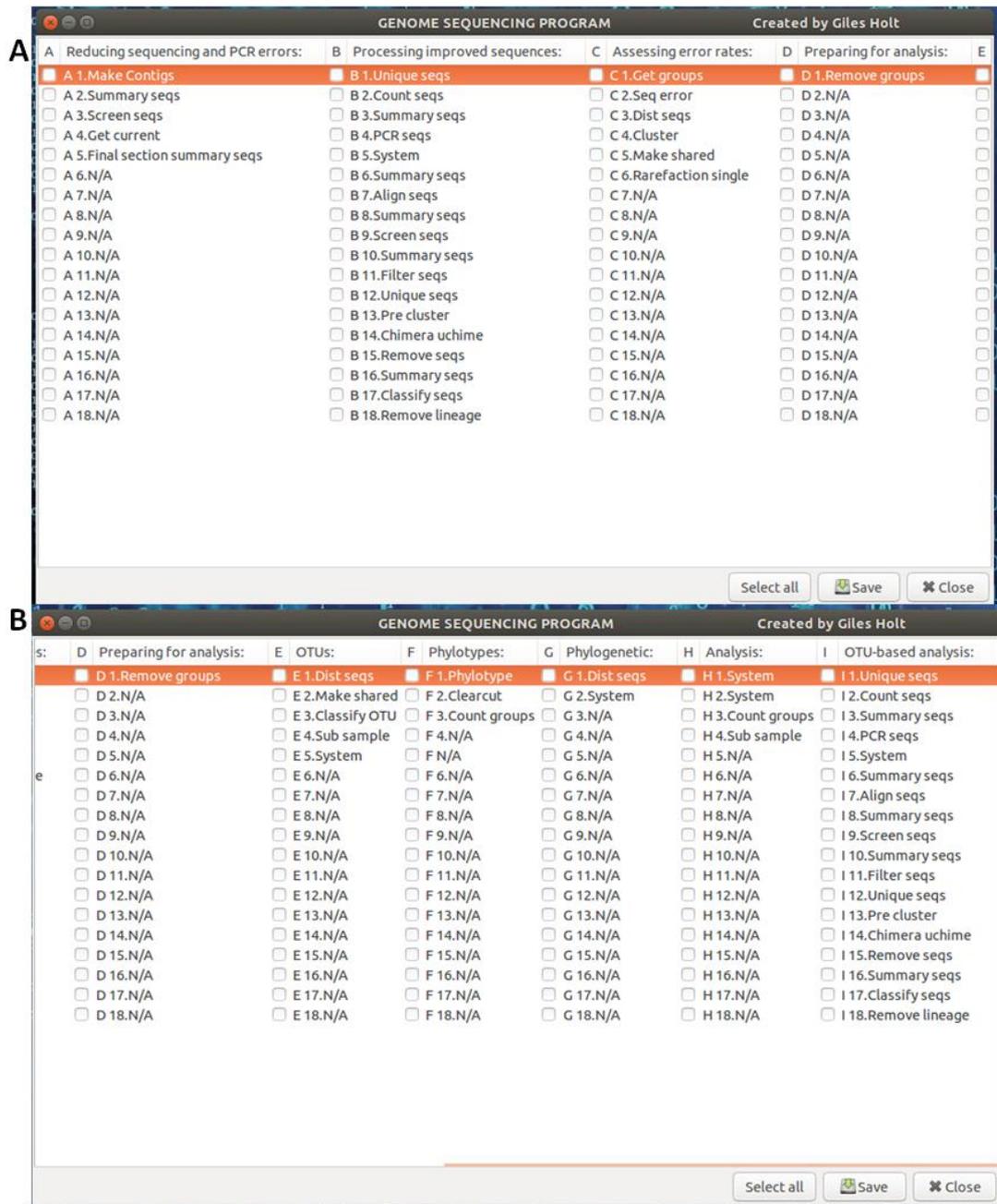


Figure 6.20 ‘GGOSS’ GUI mothur step selection. The image demonstrates the scrollable window for the Mothur step selection settings, it provides full control and function of the Mothur OSS. Image A and B represent the full settings upon scrolling left through them. Settings can be edited and saved using the save button.



Figure 6.21 ‘GGOSS’ GUI mothur run settings. The image demonstrates the window for the Mothur run settings, it provides full control and function of the Mothur OSS basic run settings. Settings provide the option of the number of processors, the amount of ram to commit, the reference database type, and pathway to customised reference database option. Settings can be edited and saved using the save button.



Figure 6.22 ‘GGOS’ GUI PIPITS settings. The image demonstrates the window for the PIPITS settings, it provides full control and function of the PIPITS OSS. Settings provide the option of the read type, the ITS region, the amount of ram to commit, choice for construction of output table for FUNGuild analysis, and output file name. Settings can be edited and saved using the save button.

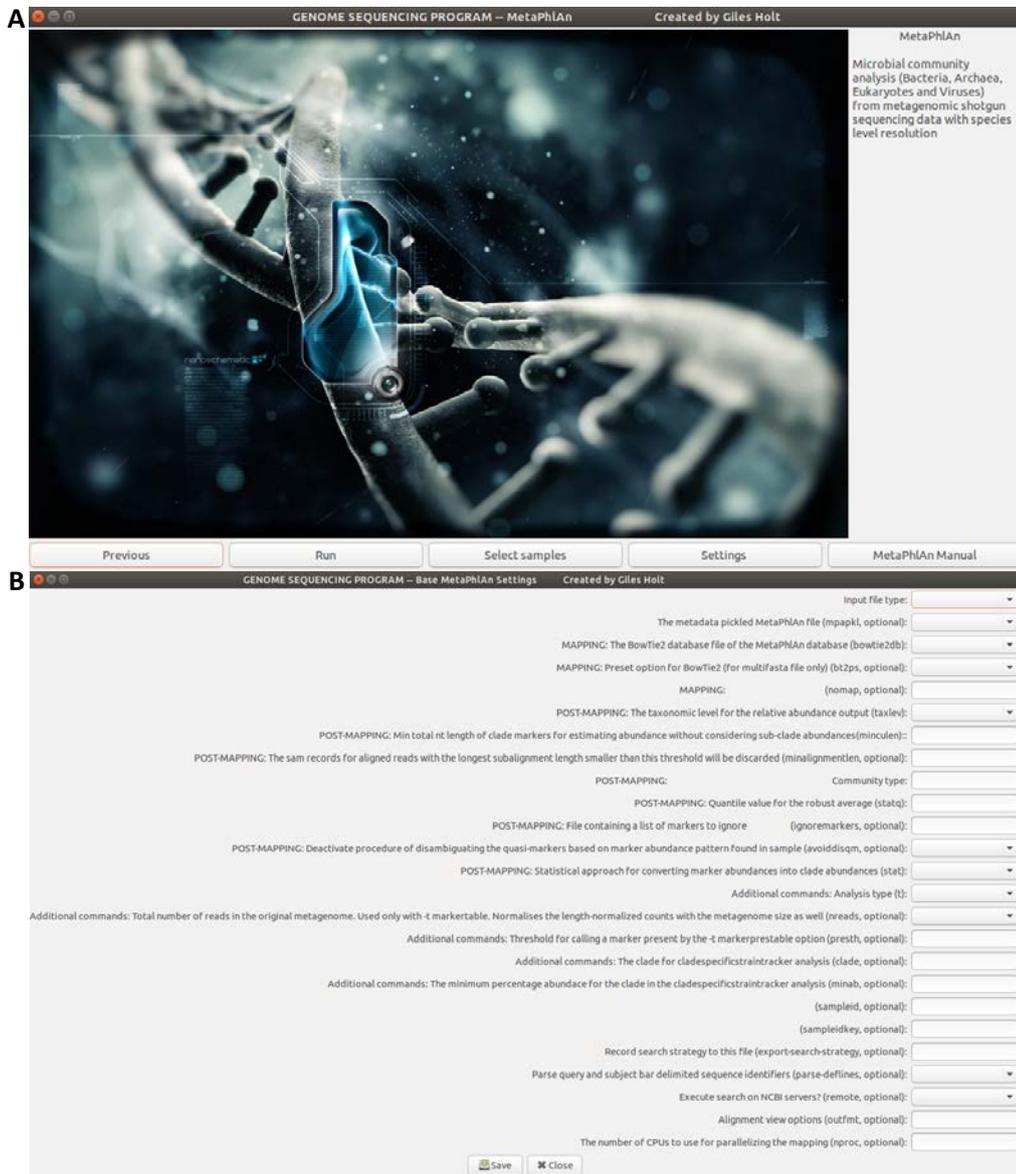


Figure 6.23 ‘GGOSS’ GUI MetaPhlAn settings. The image demonstrates the window for the MetaPhlAn main page (A), and the base MetaPhlAn settings (B), it provides partial control and function of the MetaPhlAn OSS. Settings can be edited and saved using the save button.

6.2.3.4 Annotation pipelines

With next generation sequencing platforms low costs and large outputs, automated annotation is not just essential but the only functional way to carry out annotation. There are a number of OSS annotation tools, those included into the GGOSS program thus far are Artemis (Figure 6.24) and Prokka (Figure 6.26). Artemis is already a fully functional GUI, which GGOSS opens for the user. The Prokka setting options are quite limited, it is currently built into GGOSS with a strict pipeline, where the majority of the settings are associated to unique GGOSS additions (see 6.2.4).

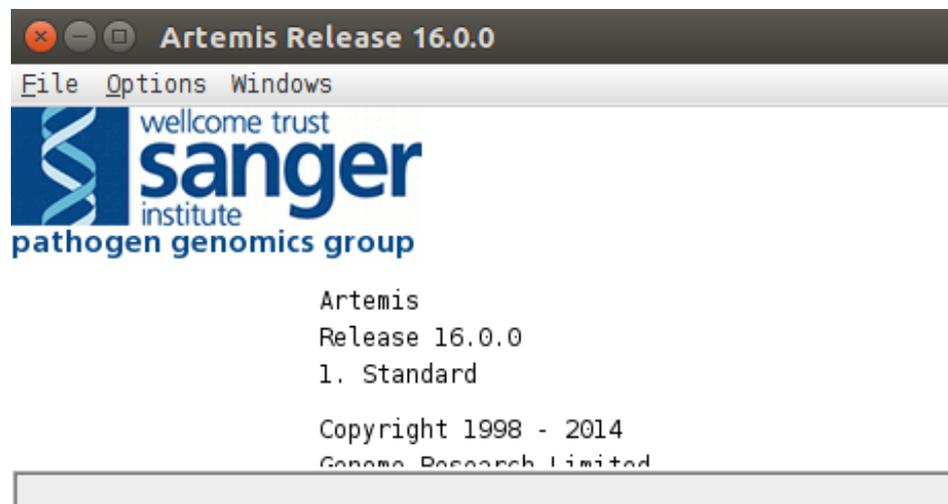


Figure 6.24 Artemis GUI. The image demonstrates the Artemis main menu window accessible through GGOSS.

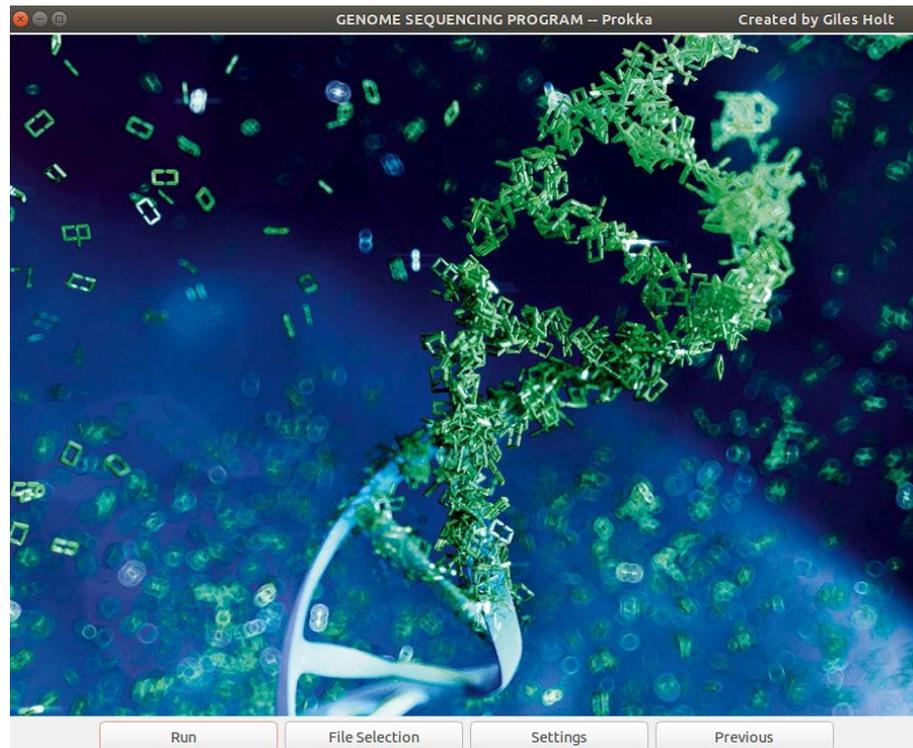


Figure 6.25 ‘GGOSS’ GUI Prokka main menu. The image demonstrates the Prokka main menu window in GGOSS.

6.2.4 Unique features to GGOSS

GGOSS is designed to streamline analysis using available terminal controlled OSS. It does this via its ability to feed countless samples through any OSS at the click of a button in its user-friendly GUI. To ensure OSS within GGOSS are as up-to-date as the user desires, the setting window of all the OSS installed can be opened and manually adjusted by giving the new name and location of the version acquired (Figure 6.29).

GGOSS has several unique additions not offered by incorporated OSS. Additions thus far are made to cutadapt, QUAST, Prokka, and mothur. GGOSS unique features for cutadapt allow 3 additional functions to the tool. These are adapter lists, reverse complement run, and default Nextera clean-ups. These are visible in Figure 6.5. Adapter list function allows any given number of adapters to be searched for and removed from a sample or mass of samples. The GGOSS unique feature for QUAST is the tabulation of data, as every sample input creates a separate file of data.

This function combines them into a single table to allow easy comparisons. The GGOSS unique feature for prokka is to allow easier transaction between assembly and annotation (Figure 6.27). This feature allows each contig within a file to be: separately annotated, annotated as a whole, annotate top so many contigs, annotate contigs of a certain size, or annotate contigs of a certain coverage. There are several GGOSS unique features for mothur (Figure 6.26), these are: stability file creation, taxonomic rank selection, community trimming methods, and plot creation.

As GGOSS is designed for simplifying mass sample numbers, a specific tool was built for GGOSS to help deal with adjusting file names on a large scale. This tool is particularly useful within GGOSS, as GGOSS edits the name of a file to include what tools it's gone through. Reduction in file name size, works by setting a delimiter and choosing how many/which columns to keep.

Though running samples on mass through tools is very useful, linking tools further improves streamline analysis. This is why GGOSS also has a 'stack tools' option, allowing any number of tools to run from one to the next, passing input and output without user interference required. It should be noted that the stacked tools option is only partially complete, as such it is only available for some of the core OSS tools. GGOSS allows smoother transitions between tools without requiring user based interaction, examples include converting fastq files to fasta and creating single files from paired files (when necessary).

Some genomic OSS tools create logfiles, those that form log files tend to create a separate logfile for each sample run through. GGOSS creates its own logfile specifically designed for mass analysis, as it places all run information into a single file, containing not just a given genomic OSS tools logs, but also the list of sample names run, the settings that were selected, and the start/completion times.

Additional tools (written entirely in BASH) have also been created here specifically for GGOSS, these include; a DNA/RNA converter, a conserved sequence finder, and a viral taxonomy finder. The DNA/RNA converter tool allows the conversion of any sequence, or file containing sequence data, to any of the following formats; reverse sequence, reverse compliment sequence, compliment sequence, RNA equivalent sequence, and DNA equivalent sequence. The conserved

sequence finding tool takes any number of fasta files and finds any conserved sequences between them. This tool uses a strict sliding window approach, as it does not allow for single nucleotide polymorphisms. The viral taxonomy finding tool takes any viral abundance table and, using the ICTV webpage database, edits the names of the viruses to an associated taxonomic rank of your choosing, allowing for quick and easy pooling of viral data.

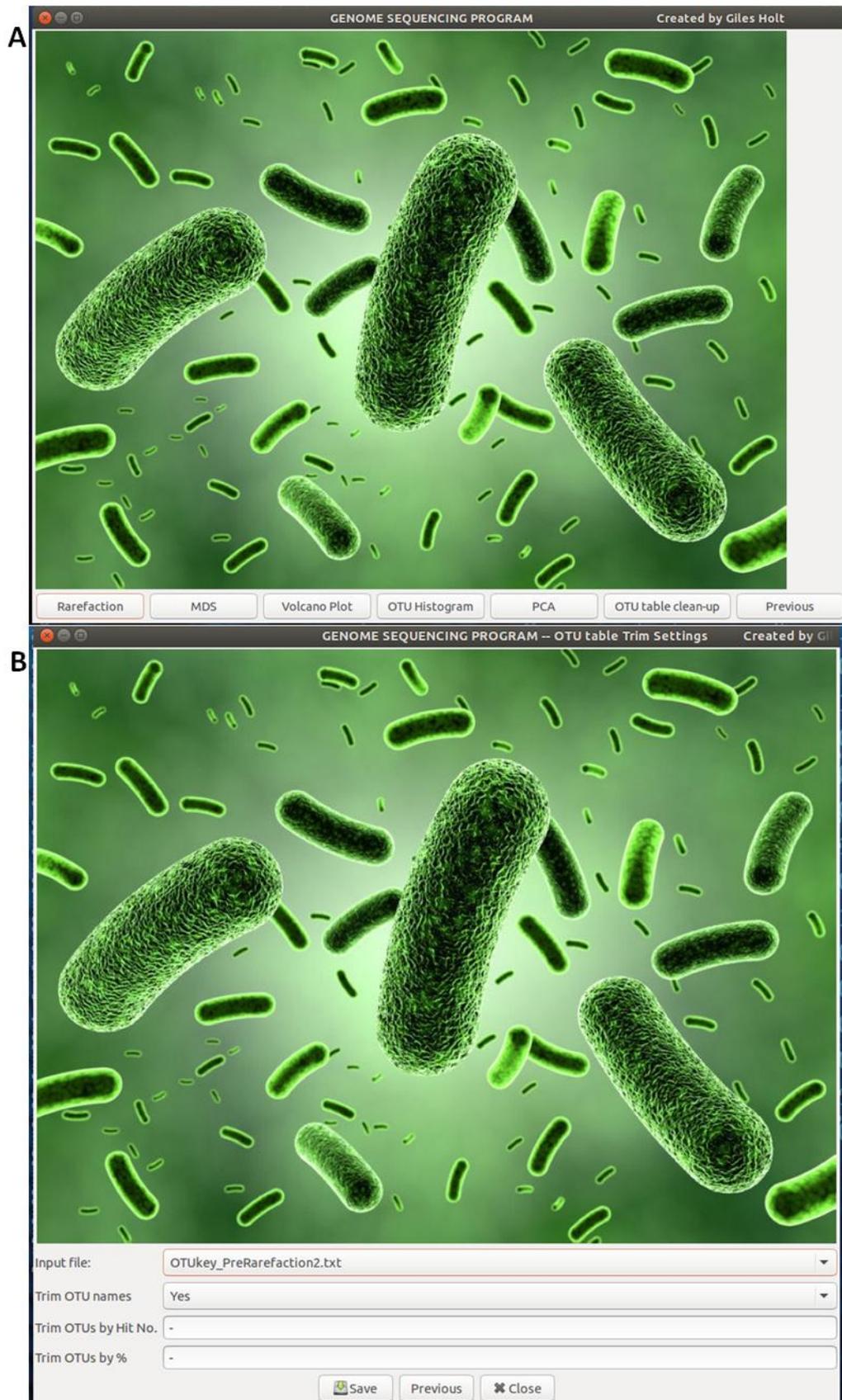


Figure 6.26 ‘GGOSS’ GUI community distribution settings. The image demonstrates the table manipulation functions and plotting options for community counts (A). Currently each tab has limited settings, and simply runs the selected function on the table provided. The OTU table clean-up function is shown in section B, as it has editable settings. Settings can be edited and saved using the save button.

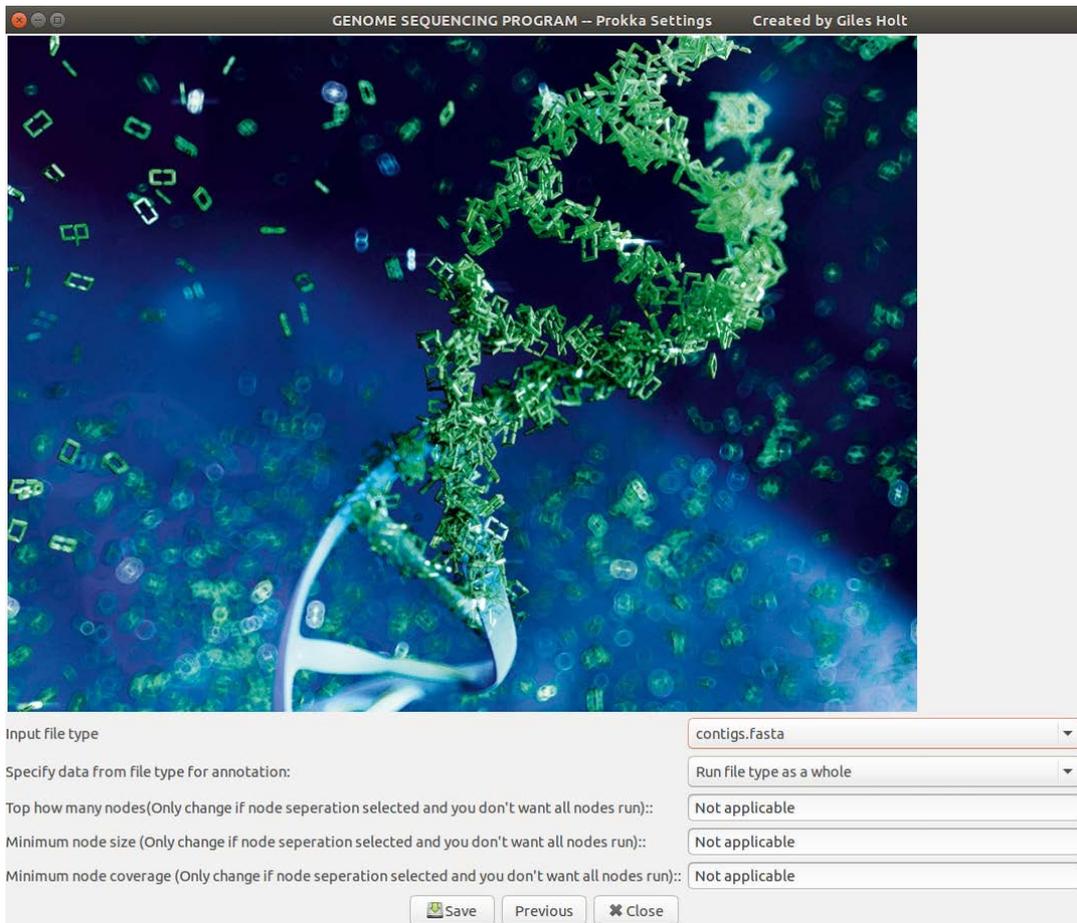


Figure 6.27 ‘GGOSS’ GUI Prokka settings menu. The image demonstrates the window for the PROKKA settings, it provides control and function of the PROKKA OSS, as well as providing additional GGOSS settings for PROKKA. Settings provide the input file type, full data annotation or selective, top no. of nodes, max node size, and min node size. Settings can be edited and saved using the save button.

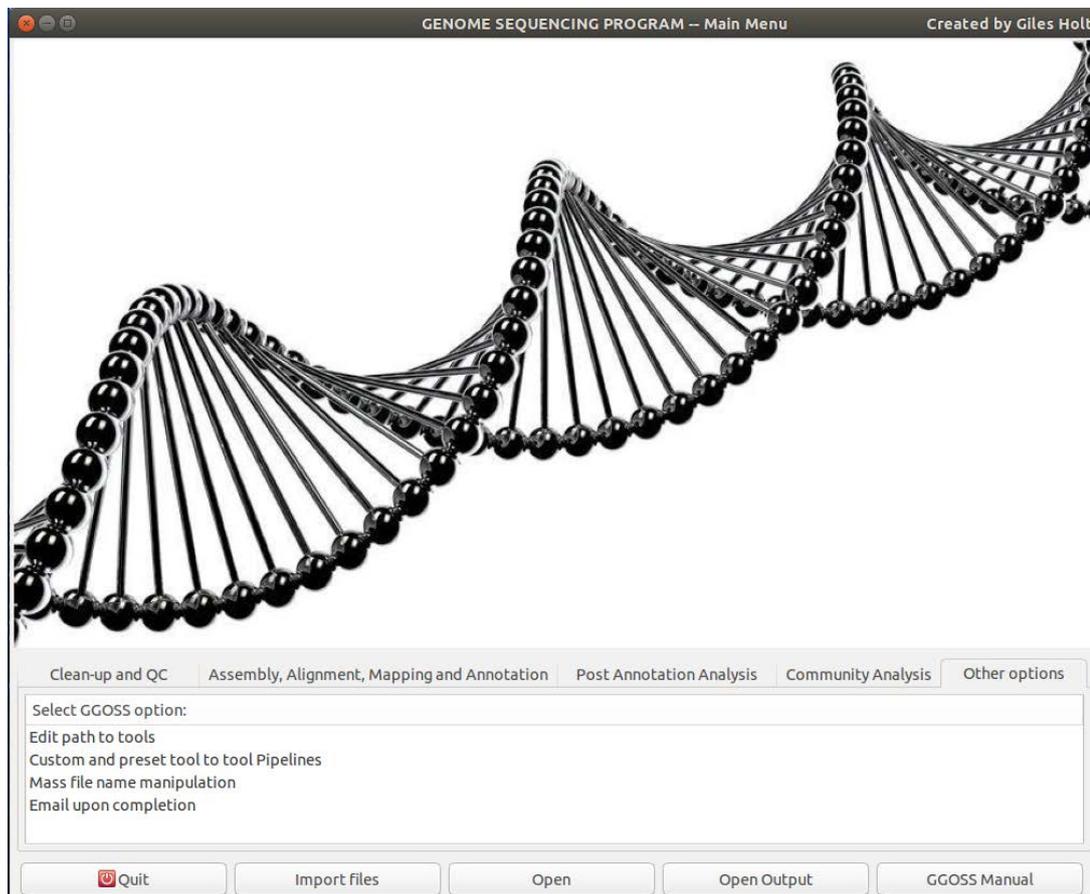


Figure 6.28 ‘GGOSS’ GUI main menu (tab 5). The image demonstrates the menu window for the GGOSS settings. Settings provide control over tool paths and names, custom and preset pipelines, and file name manipulation. The email option is currently not functional.

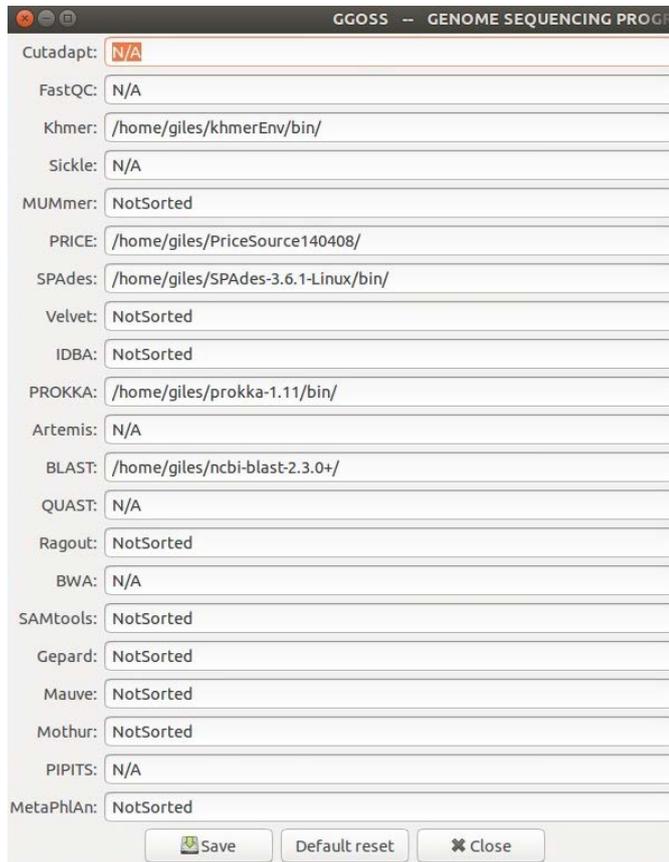


Figure 6.29 ‘GGOSS’ GUI tool path settings. The image demonstrates the settings window for the GGOSS paths to installed tools. The paths in each field are used by the GGOSS scripts to direct to a given tool, some scripts aren’t yet linked to this, and would need to be manually edited for the time being. Settings can be edited and saved using the save button.

6.3 Discussion

GGOSS is the first installable genomic analysis GUI program that includes a large variation of OSS tools covering the majority of genomic analysis techniques, whilst providing novel tools, mass sample analysis, custom setting defaults, and custom tool pipelining. Its user friendly interface, streamlining methods, progression bars and estimated completion times, allow for easy genomic analysis. Furthermore GGOSS installation can carry out installation of all OSS integrated into the program, which is otherwise another difficult and time consuming task. It should be noted that recently pre-installation of OSS has become a feature in projects such as CLIMB and Biolinux. The program was originally for analysis carried out in chapter 7, so many of the tools included have been used in GGOSS to carry out the analysis required in this thesis.

6.3.1 OSS genomic tools

The tools incorporated into GGOSS are highlighted in bold in Table 6.1. Alternative tools are shown adjacent within the table. The amalgamation of tools incorporated into GGOSS thus far, have been specifically selected to provide several general methods/types of genomic analysis. More OSS tools for any given step are intended to be included in future versions. The types of tools included allow for quality assessment and resolution, whole genome assembly and annotation, as well as viral, bacterial, fungal, archaeal, and eukaryotic community analysis. These can be summarised into whole genome analysis and community analysis.

Table 6.1 Range of some of the more prominent open source software tools available for genomic analysis, where tools incorporated into GGOSS are in highlighted in bold. Grouped categories, are generalised as some tools cross categories.

Tool category	Tool name	Function	Reference
Trimming tools	Cutadapt	Searches for the adapter in all reads and removes it when it finds it	(Martin, 2011)
	Trimmomatic	Performs a variety of useful trimming tasks for illumina paired-end and single ended data	(Bolger, Lohse et al., 2014)
	Skewer	Adapter trimming designed for (but not exclusive to) processing illumina paired-end sequences	(Jiang, Lei et al., 2014)
	PEAT	Adapter trimming designed for (but not exclusive to) processing paired-end sequences	(Li, Weng et al., 2015b)
	AlienTrimmer	Removal of alien sequences (adapters, primers) from raw reads, allows detecting and removing of multiple alien sequences in both ends of sequence reads.	(Criscuolo & Brisse, 2013)
	Sickle	A windowed adaptive trimming tool for FASTQ files using quality	https://github.com/najoshi/sickle
K-mer (searching, counting, indexing, filtering)	Khmer	a probabilistic k-mer counting data structure, a compressible De Bruijn graph representation, De Bruijn graph partitioning, and digital normalization.	(Crusoe et al., 2015)
	Tallymer	Counting, indexing, and searching k-mers	(Kurtz, Narechania et al., 2008)
	Jellyfish	k-mer counting in DNA sequence data	(Marcais & Kingsford, 2011)
	BFCOUNTER	Generating and counting k-mers in DNA sequence data	(Melsted & Pritchard, 2011)
	DSK	Generating and counting k-mers in DNA sequence data	(Rizk, Lavenier et al., 2013)
	KMC (1,2, and 3)	k-mer counting and manipulation of k-mer datasets	(Deorowicz, Debudaj-Grabysz et al., 2013, Deorowicz, Kokot et al., 2015, Kokot, Dlugosz et al., 2017)
	Turtle	Identifying and counting frequent k-mers	(Roy, Bhattacharya et al., 2014)
Sequence similarity alignment and taxonomic classifiers	BLAST	BLAST finds regions of similarity between biological sequences	(Ye, McGinnis et al., 2006)
	Kraken	Assigning taxonomic labels to metagenomic DNA sequences	(Wood & Salzberg, 2014)
	CLARK	classification of metagenomic and genomic sequences using discriminative k-mers	(Ounit, Wanamaker et al., 2015)
Assemblers	SOAP2	Short oligonucleotide alignment and assembly program	(Li, Yu et al., 2009)
	AbySS	A de novo, parallel processor, paired-end sequence assembler that is designed for short reads	(Simpson, Wong et al., 2009)
	BASE	de novo assembler for large genomes using long NGS reads	(Liu, Liu et al., 2016)
	SPAdes	Assembly toolkit containing various assembly pipelines	(Bankevich et al., 2012)
	Velvet	A de novo genomic assembler specially designed for short read sequencing technologies	(Zerbino & Birney, 2008)
	IDBA	A de nova assembler designed for short read sequencing technologies	(Peng, Leung et al., 2010)

Sequence data: quality assessment	HTQC	A quality control toolkit for Illumina sequencing data	(Yang, Liu et al., 2013)
	FastQC	A Quality Control application for FastQ files	https://github.com/csf-ngs/fastqc
	PIQA	Pipeline for Illumina G1 genome analyzer data quality assessment	(Martinez-Alcantara, Ballesteros et al., 2009)
Assembly: quality assessment and correction	QUAST	Quality assessment tool for genome assembly	(Gurevich, Saveliev et al., 2013)
	MetaQUAST	Quality assessment tool for metagenomic assembly	(Mikheenko, Saveliev et al., 2016)
	GAGE	Quality assessment tool for genome assembly	(Salzberg, Phillippy et al., 2012)
	REAPR	Quality assessment and correction tool for genome assembly	(Hunt, Kikuchi et al., 2013)
Sequence mapping	IMAGE	Improves draft assemblies by iterative mapping and assembly of short reads to eliminate gaps	(Tsai, Otto et al., 2010)
	PRICE	Uses paired-read information to iteratively increase the size of existing contigs	(Ruby et al., 2013)
	Ragout	Assembly using contigs/scaffolds and multiple references	(Kolmogorov, Raney et al., 2014)
	Multi-CAR	Contig scaffolding using multiple references	(Chen, Chen et al., 2016)
	MeDuSa	Multi-draft based scaffolder	(Bosi, Donati et al., 2015)
	BWA	Mapping low-divergent sequences against a large reference genome	(Li & Durbin, 2009)
	MUMmer/MUMmer2	rapidly aligning entire genomes, whether in complete or draft form	(Delcher et al., 1999, Delcher et al., 2002)
Annotation	MEGAnnotator	Microbial genomes assembly and annotation	(Lugli, Milani et al., 2016)
	DIYA	Annotation of bacterial genome sequences	(Stewart, Osborne et al., 2009)
	RASTtk	Annotate bacterial and archaeal genomes with custom pipeline and batch file submission	(Brettin, Davis et al., 2015)
	Artemis	Genome browser and annotation tool	(Carver, Harris et al., 2012)
	Prokka	Annotation of prokaryotic genomes	(Seemann, 2014)
	GAMOLA2	Processes, annotates and curates draft and complete bacterial, archaeal, and viral genomes	(Altermann, Lu et al., 2017)
Community analysis	QIIME	Analysis of high-throughput community sequencing data	(Caporaso, Kuczynski et al., 2010)
	Mothur	Analysis of high-throughput community sequencing data	(Schloss et al., 2009)
	PIPITS	An automated pipeline for analyses of fungal internal transcribed spacer (ITS) sequences from the Illumina sequencing platform.	(Gweon, Oliver et al., 2015)
	MEGAN	A comprehensive toolbox for interactively analyzing microbiome data	(Huson, Auch et al., 2007b)
	MetaPhlan/MetaPhlan2	Profiling the composition of microbial communities from metagenomic shotgun sequencing data	(Segata, Waldron et al., 2012, Truong, Franzosa et al., 2015)
	Bracken	statistical method that computes the abundance of species in DNA sequences from a metagenomics sample	(Lu, Breitwieser et al., 2017)
	ViromeScan	Metagenomic viral community profiling	(Rampelli, Soverini et al., 2016)
	GAAS	estimates Viral and Microbial average genome size and abundance	(Angly, Willner et al., 2009)

	GRAMMy	Relative abundance estimation based on shotgun metagenomic reads	(Xia et al., 2011a)
--	--------	--	---------------------

Whole genome and community analysis both require quality control and clean-up of sequence data, which include: adapter removal, read quality trimming, start/end read trimming, and contamination removal. There are many OSS tools available to achieve this, and the tools within GGOSS were selected for their appropriateness at the time of inclusion.

6.3.1.1 Quality assessment tools for sequence data

Quality checking is an important part in sequence analysis, applied before and after many tools used. FastQC is a quality control tool for high-throughput sequence data, quality issues with sequence data files that can be identified are: per base sequence quality, per sequence quality scores, per base sequence content, per base GC content, per sequence GC content, per base N content, sequence length distribution, sequence duplication length, over-represented sequences, and kmer content (<https://github.com/csf-ngs/fastqc>). Although there is more than one tool for this (seen in Table 6.1), FastQC is the most adaptable to input type, has a more practical broader use, and is commonly utilised in the scientific community, for this reason it was included into the GGOSS software.

6.3.1.2 Trimming and k-mer tools

Cleaning up sequence data can significantly improve down-stream analysis such as assembly, and there are several OSS tools available to accomplish the task. Cutadapt is one of the first early trimming tools, though originally designed for 454 sequencing the tool supports FASTQ, FASTA and SOLiD .csfasta/.qual input files. There are many OSS trimming tools for next generation sequence data, some of the most popular can be seen in Table 6.1. Of all the tools ‘Trimmomatic’ is arguably a better tool for the job, though specifically designed for illumina reads, it’s a newer tool with a number of improved functions and features, and kept up-to-date. There are other newer tools that are faster than trimmomatic, such as PEAT, but such tools have been shown to have poorer trimming ability (Li et al., 2015b). Trimmomatic was not selected over Cutadapt due to both the specificity to illumina reads and the date of tool inclusion into GGOSS. Furthermore

due to GGOSS additions such as adapter lists and reverse complement removal, the cutadapt tool is made more comparable to and more flexible than Trimmomatic.

Like Cutadapt, Sickle is another trimming tool, however its primary purpose is different in that its designed using sliding windows, to trim the ends of reads based on quality and length thresholds (<https://github.com/najoshi/sickle>). Although there are many OSS trimming tools available, Sickle is unique in this sliding window, quality and length thresholds aspect, which is the reason for its selection and inclusion into GGOSS.

Khmer is a software library and toolkit for k-mer based analysis and transformation of nucleotide sequence data (Crusoe et al., 2015). The tool implements a probabilistic k-mer counting data structure, a compressible De Bruijn graph representation, De Bruijn graph partitioning, and digital normalization (Crusoe et al., 2015). There are many other tools available for k-mer counting, as seen in Table 6.1, khmer was chosen because it was competitive and flexible, as shown in the tool comparison paper by Qingpeng Zhang *et al* 2014. Future versions will include a selection of k-mer counting tools, as khmer has been shown to have increasingly incorrect k-mer counting when available memory is low or a data set is too large (Zhang et al., 2014).

6.3.1.3 Sequence similarity alignment tools

Identifying regions of similarity between biological sequences is an important step in analysis, it helps confirm and/or identify a given samples taxa. There are several tools for this, the most prominent is BLAST, which has become a standard tool in the scientific community for genomic sequence alignment. For this reason, as well as the range and flexibility of the tool, BLAST was incorporated into GGOSS. Other tools can be seen in Table 6.1, the most comparable to BLAST is 'kraken'. Kraken is a faster alternative to BLAST as it runs and stores everything in RAM, the obvious downfall of this is that it relies heavily on the amount of RAM available. Although BLAST has a broad application that also spans translation to protein and protein searches, there are more specialised tools that are more efficient in a given area, such as DIAMOND (Buchfink, Xie et al., 2015), these are not currently included, but future updates intend to make ever increasing numbers of these tools available.

6.3.1.4 Genome assembly tools

Genomic assembly is integral to whole genome shotgun sequencing analysis. Because of the necessity of assembly and varied difficulties dependent on genome/sample type, there are many available assemblers. Some of the most common place assembly tools can be seen in Table 6.1. Although there are several assembler tools, SPAdes was selected for this beta version of GGOSS. Other assemblers such as velvet have been prominent in the past, though many are no longer maintained or improved. SPAdes has entered the fore-front in assembly tools, with frequent updates and improvements. SPAdes has a broad set of commands for a variety of data types, each shown to have competitive accuracy and speed. However a relatively recent assembler 'BASE' has been built with the increased read lengths of current next gen sequence technology (Liu et al., 2016). BASE is improved on other assemblers and comparable to SPAdes for longer read length, while significantly improving assembly speed, taking as little as ~25% of the time. As such, BASE is considered for future inclusion into the program.

6.3.1.5 Quality assessment tools for genomic assemblies

QUAST is another quality checking tool, its used to assess genomic assemblies, providing insight on misassemblies and structural variations, genome representation and its functional elements, and variations of N50 based on aligned blocks (Gurevich et al., 2013). QUAST provides quality assessments but does not provide the tools to address them. There are few OSS command line tools for genomic assembly evaluation, QUAST was selected as its commonly used in genome assembly comparison (Alhakami, Mirebrahim et al., 2017, Loman, Quick et al., 2015, Scott & Ely, 2015), its simple to use, and it provides a core set of quality scores. QUAST was chosen over GAGE, as it can perform quality assessment of assemblies with or without a reference genome. Like QUAST, another assembly assessment tool 'REAPR' can evaluate the accuracy of an assembly using mapped paired end reads, without the use of a reference genome for comparison (Hunt et al., 2013). QUAST and REAPR are similar tools, however as REAPR provides the means to start addressing assembly error, because of this, future inclusion of the tool is planned.

6.3.1.6 Sequence mapping tools

Sequence mapping can help improve contigs (Lischer & Shimizu, 2017), identify likelihood of assembly errors, genome comparison, and contaminant removal (Ekblom & Wolf, 2014). It should be noted that when confirming quality of assembly, sequence mapping should not be the only tool used, as it has been shown to over-estimate the lack of assembly errors (Lehri, Seddon et al., 2017). There are many tools that apply sequence mapping, several have been included into GGOSS to cover the range of sequence mapping uses. Mapping tools included are: PRICE, Ragout, BWA, and MUMmer (see Table 6.1).

PRICE is included in the program because unlike other tools it uses mapping to improve contig size by iteratively mapping reads to existing contigs, this is particularly useful for metagenomic subcomponents of interest, like bacterial viruses. Another advantage of PRICE based sequence mapping is that it does not require a reference genome. Other tools that use sequence mapping in a similar manner include IMAGE (see Table 6.1).

Ragout is a very different mapping tool to PRICE or IMAGE, and more similar to most other mapping tools in that it requires reference genomes to map against. Ragout was included into GGOSS as it improves contig construction and reduces assembly gaps. It does this in part by its more unique feature of mapping to multiple reference genomes and their evolutionary relationship. There are many tools for mapping to a reference genome, but few that take into account several references in this manner. Other tools that carry out sequence mapping using multiple reference genomes in this way include Multi-CAR and MeDuSa. Multi-CAR is the revised version of 'CAR' a single reference-based scaffolding tool, Multi-CAR is a particularly new tool that has been shown to out-perform in sensitivity, precision, genome coverage, scaffold number and scaffold N50 size (Chen et al., 2016). Multi-CAR is not incorporated into GGOSS as it was not available at the time of inclusion.

BWA is a more traditional mapping tool, in that it aligns sequences to a single reference genome, BWA also outputs the alignments in the SAM format, which allows for further downstream analysis using the SAMtools software package. BWA was included in GGOSS for alignment where a single reference genome is sufficient, allowing faster alignment.

MUMmer is incorporated into GGOSS because unlike other built in alignment based tools, it uses read alignment in its process for comparing genomes. There are a number of other tools for genome comparison such as AVID, SSAHA (Ning, Cox et al., 2001), MGA (Hohl, Kurtz et al., 2002), LAGAN (Brudno, Do et al., 2003), and BLASTZ. MUMmer was selected for incorporation over the other similar tools as its arguably the most comprehensive, most frequently used in research, and kept most up-to-date, with its latest version (4) being released in 2017.

6.3.1.7 Annotation tools

The aim of annotation is to identify genes and their products, this is an important analysis step fraught with difficulties from SNPs to SVs. Due to this there is a range of annotation tools available, each focusing on varying annotation types i.e. viral, bacterial, and eukaryotic annotation, see Table 6.1.

Artemis, built and maintained at the sanger institute, was included as an annotation tool as it's well established and commonly used for bacterial and eukaryotic annotations in research (Carver et al., 2012). Furthermore Artemis is written in Java and wrapped in its own GUI, making it easily installable and usable. Artemis's already functioning GUI program means only a button for Artemis has been included into GGOSS which opens the Artemis GUI. The access of Artemis from GGOSS is important, as the aim of GGOSS is to create a single point from which all genomic analysis can be carried out.

Prokka is also incorporated, with the purpose of providing a more bacterial focused annotation tool. Prokka is usable for bacterial, viral and eukaryotic annotations, but was purpose built for bacterial annotation (Seemann, 2014). There are no installable OSS viral (prokaryotic and eukaryotic) specific annotation tools for whole genome annotation that are not target/reference specific. Because of this RastTK is intended to be incorporated due to its more flexible pipelines that allow for more user/genome specific annotation (Brettin et al., 2015). It should be noted that though no viral specific OSS tool for genome annotation is downloadable, the html based web server MetaVir2 is capable of viral whole genome annotation. A new tool for annotation 'MEGAnnotator' shows promise as an improved tool for bacterial genome annotation, as it benefits

from reduced ambiguous annotations and annotation of metagenomic assemblies (Lugli et al., 2016). For these reasons the tool would be considered for future inclusion.

6.3.1.8 Community analysis tools

Culture independent community analysis has developed into a common place, high-throughput tool in biological research over the last 2 decades, due to modernisms that have transformed the process into an easy, low cost procedure. There are a variety of tools available for community analysis, the most frequented and robust are specialised in the bacterial microbiome, perhaps in part due to the use of 16S rRNA community research prior to next-gen high-throughput sequencing. The most common tools for bacterial community analysis are QIIME and Mothur (see Table 6.1). Both QIIME and Mothur create operational taxonomic units from amplicon sequence data of hypervariable region/s of the 16S rRNA.

QIIME was not incorporated into the program due the extent of its language complications caused by numerous version requirements of high level languages like python. The difficulties with QIIME are well known, and those who have been involved in its construction have had difficulties in its use, as such it's only practical to use and install QIIME via their more recent construction of a linux virtual environment that has been purpose built for QIIME (http://qiime.org/install/virtual_box.html). Mothur was selected for inclusion into GGOSS due to its simple installation, and its common and well tested use in the scientific field.

Viral and fungal community analysis is less commonly used than bacterial community analysis, partly due to greater complications in its methodology. Due to less investigation into viral and fungal communities, and the associated methodological complications, they currently lag behind in bioinformatic support. Though there are far fewer tools for these communities, several multipurpose tools can be used to analyse whole genome data, these include MEGAN and MetaPhlAn (Table 6.1). More recently there have been a few tools built for specialised community analysis, these include PIPITS and ViromeScan (Table 6.1).

PIPITS is an automated pipeline for the analysis of fungal ITS sequences from the Illumina sequencing platform (Table 6.1). This pipeline is the only current OSS dedicated for fungal ITS

sequences, which is why it's built into the program. There are other tools usable within the program for fungal analysis, this is necessary as although PIPITS is capable of extracting subregions of ITS, its still quite strict to the individual ITS region types (ITS1 and ITS2) and complications can arise in its use on ITS regions acquired from customised methodologies/primers. Though Mothur was specifically designed for bacterial community analysis, fungal community analysis can also be carried out when running the data against a fungal database. Problems with using the Mothur pipeline, is that it's not designed to account for the significantly variable ITS region sizes, which may lead to poor analysis.

Like fungal analysis, there are many difficulties involved in virome analysis. Most techniques for virome characterisation tend to underestimate the diversity and quantity of viruses in a community (Mokili, Rohwer et al., 2012). Methods for viral isolation that use filtering procedures miss giant virus (Colson, Fancello et al., 2013), and viral communities are difficult to characterise since there is no conserved genomic region within all viral genomes that can provide taxonomic classification (Virgin, 2014).

A relatively new tool 'ViromeScan' is a metagenomic viral community profiling tool (see Table 6.1), which is intended for inclusion into GGOSS in a later version. When community tools were built into the program there were no viral specific OSS tools available for viral community analysis (currently there are still no viral specific OSS tools for whole genome annotation). As such, the generic community analysis tool 'MetaPhlAn' was selected for inclusion into GGOSS. MetaPhlAn is used for profiling the composition of microbial communities from metagenomic shotgun sequencing data.

Other tools capable of community analysis from whole genome sequencing data include MEGAN. MEGAN is particularly good for metagenomic community analysis as it uses tools like GRAMMy (Xia et al., 2011a) that take into account a number of biases, providing more accurate relative genome abundance. Although a newer tool 'GASiC' is arguably more accurate for highly similar reference genomes as it directly accounts for the reference genome similarities. GRAMMy's similarity parameters are estimated from the alignment qualities of the reads to the reference genomes which mean the relative abundances are less robust. MEGAN was originally

included for community analysis, but its command line access is only provided at cost, and cannot be made accessible within the program long term. There are benefits to using MetaPhlAn over MEGAN, particularly in its flexibility in methodologies for ascertaining the community structure and abundances, however the incorporation of GASiC into GGOSS will add alternative high quality community analysis methods.

Bracken is a particularly new tool for metagenomic community analysis, which is both fast and accurate even in samples containing near identical species. Bracken is not included, due to its reliance on Kraken sequence alignment instead of BLAST.

6.3.2 Future additions to GGOSS

The aim of this chapter was to create a streamline analysis for chapter 7, as other available systems lacked the flexibility, choice of tools, and mass sample analysis. The outcome of this chapter is the creation of a novel genomic program, however external requests in its use have stimulated a push toward publication as an open source software, aka 'GGOSS'. Creation of a beta GGOSS version 1.0 is intended to be publically released in late 2018. Because of this and personal analytical needs, future plans are in place to significantly improve the program, with the inclusion of many prominent OSS tools, as well as furthering unique GGOSS tools. Tools of primary interest of inclusion include RASTtk, trimmomatic, velvet, BASE, GAM_NGS, a selection of k-mer counting tools, REAPR, MEGAnnotator, Mauve, Gepard, and ViromeScan. GAM_NGS would be part of a new section of GGOSS specialised in assembly reconciliation tools. The section for post annotation analysis is also to be built in future versions.

Future GGOSS specific tools will include automated accumulation of each community analysis output for when more than one community has been mapped (i.e. bacterial, viral and fungal). This allows simplified plot forming and trend identification. Other additions would include: selectable automation of OSS updates, automated primer design for illumina SBS platform with selectable 2 step or 1 step PCR based design, automated distributed computing function, and reference list and hyperlinks of used OSS built into logfiles.

By uploading GGOSS onto GitHub the open source software will hopefully become a common place platform from which to conduct genomic analysis, that's developed and improved upon by the community.

Chapter 7. Metagenomics approaches of bacterial and viral fraction of stool samples from low birth weight, preterm neonates

7.1 Introduction

Significantly preterm infants (< 28 wks gestational age) are dependent largely on innate immunity at birth (Levy, 2007, Strunk, Currie et al., 2011a) and are at greater risk of developing necrotising enterocolitis (NEC) and late onset sepsis (LOS) (Hsueh, Caplan et al., 2003, Lin & Stoll, 2006). Gut microbes are a key component of NEC and LOS development (Morowitz, Poroyko et al., 2010, Neu & Walker, 2011, Stewart, Embleton et al., 2017) and as such have been the focus of intensive study over recent years (Dittmar, Beyer et al., 2008, Mai, Young et al., 2011b, Morrow, Lagomarcino et al., 2013b, Stewart, Skeath et al., 2015, Zhou, Shan et al., 2015a). La Rosa et al. (2014) demonstrated non-random, patterned development of microbial communities in the preterm infant gut suggesting factors associated with gestational age at birth (La Rosa, Warner et al., 2014). One potential mechanism contributing to NEC and LOS development is imbalance between pro- and anti-inflammatory mechanisms in the gut leading to increased gut permeability and bacterial translocation or tissue damage and 'NEC'. Despite the depth of research, including several large-scale analyses of bacterial communities in the preterm infant gastrointestinal tract, reviews and meta-analyses (Pammi, Cope et al., 2017, Warner, Deych et al., 2016b), a universal bacterial pathogen has not yet been identified, nor has any single repeatable 'pattern' of gut dysbiosis been illustrated. This has promoted theories about community dysbiosis, microbial and metabolic instability in disease development (Pammi et al., 2017, Stewart, Embleton et al., 2016). Attempts have also been made to link fungal (Hallstrom, Eerola et al., 2004, Karlowicz, 1993, Stewart et al., 2013a) and viral (Chany, Moscovici et al., 1982), (Williams, Kadambari et al., 2014) communities with health and disease in a neonate background.

There is symbiotic relationship between bacteria and viruses in symphony within the gut that are important for the development of healthy epithelium and immunity. From the literature it is clear the bacteriophages are an understudied part of the microbiota. This is surprising, as Breitbart

et al. (2008) illustrated phage abundance in the infant gut a decade ago (Breitbart et al., 2008). Phages have the ability to infect, and lyse or integrate into the bacterial chromosome, which in turn provide several means of influence that phage can have upon the bacterial community. Each viral infection enriches the population of phage which in itself may confer selective pressure on colonising bacteria. There are multiple studies showing how bacteriophages alter microbial communities (Koskella & Brockhurst, 2014). They have shown the expansion and shift in the composition of phage taxa over a 24-month period post birth (Lim et al., 2015b). However, whether or not this transition is associated with a stabilising gut microbiota remains unclear.

7.1.1 Next generation sequencing technologies

Next-generation sequencing (NGS) is a massively parallel sequencing technology, where advancement in DNA sequencing chemistry over the last two decades has transformed it into a practical and mainstream approach in biological sciences. NGS technology has; high throughput, scalability, speed, simplicity and low cost, that enables researchers to carry out a broad variety of applications and investigate biological systems in a previously unachievable manner.

All of the research used in this thesis uses the Illumina Miseq and sequencing by synthesis (SBS) chemistry. After sample preparation and sample loading the Miseq workflow can be categorised into three major steps; clustering, SBS/imaging, and data-sorting/demultiplexing. The basic principles of SBS can be seen in Figure 7.1. Under isothermal amplification, complementary binding of fragmented samples to flow cell oligonucleotide attachment sites occurs. Sample fragments acquire their complementary oligonucleotide flowcell attachment sites during sample prep fragmentation steps. There are two different attachment sites (3' and 5'), associated to forward and reverse reads. Fragments only hybridise to the complementary 5' oligonucleotide attachment sites, as the 3' attachment sites on the flow cell are identical to the 3' attachment sites on the sample fragments. DNA polymerase creates a complement of the hybridised fragment, and the original fragment is denatured and washed away, leaving behind the complementary template DNA strand. The 3' end of the template strand is now complementary to the 3' oligo flowcell attachment site. This allows bridge amplification to now occur. DNA strands bend to secondary aforementioned attachment sites, and while bridged, polymerisation forms a complementary strand. The dsDNA

bridge is then denatured resulting in a forward and reverse strand. This process is repeated to form clusters of each fragment. After clustering the forward strands remain, while the reverse strands are cleaved and washed away, after which the 3' end is blocked to prevent unwanted priming.

SBS and imaging now commences by attaching a sequence primer to the DNA templates. Starting from the primer, each cycle is represented by a single fluorescently tagged reversible terminator nucleotide binding to the DNA template. With each bound nucleotide, synthesis stops, the nucleotide is fluorescently excited by a light source (LED or laser, depending on the sequencing platform), and an image is taken, and synthesis continues. Upon completion of maximum cycle number the read products are washed away and their template strand associated index read 1 primers are read. The 3' ends on the flow cell are unblocked and the template strands bridge, the read 2 index is then read in the same manner. While bridged a complementary strand is synthesised with DNA polymerase, and the bridge is denatured to create two linearised strands. The 3' ends are blocked again, and the original template strand (forward strand) is washed away, leaving just the reverse strands bound to the flow cell.

On completion of sequencing, if multiple samples are present with individual barcoding demultiplexing must take place; this is where the sequences are pooled and separated based on their indexes that were introduced during sample preparation. Reads that have similar base calls are locally clustered. The forward and reverse reads are paired thus giving paired end contigs in the form of read 1 and read 2. Finally the reads are demultiplexed (removal of indexes) (Illumina, 2016).

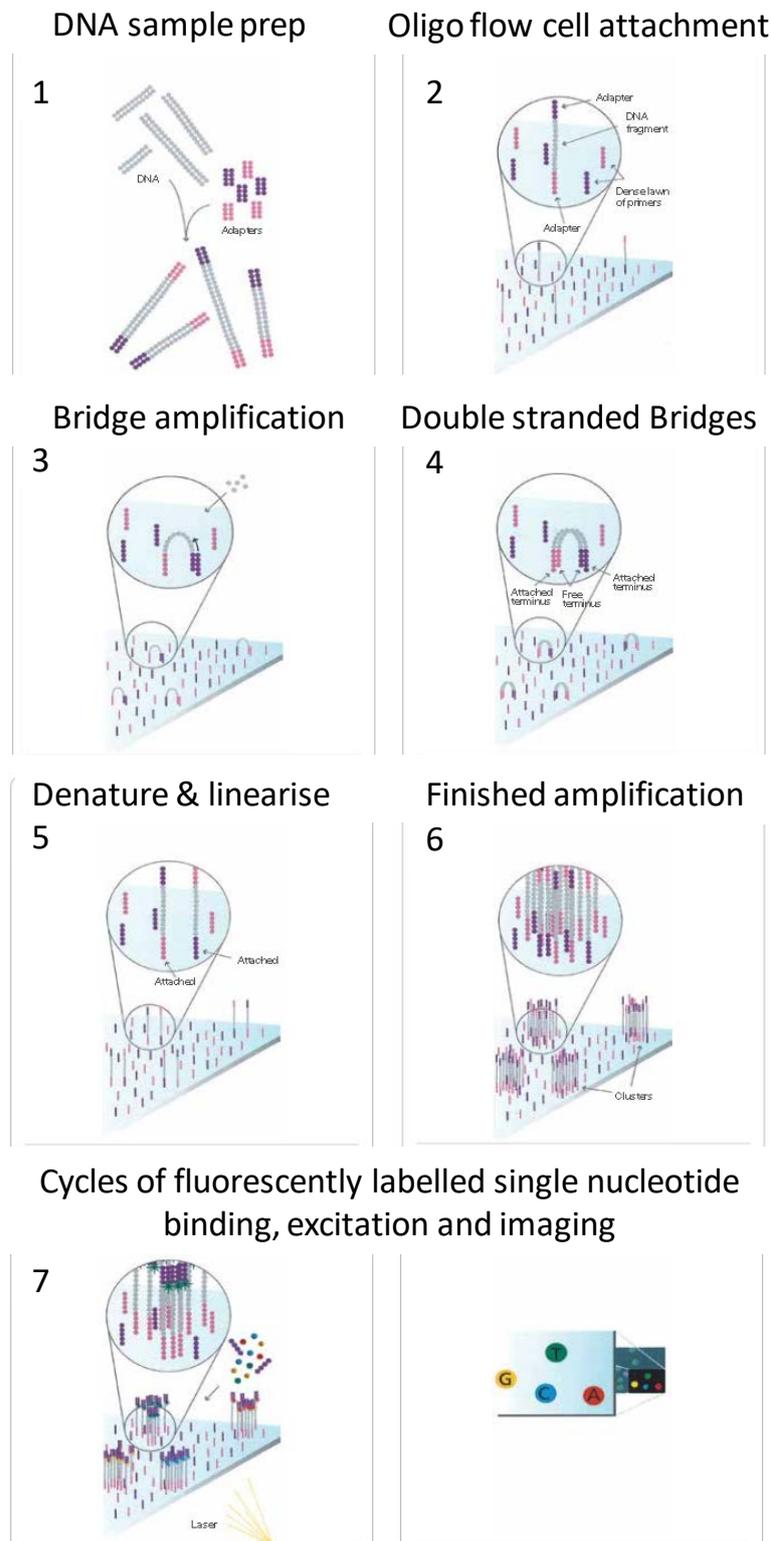


Figure 7.1 Illumina SBS workflow. Core steps involved in the sequencing by synthesis technique employed in Illumina sequencing technology, from sample prep to nucleotide imaging and data collection. Edited from the online Illumina documentation (Illumina, 2010)

7.1.2 Genome sequencing analysis

Prior to any sequencing analysis files need to be cleaned of short and/or low quality read issues, oligo flowcell attachment site presence, barcode presence, and contamination. With the multitude of problems associated with the quality of sequencing data, there are likewise several clean-up methods and quality checking tools to broach this, these include; seqtk (<https://github.com/lh3/seqtk>), FastQC (<https://github.com/csf-ngs/fastqc>), khmer (<https://github.com/dib-lab/khmer>), Trimmomatic, Cutadapt, etc. (Bolger et al., 2014, Gurevich et al., 2013). For further information see methods section 2.17.4 for the approach used in this study, and see section 6.3.1 for more details on the bioinformatic tools. Once the quality sequencing data is achieved, downstream assembly or read mapping can occur.

7.1.2.1 Bacterial community analysis

Bacterial community analysis using target gene amplification is a common practice in modern biological science. The technique is far easier than culture dependent based methods, and identifies the community diversity and abundance with far greater depth due to the inability to culture all bacteria from a sample. However pitfalls in the SBS genomic technique also exist, where bias can be introduced in numerous ways that include; GC rich samples (Chen, Liu et al., 2013), short sequence clustering (Dohm, Lottaz et al., 2008), PCR amplification (Aird, Ross et al., 2011), and non-viable cell inclusion (Rogers, Marsh et al., 2010). Currently the two core genomic community analysis approaches are targeted amplicon sequencing and untargeted, whole shotgun sequencing. See method section 2.17.4 for this studies approach, and see section 6.3.1 for more details on the bioinformatic tools.

7.1.2.1.1 Target sequencing community analysis

This approach targets conserved target region on bacterial genomes that has genus level variability that can be utilised to show differences in the bacteria present within the sample. This has moved from amplification and techniques that separate the communities on gels, by denaturation of the amplicon by the use of a chemical denaturant (DDGE) (Muyzer, de Waal et al., 1993) or temperature (TTGE) (Vasquez, Ahrne et al., 2001). We now sequence this data where

Illumina DNA sequencing > ~200 samples can be run at a time on any given miseq offering enough sequencing depth to complete downstream analysis.

In bacteria the most common targeted gene is the 16S ribosomal RNA (rRNA) gene, which consist of nine hypervariable regions, separated by nine highly conserved regions (Baker, Smith et al., 2003, Wang & Qian, 2009). It has been shown that the central regions of the full-length 16S rRNA gene sequence (V4-V6) are the most reliable regions for representing it in the phylogenetic analysis of most bacterial phyla, while V2 and V8 were the least reliable regions (Yang, Wang et al., 2016a).

In fungi the conserved gene sequence containing hypervariable regions targetted for community analysis is the ribosomal internal transcribed spacer (ITS) region (SchochSeifert et al., 2012). There are two ITS regions in fungi, ITS1 and ITS2. The ITS1 region is located between the 18S and 5.8S rRNA genes, and the ITS2 region is located between the 5.8S and 28S (White, Bruns et al., 1990). In both bacterial and fungal target based sequencing, improved accuracy in abundance and taxonomy can be achieved by using sets of primers to cover more of the target region.

7.1.2.1.2 Analysis of microbial community amplicon sequencing data.

Tools for targetted community analysis include QIIME (Caporaso et al., 2010), Mothur (Schloss et al., 2009), and PIPITS (Gweon et al., 2015), where the majority of publications are associated to those used for bacterial community analysis, particularly Mothur. This is likely due to the user friendly nature of the Mothur pipline, where QIIME can be somewhat complex to set up initially. The difficulties with QIIME are well known, especially installation, which has been overcome somewhat with the QIIME programmers designing a linux virtual environment purpose built for QIIME (http://qiime.org/install/virtual_box.html). PIPITS is an automated pipeline for calculating taxonomy and abundance of fungal communities from ITS sequences on the Illumina sequencing platform. PIPITS is currently the only open source software dedicated for fungal ITS sequences analysis.

Mothur was initially (and still fundamentally) designed for bacterial community analysis, fungal community analysis can be carried out against a fungal ITS database, such as UNITE.

Problems with using the Mothur pipeline for fungal analysis, is that it's not designed to account for the significantly variable ITS region sizes.

7.1.2.1.3 Whole genome shotgun sequencing; community analysis

The whole genome shotgun (WGS) sequencing approach by SBS (described in section 7.2) can also be used for microbial community analysis. There are multiple programmes to study the taxonomy of bacteria, viruses and fungi in this sequence data including; MEGAN (Huson et al., 2007b), ViromeScan (Rampelli et al., 2016), MetaPhlAn (Segata et al., 2012, Truong et al., 2015), and Bracken (Lu et al., 2017). Most tools like this use local searches against non-redundant database files as input. These are then used for each sequence to determine their lowest common ancestor (LCA).

WGS sequencing provides greater taxonomic classification strength, as well as a more accurate interpretation of microbial presence. However, unlike targeted methods, WGS sequencing analysis relies on estimated relative abundance, which is a challenging computational problem because of the complexity and differences in genome sizes present. One problem is the large numbers of short reads that cannot be uniquely mapped to a comparator within the database, a specific location on one genome, instead mapping to multiple locations on one or multiple genomes. Another problem is that microbial communities have large numbers of microbes with similar genomes. Nonetheless, existing tools have been improved as well as novel tools created, addressing many of these issues as best as possible. For example, MEGAN now uses GRAMMy, which uses algorithms like Maximum Likelihood Estimation (MLE) of the genome relative abundance (GRA) levels. This takes into account the probability of read assignment to genomes (Xia, Cram et al., 2011b).

7.1.3 Viral metagenomics or viromics

A key component of the microbiota that is sometimes overlooked in microbial communities are viruses. There are no ubiquitous genes present on viruses and therefore we need to use these metagenomics shotgun approaches. Within the viral fraction of the neonate gut there will be both bacterial and eukaryotic viruses that can be classified as part of the Baltimore classification that is based upon the carriage of nucleic acid within the viral capsid; dsDNA, ssDNA, dsRNA, ssRNA (+/-), rtDNA, and rtRNA (see appendices section 10.6.1-10.6.4). Their role in the structure and influence on the microbial communities of the gut are difficult to ascertain. This will improve as larger studies look to compare the viral and bacterial communities back to health and disease.

7.1.3.1 Viral types: Prokaryotic (bacteriophage) and eukaryotic viruses that could be found in the gut

Viruses infect all forms of cellular life, and are the most physically abundant and genetically diverse entities in the biosphere. With the phage population alone estimated at $\sim 10^{31}$ particles, viruses outnumber cells by multiple orders of magnitude (Whitman, Coleman et al., 1998). The virome has a significant impact on geological (Pacton, Wacey et al., 2014) and ecological systems (Rodriguez-Brito, Li et al., 2010), shaping the biome as we know it, manipulating the metabolic (Holt et al., 2017) and genomic scaffolds (Moelling, 2013), lysing cellular populations, killing multicellular organisms, and driving evolution (Moelling, 2013). The immediate impact of viruses is most apparent in mammalian health. However viruses can have diverse and far reaching impact. Examples of viral impact on diversity can be seen in the energy conversion in the biosphere and sediment formation in water bodies by killing off populations of abundant, ecologically important organisms such as cyanobacteria or eukaryotic algae (Fuhrman, 1999, Rohwer & Thurber, 2009, Suttle, 2007b), as well as viral diseases of pollinating insects having a role in our food security, ecosystem system and biodiversity (Manley, Boots et al., 2015). Viral genome size range from ~ 4.5 kilobases (kb) (Brunham, R.C. et al., 2000) to over 2 megabases (Mb) (Philippe, N. et al., 2013). Though individual viral genomes are relatively small compared to their host eukaryotic and prokaryotic counterparts, the dominant abundance of viral particles make up the majority of the biospheres genetic diversity (Hendrix, 2003, Kristensen, Mushegian et al., 2010, Kristensen, Waller

et al., 2013). The extent of viral biology and their interaction with their host cell, the dynamic excessive abundance, diversity, and selfish genetic biology, makes a strong argument for virus-host coevolution playing a major role in the evolution of bacteria and their colonising environment since life began (Forterre, 2005, Szathmary & Demeter, 1987, Takeuchi & Hogeweg, 2012).

Viruses were originally classified solely on their morphology, the template of which was formatted by David Bradley and Hans Ackermann. A new form of viral classification was proposed in 1962 by Lwoff, Horne, and Tournier (Lwoff, Horne et al., 1962) which attempted to box viral taxonomy in a similar structure as bacterial taxonomy, i.e. the Linnaeus system (phylum, class, order, etc). This approach was partially taken on, and viral classification of these groups is a combined morphological and genomic typing approach. This current classification system is polythetic, which is a necessity due to the genetic mosaicism of viruses. This viral taxonomy system is governed/regulated by the ICTV. The varying viral genomic types cover every known and conceivable nucleic acid construct. Their genomes and replication cycles can be represented by RNA, DNA, or retro-transcribing (see Figure 1.13). RNA or DNA viral genomes can be either double-stranded (ds) or single-stranded (ss), positive or negative sense, circular or linear, and consist of single or multiple segments (Agol, 1974, Baltimore, 1971). The diversity of viral genomes differ depending on the type of cellular life they parasitize, this is most notable in Figure 1.11. Viral structures are also very varied, where many general and viral type associated morphologies can be observed in Figure 1.12.

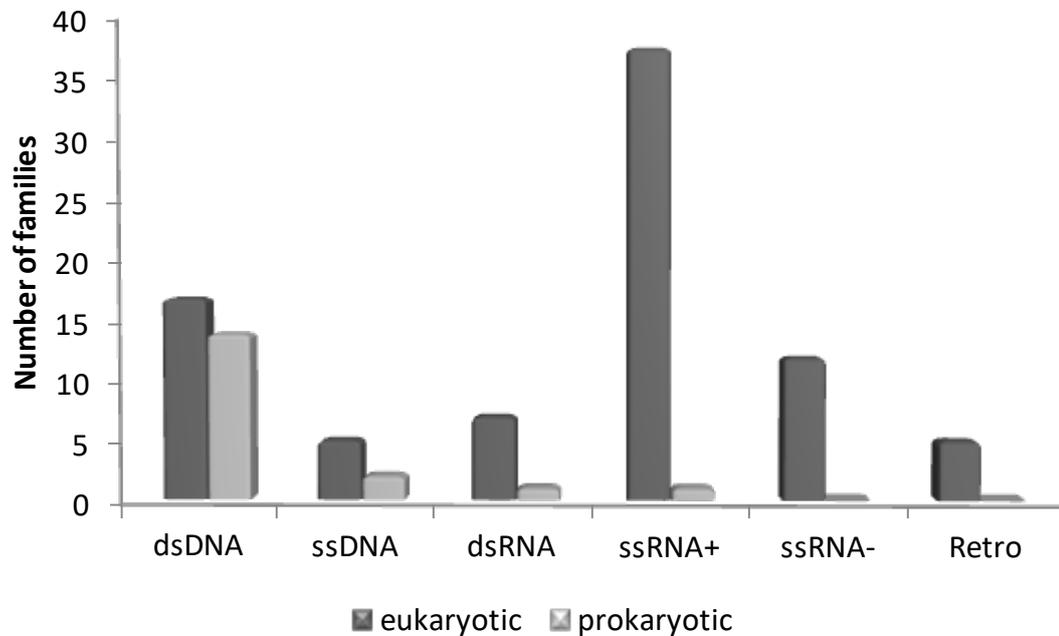


Figure 7.2 The number of currently known prokaryotic and eukaryotic viral families within each viral genomic type. The bar plot illustrates the difference between currently known eukaryotic and prokaryotic diversity across the viral types. The plot was created from data accumulated via ICTV.

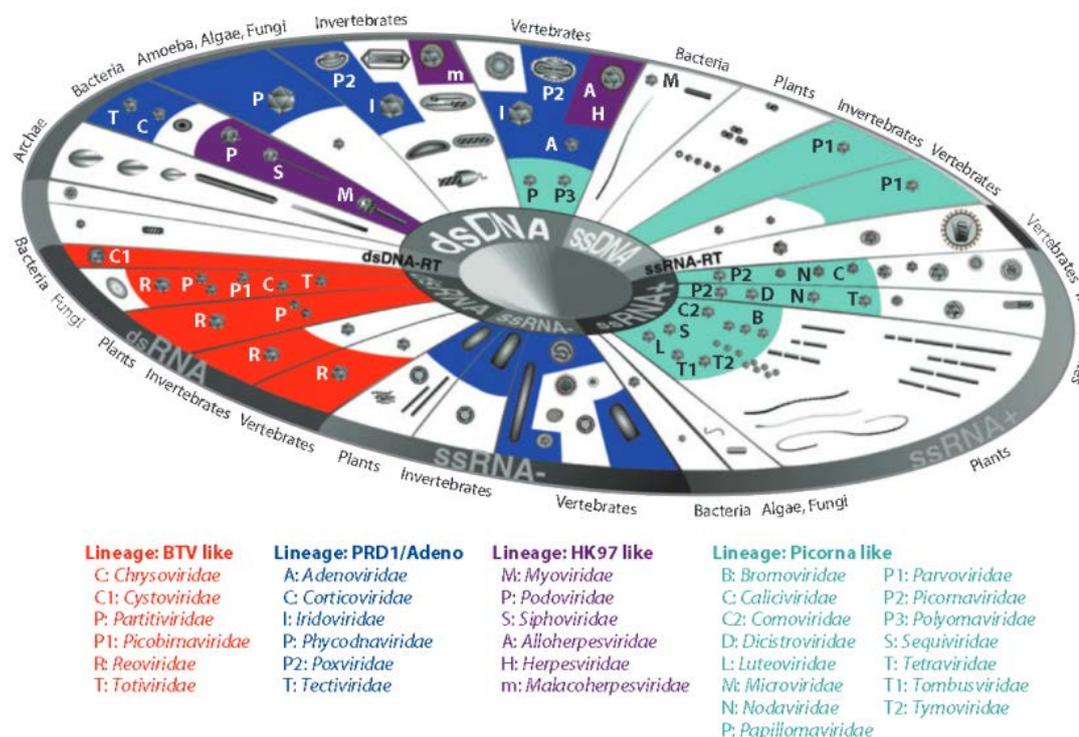


Figure 7.3 Structure-based viral lineages mapped onto current ICTV taxonomy with each lineage coloured separately. Individual viral families within each lineage are labelled and coloured according to the key. Abbreviations: BTV, bluetongue virus; HK97, bacteriophage Hong Kong 97; PRD1, prototype member of the Tectiviridae double-stranded DNA bacteriophage (Abrescia, Bamford et al., 2012)

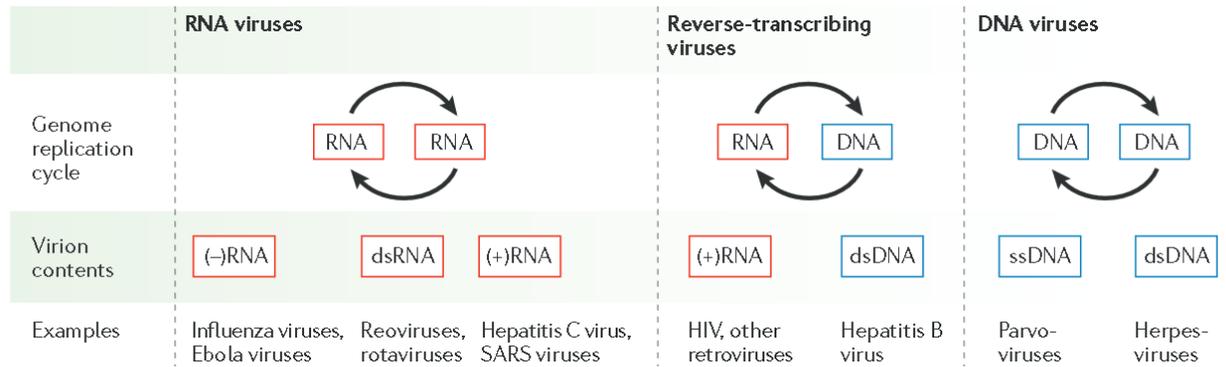


Figure 7.4 Seven classes of viruses distinguished by genome replication and encapsidation strategies (Ahluquist, 2006).

7.1.3.2 Bacteriophage

Bacteriophage (phage) are the most common and abundant viral entity, current known phage genomes are predominantly double-stranded (ds) DNA genomes most of which are in the order Caudovirales, with few single-stranded (ss) DNA viruses and limited presence of RNA viruses Figure 1.11.

Phage play key roles in microbial evolution (Bondy-Denomy, Qian et al., 2016, Canchaya, Fournous et al., 2004), human disease (Brussow, Canchaya et al., 2004), reduction and driver of disease and mortality in livestock (Cheeke, 1995, Doyle & Erickson, 2006, Tamang, Sunwoo et al., 2017), reduction and driver of disease in agricultural farming (Jones, Vallad et al., 2012) and marine nutrient cycling (Rodriguez-Brito et al., 2010). Phage and host evolution is driven by horizontal gene transfer (HGT) that occurs between phage-phage and phage-host genomes (Pedulla, Ford et al., 2003). HGT plays a key role in the significant genetic diversity of phage, which have a typically mosaic genomic architecture. Though phages have been tested in association with the medical field since the early 20th century (Davison, 1922), little is known to the extent in which they impact on the biosphere. With recent advances in sequencing technology over the last decade, more is understood about the interaction within their environments, and the extent of their importance is being realised.

Though only detected in very low abundance and frequency within the gut environment (Hoffmann, Dollive et al., 2013), archaeal viruses should be noted, if only to highlight our currently limited understanding. Archaeal viruses are the least studied viral group to date, even though they are more morphologically diverse and unique from phage (Pietilä, Demina et al., 2014). Most currently known archaeal viruses have a dsDNA genome, though ssDNA and RNA archaeal viruses have also been identified.

7.1.3.3 Eukaryotic viruses

The Eukaryotic virome differs greatly to the prokaryotic virome. In contrast to bacteriophage, the majority of eukaryotic viruses are ssRNA viruses, though eukaryotic viruses also span all viral genomic types to a far greater extent than prokaryotic viruses (see Figure 7.2). This is likely due to the additional barrier (eukaryotic nucleus) which will have given an evolutionary drive for alternative viral replication methods to DNA. The evolutionary ancestors of eukaryotic viruses are thought to originate from prokaryotic viruses due to traceable core prokaryotic gene modules in most eukaryotic viruses.

7.1.4 Aim

This study aims to employ a novel approach, testing the hypothesis that using a combination of bacterial, viral (including inducible) and fungal communities as a pan-kingdom analyses will support and resolve current bacteria-centric methods of assessing gut microbial communities. No research to our knowledge has been conducted to compare immediately processed versus frozen stored stool samples for viral communities and the ability to use stored samples. Therefore this research also aims to investigate freeze thaw and reduction in sequencing data quality. This is of practical relevance to clinical researchers with large biobanks nationally who generally rely on the ability to freeze samples and analyse in batches later. Previous studies of bacterial communities before and after freeze thaw showed limited effect (Lauber, Zhou et al., 2010, Rubin, Gibbons et al., 2013).

7.2 Results

This study, in collaboration with Clinicians Janet Berrington and Nick Embleton at the Royal Victoria Infirmary, sees fresh samples isolated from 2 sets of monozygotic twins and another baby on the neonatal intensive care unit to allow both bacterial and viral community studies with focus on the health of the child and the similiary in handling between twin pairs. The grouping and codes of neonate patients in this study can be seen in Table 7.1, abbreviations of note; twin pair (P) A and B, and time points 1 (T1) and 2 (T2).

Table 7.1 Describes clinical and demographic data for all 5 patients enrolled on this study.

Patient Code	Delivery mode	Twin pair (P)	Sex	Birth-weight (g)	Gestation at birth	NEC/LOS	Abdominal details	LOS details	Antibiotics (days exposed pre sample 1 and 2)	Full enteral feeds	Sample timing (day)
379	C/S	A	F	710	25+2	N/N			2, 18	14	26 (T1), 57 (T2)
381	V	A	F	700	25+2	N/N			7,14	14	26 (T1), 57 (T2)
386	C/S	B	F	680	24+6	N/Y	Day 5 isolated ileal perforation and stoma formation	LOS	13, 20	31	29 (T1), 60 (T2)
387	C/S	B	M	770	24+6	N/N			2	17	31 (T1), 60 (T2)
388	V	-	M	785	24+1	Y/N	Medically managed NEC day 31		5, 11	15	23 (T1), 54 (T2)

7.2.1 Fungal community analysis

Quantity of fungal sequencing reads yielded from this study limited any worthwhile analysis and is probably as a result of universal antifungal use in these infants.

7.2.2 Effect of frozen storage conditions on viral communities

Significant differences in measured variances in viral data were observed between frozen and unfrozen samples. Table 7.2 illustrates the reduction in read distribution, total taxa, and mean abundance in temperate phages in frozen samples. Interestingly there is no real significance in reduction of FVP (for a description of how FVP were attained, see methods section 2.16.3). This would play an important role in immediate and delayed processing of samples and prove pertinent

for any future experimental design. For this reason, immediately processed samples were used for the remainder of the study.

Table 7.2 Shows differences in total counts (being the sum of all counts for all taxa in all samples), total taxa (being the counts of all taxa identified across all samples) and mean counts per taxa between frozen and non-frozen samples for both temperate and lytic viral taxa.

	TEMPERATE		FVP	
	Non-frozen	Frozen	Non-frozen	Frozen
Total counts	4165	249	2120	5201
Difference	3916		3081	
Sum of squares	66.44		85.56	
Z-score	58.94		36.01	
P-value	<0.0001		<0.0001	
Total taxa	145	40	108	138
Difference	105		30	
Sum of squares	13.60		15.68	
Z-score	7.72		1.91	
P-value	<0.0001		<0.05	
Mean counts per taxa	6.30	28.73	19.63	37.69
Difference	22.43		18.06	
Sum of squares	5.92		7.58	
Z-score	3.79		2.39	
P-value	<0.0001		<0.05	

7.2.3 Bacterial community analysis

Operational taxonomic units were attained as per methods section 2.17.4. Bacterial taxonomic composition of each sample was initially assessed by comparing relative abundance at the genus and phylum level per sample (Figure 7.2). General observations regarding the microbial community differences in taxonomic composition between time points appeared to be characterised by an expansion in Actinomyces (appendices Figure 10.16). Specifically, the abundance of two observed *Bifidobacteria* taxa (appendices Figure 10.16). Further longitudinal differences in taxonomic composition of samples included the reduced abundance of Proteobacteria. This may be due to the diminished *Salmonella* population, specifically between samples 387 (PB,T1) and 387 (PB,T2). The only features of taxonomic composition that were consistent across both the 2 timepoints and all samples were; increased taxonomic richness ($P < 0.05$); (Figure 7.3) and diversity

(Figure 10.16) ($P > 0.005$). When comparing bacterial communities between clinical parameters, significant separation was observed between patients (Adonis PERMANOVA, $P < 0.01$), and twin pairs (Adonis PERMANOVA, $P < 0.005$) (Figure 7.4).

Unlike all other samples, the taxa identified as *Escherichia Shigella* genus is a major part of the core bacterial community of twin pair A (PA). In twin PA *E. shigella* makes up, on average, 39% and 19% of the total bacterial gut community at time point 1 (T1) and time point 2 (T2) respectively. In both sets of twins the presence of this taxon is reduced by T2. Neonates bacterial communities characterised by dominance of *Salmonella* or *Escherichia Shigella* in T1 show a great reduction of both by time point 2. Abundance of *Bifidobacterium* is inversely correlated with abundance of these opportunistic pathogens. Development of a *Bifidobacterium* dominant community develops individually in all but one neonate. The only patient not to follow this trend went on to develop NEC (patient 388). Furthermore this patient is the only patient not to have proteobacteria making up their core bacterial community. This is only seen in the single child rather than the twin pairs, however the hypothesis would be that the development of NEC is the confounding factor for these traits. Another universal trend between time points was the development of *Lactobacillus* (potentially from the administered probiotics) at time point 2 within gut communities.

7.2.4 Overlaying viral communities onto bacterial community analysis

Viral shotgun metagenomics analysis allows genus level determination of viruses. Analysis of data and determined relative abundances were carried out primarily in MEGAN, as per methods section 2.17.4. These results show observational variance in all viral community structures, in particular twin sets using FVP and lysogenic viral communities compared to bacterial communities (Figure 7.5). Analysis of the most abundant viral taxa illustrated more unique community profiles compared to bacterial analysis. Identical analyses of these were used to compare differences between clinical parameters in both FVP and matched bacterial communities. Significant separation was observed between patients (Adonis PERMANOVA, $P < 0.01$) and twin pairs (Adonis PERMANOVA, $P < 0.005$) using FVP and bacterial analysis (Figure 7.8). In contrast, when

comparing the lysogenic viral communities significant separation was observed by patient (Adonis PERMANOVA, $P < 0.05$), but not twin pair (Figure 7.8).

Pairwise PERMANOVA identified separation between FVP communities of the twin pairs was caused mainly by dissimilarity between twin sets (Figure 7.7). This is most likely due to the large relative abundance of *Salmonella* phage in twin pair B (Figure 7.5). The lack of significance in separation between twin pairs in the lysogenic viral communities (Figure 7.5) could be attributed to sample 386 (PBT1) grouping with samples from twin pair A as they are identified as high risk. 386 (PBT1) has much lower abundance of *Salmonella* phage in comparison to the rest of the samples taken from twin pair B, however this reduced abundance is only observed in the lysogenic viral communities, not the lytic (Figure 7.7) and therefore associated with a viable stool community.

Both the viral and bacterial communities of patients in twin pair B (386 & 387) were characterised by dominance of *Salmonella* phage and *Salmonella* bacteria, respectively (Figure 7.5). Patient 386 does not have a high *Salmonella* bacterial abundance at either time point, however *Salmonella* phage is the dominant viral taxon in both lytic and temperate viral communities obtained for this patient.

When we move to compare the same analyses in twin pair A patient 379, the dominant bacterial taxon found at both time points was *Enterococcus* (Figure 7.5). Such community dominance is reflected in the viral community, where *Enterococcus* phage account for the vast majority of the population in both FVP and Lysogenic phage fractions (Figure 7.7).

Escherichia is highly abundant in the bacterial community of twin pair A (PA), with greater abundance at T1 than T2. This varies in the viral analysis. No *Escherichia* was identified as highly abundant in the viral community of 379 (PA), furthermore the abundance in the lysogenic data for 381 (PA) increases, as opposed to decreases. The lysogenic viral community is also more diverse in patient 381 (PA) than its twin.

Patients 381 (PA) and 388, show disparity between taxa within the viral community and the bacterial community. Samples from these patients are characterised by dominance of

Pseudomonas phage in early time point samples and a transition to dominance of Enterobacteria phage in late time points. Interestingly, these two patients are the only two of the cohort to be identified by consultant neonatologists at the RVI, Newcastle as ‘high risk’ for NEC, where patient 388 went on to develop NEC.

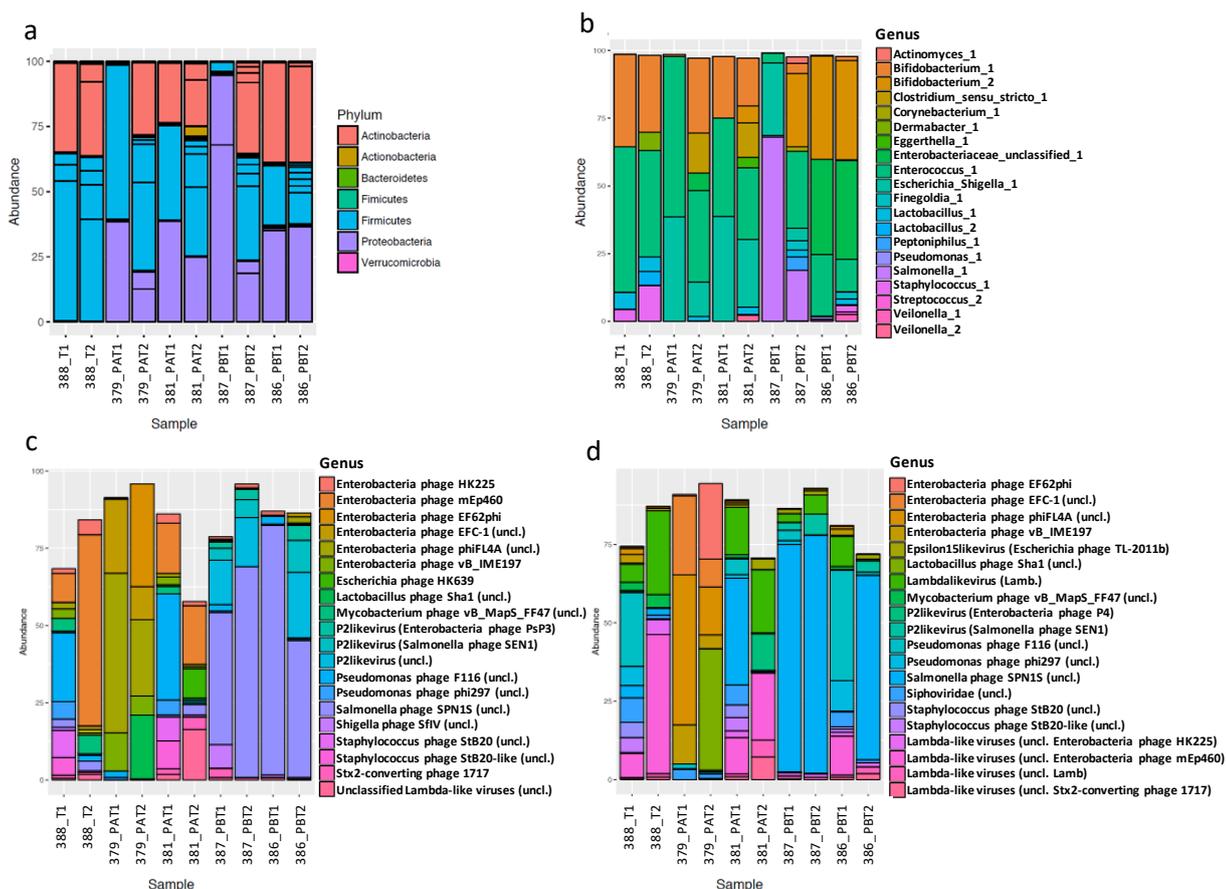


Figure 7.5 Stacked plots illustrating relative abundance of bacterial and viral communities. Shows relative abundance of both phyla (a) and the top twenty most abundant genera (b) in each sample following rarefaction to 20000 reads per sample. In addition the relative abundance of the top twenty most abundant taxa in both lytic (c) and temperate (d) viral communities.

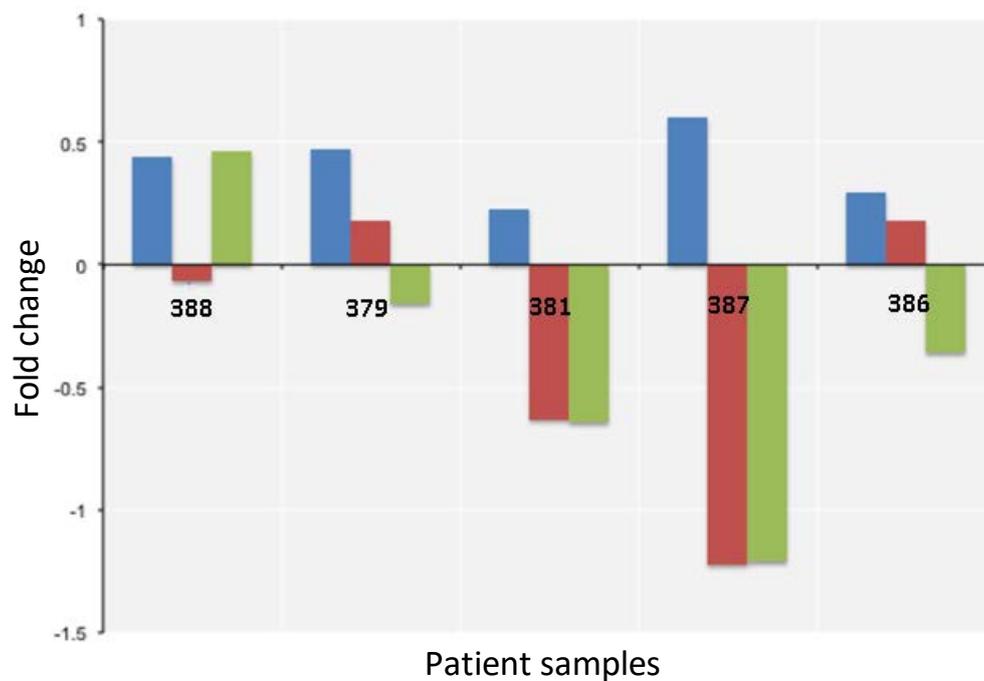


Figure 7.6 Illustrates fold change in bacterial (blue), FVP (red), and lysogenic (green) taxonomic richness between time points 1 and 2. Communities were normalised prior to comparison by rarefaction, and/or calculation of relative abundance. Taxonomic diversity was calculated by Reciprocal Simpson index and Bray-Curtis dissimilarity was used to compare communities.

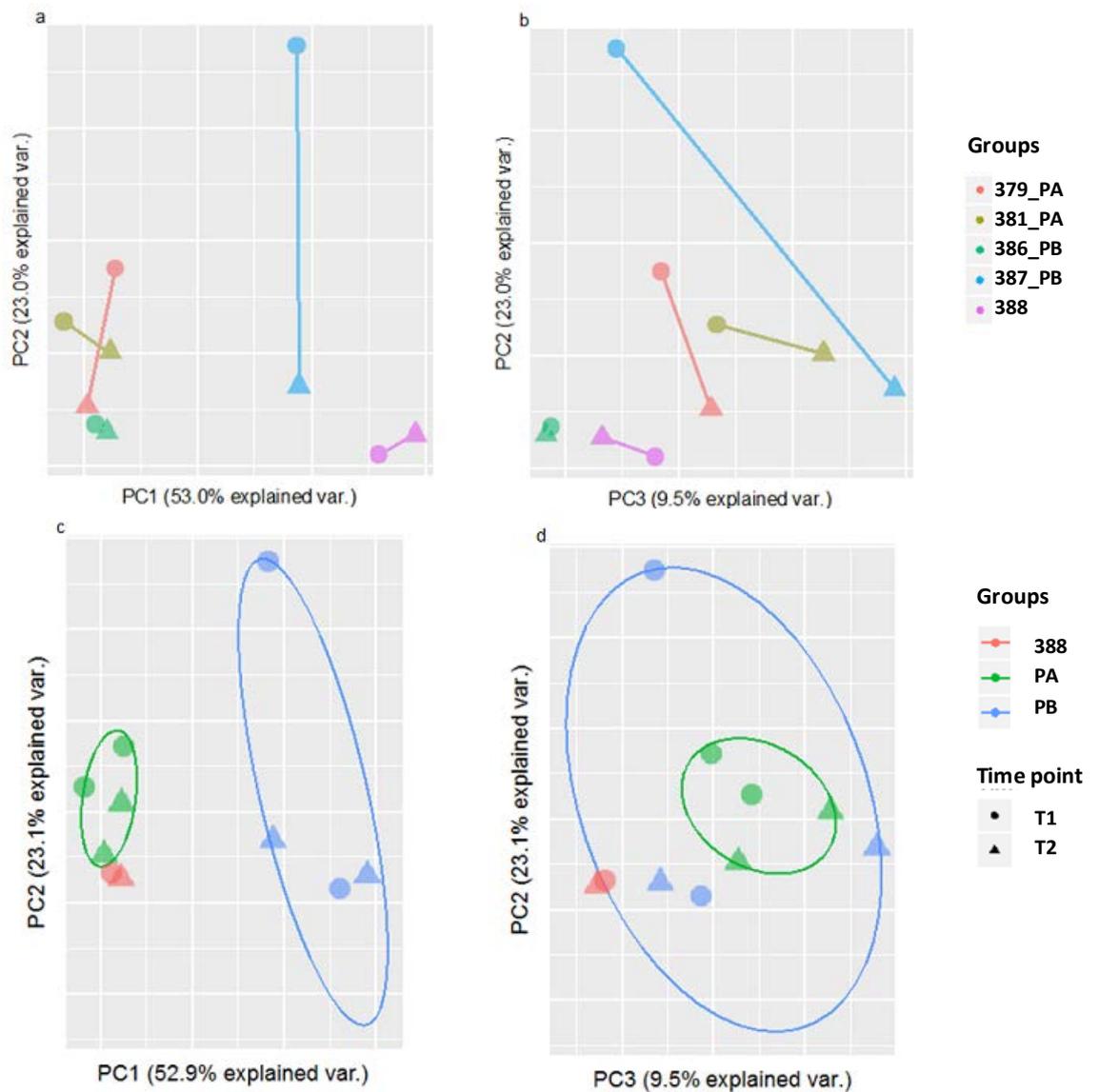


Figure 7.7 Principle coordinates analysis of bacterial communities based on bray-curtis dissimilarity. Points are coloured by patient (a. b) or twin pair (c, d). Circles represent early time points while triangles represent late time points. A total of 85.5% of the total variance is explained using the first three principle components (a - 1&2; b - 2&3). Bacterial communities are significantly different between all patients (ANOVA ($P < 0.01$)), however no significant differences were observed between individual patient pairs using pairwise PERMANOVA. In plots c and d points are coloured by twin pair and 95% confidence ellipses are included. A total of 85.5% of the total variance is explained using the first three principle components (a - 1&2; b - 2&3). Bacterial communities are significantly different between all twin pairs (ANOVA ($P < 0.005$)), pairwise PERMANOVA identified a significant difference between twin pair 1 (green) and twin pair two (blue) ($P < 0.05$) however no significant difference was observed between any twin pairs and the individual patient samples (red).

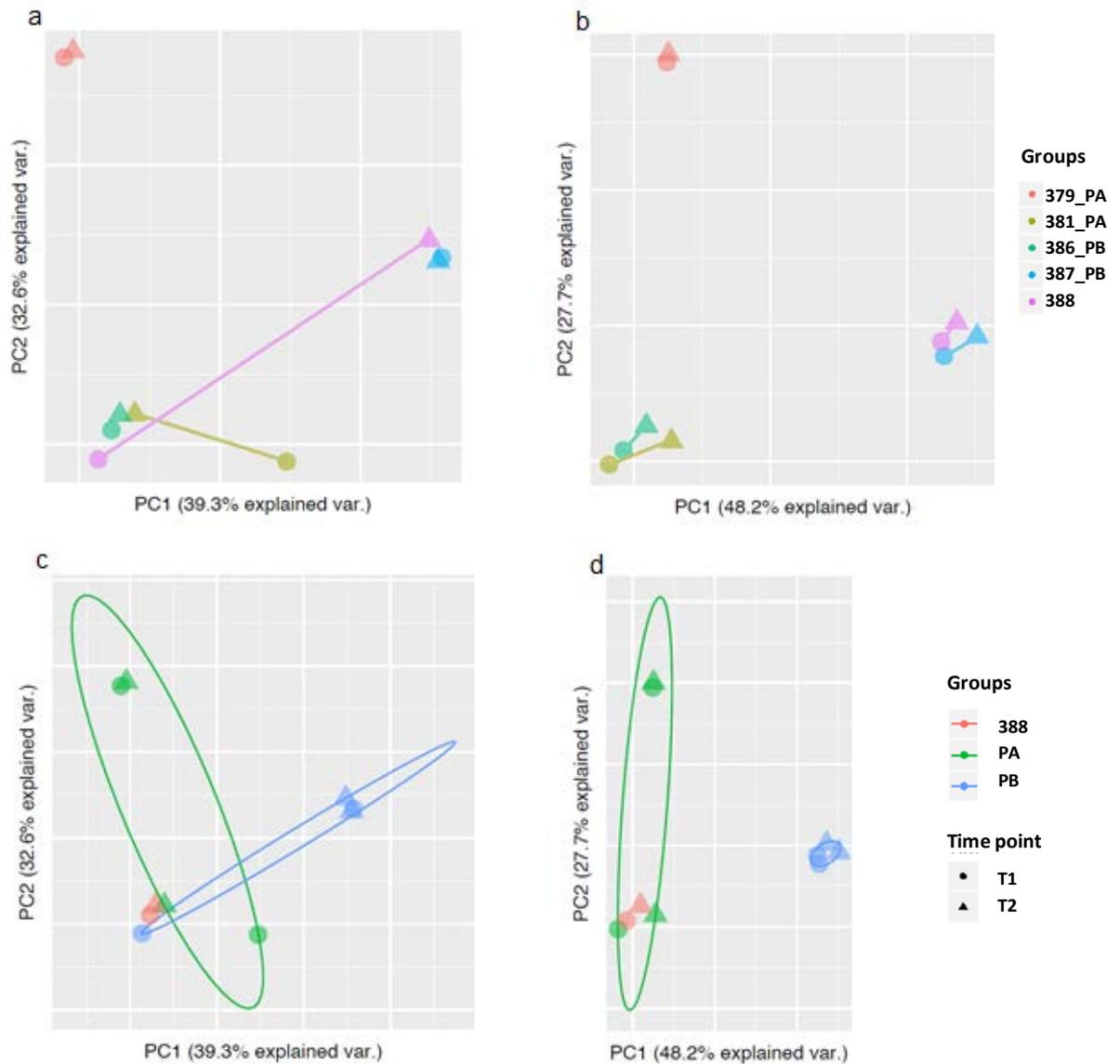


Figure 7.8 Principle coordinates analysis of temperate (a, c) and lytic (b, d) viral communities based on bray-curtis dissimilarity. Points are coloured by patient (a, b) or twin pair (c, d). Circles represent early time points while triangles represent late time points. The first two principle components are capable of describing substantial majorities of dissimilarity within the data (a, c = 72.1% total variance), (b, d = 75.9% total variance). Lytic communities (b, d) are significantly different between all patients (Adonis PERMANOVA, $P < 0.01$) and twin pairs (Adonis PERMANOVA, $P < 0.005$). Temperate communities are significantly different between all patients (Adonis PERMANOVA, $P < 0.05$) however, no significant difference was observed between any twin pairs.

7.2.5 Common comparisons between bacterial and viral communities between patients

Unidentified via bacterial 16S rRNA analysis, a direct association to the bacterial *Rhodococcus* genus was found in the inducible viral community. Six free viral particle (FVP) genera that had no infectable host within the observed bacterial community were identified. Several genera are identified in both FVPs and bacterial communities, but not the inducible virus community. The greatest number of genera associated to a single community kingdom, is found in the bacterial community. A core microbial presence (but not abundance) was identified across all communities and patients. These unique and common features are shown in Table 7.3.

Table 7.3 Highlights similarities and unique features as a whole when using FVP, inducible viruses, and bacterial community analysis within our patient cohort

Conserved	Bacterial & Free virus	Bacterial only	Free virus only	Lysogenic virus only
Clostridium	Acinetobacter	Acidaminococcaceae_unclassified	Bordetella	Rhodococcus
Enterobacteriaceae_unclassified	Stenotrophomonas	Acinetobacter	Caulobacter CcrColossus (uncl.)	
Enterococcus	Actinomyces	Actinomyces	Cyanophage	
Escherichia_Shigella	Akkermansia	Akkermansia_2	Edwardsiella	
Klebsiella	Bacteroides	Anaerococcus	Erwinia	
Lactobacillus	Bifidobacterium	Bacteroides	Vibrio	
Mycobacterium	Bilophila	Bifidobacterium		
Pseudomonas	Corynebacterium	Bilophila		
Ralstonia	Dermabacter	Brucellaceae_unclassified		
Salmonella	Devosia	Corynebacterium		
Staphylococcus		Dermabacter		
Streptococcus		Devosia		
		Eggerthella		
		Enterorhabdus		
		Faecalibacterium		
		Finegoldia		
		Gemella_1		
		Lachnospiraceae_unclassified		
		Leuconostoc		
		Methylobacterium_1		
		Methylophilus_1		
		Negativicoccus		
		Parabacteroides_4		
		Parvimonas		
		Pedobacter		
		Peptoniphilus		
		Porphyromonadaceae_unclassified		

7.3 Discussion

Previous studies in preterm neonates have focussed mainly on the bacterial communities present within the infant gut to see if bacterial taxa or community structure relate to onset of NEC and sepsis (Mai, Young et al., 2011a, McMurtry, Gupta et al., 2015, Morrow, Lagomarcino et al., 2013a, Warner, Deych et al., 2016a, Zhou, Shan et al., 2015b). Lim and Zhou et al. 2015 illustrated the longitudinal establishment of viral communities from birth in 8 twin pairs, delivered at term. They illustrate a high-predator, low prey dynamic that relates to the Lotka-Volterra model of predator-prey relationship. Lotka-Volterra model is a pair of nonlinear, differential equations, which describe the dynamics in which two species interact based on predetermined assumptions. These being; Prey population has continuously ample resources, the predator populations resources depends entirely on prey population size, the rate of change of population is proportional to its size, the environment never changes in favour of one species, genetic adaptation is inconsequential, and predators have limitless appetite (Lotka, 1932, Pearson, 1927) .

In significantly preterm infants with poor gut epithelial integrity (Beach, Menzies et al., 1982, Robertson, Paganelli et al., 1982) and an immature immune system (Strunk, Currie et al., 2011b), however, the interaction between host and gut community is different than in term infants, and the role of viruses within the community may be expected to differ. Here, we demonstrate the same trends as Lim, in that twin pairs show greater microbial community similarity. Despite limited patient numbers within this data set, we present important findings combining FVP communities, as per Lim et al. (2015), and inducible prophage metagenomics to complement bacterial community analysis. We propose this approach allows greater resolution for community comparison and understanding when studying development of the preterm infant gut in health and disease. We have limited understanding of the relationship of bacteria-phage interactions in neonates. Recent studies in adults, however, have shown phage readily cross epithelial surfaces and may be linked with conducting immune responses (Nguyen, Baker et al., 2017).

As NEC is a major cause of morbidity and mortality in preterm neonates (Allin, Long et al., 2017, Gephart, McGrath et al., 2012, Hack, Horbar et al., 1991, Qian, Zhang et al., 2017, Uauy, Fanaroff et al., 1991) there has been significant investment in understanding microbial community establishment in health as well as ‘drivers’ of dysbiosis in NEC (Mai et al., 2011a, McMurtry et al., 2015, Morrow et al., 2013a, Warner et al., 2016a, Zhou et al., 2015b). Perhaps unsurprisingly there has been no single causative microbiota associated with disease development (Abdulkadir, Nelson et al., 2016, Caplan & Jilling, 2001, de la Cochetiere, Piloquet et al., 2004, Heida, van Zoonen et al., 2016, McMurtry et al., 2015, Pammi et al., 2017, Smith, Bodé et al., 2012a, Stewart, Nelson et al., 2013b, Warner et al., 2016b). Therefore, even though it is widely accepted that disease onset is related to changes in microbiota structure or function, relating this to changes in clinical practice or using microbial markers of risk categorisation, development or progression of NEC remain enigmatic.

Lower diversity in bacterial communities in gastrointestinal disease states has been associated with NEC (Pammi et al., 2017, Warner et al., 2016a), and other gastro-intestinal pathologies such as Crohn’s (Hansen, Russell et al., 2012, Manichanh, Rigottier-Gois et al., 2006), but causation or correlation of such associations have been difficult to define. Importantly Norman and Handley et al. 2015 show increases in the abundance of both lytic and lysogenic Caudovirales and dsDNA phages: in patients with either Crohn’s or ulcerative colitis. It is known that reactive oxygen species (ROS) can induce bacteriophages (DeMarini & Lawrence, 1992). Thus, inflammation may be a driver of reduced bacterial diversity through lysogenic phage induction. We hypothesise that lysogenic phage are possible markers for gut inflammation. By mapping the complexity of the microbiota and overlaying both free viral; and chemically induced lysogenic phage communities, greater resolution is achievable to differentiate community types. Previous work in sputum has used inducing agents, including fluoroquinolones and mitomycin C (Tejedor, Foulds et al., 1982), to induce bacteriophages, enabling the lysogenic fraction to be studied. This phage population in the gut is well reviewed in Manrique and Drills et al. 2017. Employment of these further steps allows deeper understanding of the selective pressure that is placed upon the bacterial community. In addition, these techniques allow study of viral species traversing the gut and identification of those shared between matched samples

or environmental conditions. This may enable identification of potential interactions between the viral, bacterial and neonatal gut epithelium development or immune regulation. By improving the resolution of such observational studies focuses avenues for mechanistic studies.

The antibiotic used as a chemical inducing agent only stimulates bacteriophage induction from live cells. This is because the cellular machinery is needed for propagation. In this study we see this correlation as chemical induction of lysogenic phage complements identification of the viable bacterial community.

Key findings of this observational and methodological study include no significant differences in bacterial community composition when comparing patients who did or did not develop NEC or between early and late time points (assessed by Adonis-PERMANOVA).

The greatest differences between bacterial communities in this study were observed between individual patients and between twin-pairs sets. This is most probably due to the small number of samples investigated and therefore insufficiently powered to identify differences driven by other patient demographics. The observation of high levels of inter-patient variability is consistent with previous studies (Arboleya, Binetti et al., 2012, Dethlefsen & Relman, 2011, Magne, Abely et al., 2006, Mshvildadze, Neu et al., 2010). Particularly, Dethlefsen and Relman observed communities from individuals to group tightly together, even following antibiotic administration. We would suggest to successfully identify clinical treatments or temporal measures significantly associated with specific community features a much greater sample size is required.

In addition, we illustrate the additive effect of combining bacterial and viral data. Interestingly, we observe greater differentiation between viral communities than bacterial communities of samples when comparing the stacked bar graphs by eye (Figure 7.5). The significance of such differences is supported through separation on the PCoA (Figure 7.7 and Figure 7.8) and results of the Adonis-PERMANOVA, however bacterial communities are also significantly different. Greater turbulence is seen in the viral communities compared to the bacterial communities of the 2 high-risk patients, one being from twin set A. It should be noted that these are the only two patients

born vaginally. Results show an increase in bacterial taxonomic richness between the 2 time points in all patients over time despite 2 of the children being deemed as high risk for NEC. An increase in bacterial richness was observed between the two time points in infants classified as low risk. Viral taxonomic richness was reduced which compares to the development of microbial communities in the gut of full gestational age neonates described by Lim et al. (2015). The data presented here offers some inference on the viruses that are present as a reservoir within the bacterial community as prophages and these data also illustrate the viable community of bacteria within the sample as the mechanism of virus induction is based upon cell respiration. Identifying the active bacterial community within the preterm infant gut is difficult as non-invasive approaches are used which means analysis is based on stool communities. Previous work by Young et al. (2017) illustrates the bias associated with identifying bacteria without intact membranes through treatment with PMA. We use a similar approach using phage enrichment as a marker of bacterial viability.

The use of viral community analysis, particularly the novel approach of inducible and FVP communities demonstrated several interesting generalised themes across the cohort (Table 7.3.). Unidentified via bacterial 16S rRNA analysis, a direct association to the bacterial *Rhodococcus* genus was found in the inducible viral community. This could suggest a number of factors, either a very unlikely cross genus viral host range, failure of the 16S rRNA technique to pick up presence of the genus, or identification of a persistent virus after loss of its host. Persistent temperate viral strains are of particular interest, as they would provide a means retaining in the gut evolved genetic adaptation. Viral persistence may be seen in the FVP community, as six FVP genera with no sensitive bacterial host within the observed bacterial community were identified. However this may again be a lack of sensitivity from 16S rRNA. Though the greatest number of genera associated strictly to just a single kingdom community are found in the bacterial community. While this may be a cell viability associated product, it still further stresses the need for pan-kingdom analysis in the study of the gut microbiome.

Low sample number is a limitation of this study, yet the use of tools and methods to aid resolution of microbial communities is important. Bias may be introduced by only inducing

bacteriophages from bacteria that are sensitive to Norfloxacin. Furthermore, some bacteria may not have inducible phages. This is of particular interest as the mechanism of phage induction and release is multifactorial and a pertinent way to draw inference on this viral reservoir. This approach has not been previously used in the preterm neonate gut.

7.3.1 Conclusion

We provide additional understanding of the complexities of microbiota analysis by combining multiple approaches to address bacterial and viral communities in parallel. The relatively low complexity means that a single run on an Illumina MiSeq v3 cartridge is enough to study 60 dsDNA metaviromes. This approach has the potential to reduce costs associated with meta-community analyses in clinical settings. Viruses are involved in the early stages of microbial community development and act as an immune stimulus in the gut. Inclusion of viruses in the metataxonomic analyses offers additional resolution between patients including twin sets. This may be important in progressing understanding of mechanisms of dysbiosis associated with disease states, and of potential therapeutic relevance. The differences illustrated in fresh and frozen samples suggests that maximal understanding will come from fresh samples, posing additional study design issues for already hard to study populations. However, the lack of therapeutic progress that has so far arisen from purely bacterially focused studies suggests that this additional effort may prove pertinent.

Chapter 8. General Discussion

8.1 Discussion

The most direct and observable change in gut microbial communities is through infection and out growth of a pathogenic microbe. Current global consumption of antimicrobials, or other orally administered drugs which have weak antimicrobial potential, is at an all-time high, adding greater evolutionary pressures on microbial communities. This is of particular concern with the rise in antibiotic resistance (Ventola, 2015), as antibiotics are still the leading method of treatment in cases of severe/lethal pathogenic microbial gut infections. Viruses such as shiga-toxin encoding phage, increase the versatility of these gut pathogens (as shown in this study). Understanding the viral role in acquired resistance, virulence, and pathogenicity in bacteria is essential and should be a focus of continuing study.

This thesis aimed to study the role and impact of bacteriophages in the gut microbiome, with particular emphasis on temperate phage, while streamlining analytical techniques and investigating common sample storage methods. Chapter 3 illustrates how $\phi 24_B$ lysogeny of commensal gut associated microbe *E. coli* can immediately impact bacterial fitness. The first and foremost finding presented in this study was that lysogeny significantly increased growth rate during early growth phase, with greater difference seen with secondary infection (see section 3.2.1.1). This is supported by a study characterising lysogen infection by lambda phage (Lin et al., 1977). The chapter aimed to identify mechanisms behind this as well as the other possible traits that might have been immediately acquired from a lysogenic state. To achieve this, the study presented in chapter 3 determines the antimicrobial tolerances and metabolic changes of *E. coli* attained from $\phi 24_B$ conversion to lysogeny. It determines the potential dynamic interplay and symbiotic relationship between phage and bacteria. The study showed that lysogeny caused greater growth rate during early growth, which was further increased by secondary infection. However, a more rapid movement to death phase was also observed, which was exacerbated by secondary infection. When combined with metabolite data, it showed that during early growth the lysogen had a metabolic profile similar to that of mid-exponential growth phase. During mid-exponential growth phase the lysogen had a similar metabolite profile to

that of stationary phase. We hypothesise an enforced metabolic rate by the phage subversion of the cell, resulting in early cell exhaustion. Furthermore a known growth promoter 'biotin' was found to correlate to cell growth profiles. Biotin has been linked to enhanced growth in many publications (Snell & Mitchell, 1941a, Underkofler, Bantz et al., 1943a, Williams et al., 1940), but has never been shown to be upregulated in bacteria by phage conversion and alteration of bacterial respiration.

There are an array of antimicrobial resistance genes with mechanisms including efflux pumps, porins, target site alteration etc. (Munita & Arias, 2016). The study showed how infection of *E. coli* with $\phi 24_B$ without any known resistant gene cassette, was able to tolerate higher doses of antimicrobials. We propose that this could be due to a change in metabolic profile, potentially linked to an increase in fatty acids driven by the subversion of the biotin pathway. This study is not the first to investigate bacterial or lysogenic metabolite profiles. However this studies novel approach to metabolite profiling illustrates the importance of monitoring metabolite profiles under stress. Additionally this study has illucidated to one possible mechanism/factor behind the bacteriostatic nature of chloroxylenol.

This thesis also aimed to characterise an evolution/adaptation differences between the associated gut bacteria *E. coli* and its infecting phage $\phi 24_B$. The need to characterise bacteriophage impact on bacterial evolution, particularly in respect to antimicrobial resistance, can be exemplified by comparing publications on antibiotic resistance. To date, of all the publications investigating bacterial antibiotic resistance, only ~5% take into account bacteriophage (based on filter based Pubmed search). At $\sim 10^{31}$ phages with an estimated 10^{23} infections every second that can lead to a turnover rate of the entire phage population in just a few days (Suttle, 2007), it seems remis to not investigate their role in acquired antimicrobial resistance.

This therefore shows the importance of the study in chapter 4 and the novelty of the method to monitor the growth and lipid profile differences of the host and lysogen under increasing antimicrobial challenge. The novelty of this method was two-fold. Firstly this method was novel as it aimed to characterise the phage associated membrane lipid component differences between naïve host

and lysogen. Secondly it monitored the real-time evolution and adaptation difference between lysogen and naïve host. Fatty acid changes were previously observed in chapter 3. The key advantage of this method is identifying if fatty acid changes are structurally significant to the cell. This study found that under standard laboratory growth conditions the lysogen had an overall reduction in cell wall lipids. We hypothesise that this is perhaps a result of resources related to the biotin and fatty acid pathway being re-assigned to drive the increased lysogen growth seen in both chapter 3 and chapter 4. This study identified that the lysogen was capable of acquiring resistance against high concentrations of antimicrobials, something the non-infected *E. coli* was not able to do. We hypothesise this resistance is partially aided by, and associated to, a more sophisticated fatty acid cell wall structural response.

Following chapter 3, this study monitored the lysogen and naïve host growth under different environmental stresses, including temperature, oxygen limitation and bile salt addition. The study showed that under any condition the lysogen drives increased early growth, but the level of significance in growth deviated depending on condition. Infection by $\phi 24_B$ provided the greatest support in host growth under non-optimal growth temperatures (19°C). This would help bacteria survive in ambient environmental temperatures. Building from chapter 3's finding of increased growth factor (biotin) presence in the lysogen, this study identified an increase in expression of several biotin genes, from data mining Veses-Garcia et al transcriptomics (Veses-Garcia et al., 2015). From this data mining, it was also found that the lysogen influenced fatty acid gene expression, particularly regulator, repressor, and transporter genes. While this study identifies the lysogens manipulation of fatty acid plays a role in antimicrobial resistance, trends observed in the fatty acid profile suggests it is not the core mechanism of lysogen derived resistance. We hypothesise that the lysogen perhaps increases mutation rates in its host, making it more prone to swifter evolutionary adaptation. This hypothesis stems from the literature, as we have shown that the lysogen has increased growth, metabolic rate, and cellular stress, all of which have been previously identified as significant contributing factors in mutation (Chou et al., 2009, Fong et al., 2003, Nishimura et al., 2017, Poole, 2012, Tenaillon et al., 2004).

It should be noted that the fatty acid gene most effected by $\phi 24_B$ infection was the SCFA transporter (>2 fold difference), this is of particular interest, as SCFA are key players in nearly all associated impacts of the gut bacterial community. Metabolites function as the intermediate between cells, particularly in a pan-kingdom environment such as the gastrointestinal tract (Fischbach & Segre, 2016, Rooks & Garrett, 2016). As such, it was essential that metabolomic analysis was carried out in the opening studies (chapter 3 and 4). However, unlike comparing metabolic differences in species, genus, and family etc, we compared cells that were identical, were it not for viral infection. This created issues with alignment and compound comparison across conditions.

This therefore demonstrates the importance of chapter 5 and the novelty of its profiling of significant compounds across conditions. This chapter aimed to bridge the gap between peak calling/identification and graphical representation addressing the research question. This study aimed to simplify the handling of difficult datasets containing alignment issues, and was fundamental in chapter 3 metabolite analysis. This study also provides the means to analyse data as you go, by simplifying dataset pooling. Program construction and function is based in the linux environment. Linux is the lab bench for bioinformatics due to its user versatility at the command prompt and its reduced demand on the CPU. This study will add to the growing bioinformatic toolbox, and provided the means of swift metabolic analysis in chapter 3.

High-throughput systems, such as those in genomics and metabolomics, have resulted in a colossal output of complex data that requires powerful computational analysis. The average computational specs are too low to perform high level analysis at a functional speed, because of this, servers are often turned to for analysis (Choi & Chan, 2015, Langmead & Nellore, 2018, Ramirez, Ryan et al., 2016). The rate of technological advancement in data gathering has led to difficulties and gaps in data management tools. These gaps are bridged in groups or labs with bioinformatic access, but are difficult and labour intensive to those that do not, which is a common bottleneck for most laboratories (Edwards & Holt, 2013). The need for these skill sets has resulted in businesses profiting from making user friendly systems, such as; CLC workbench (CLC Bio), DNAnexus (DNAnexus, Inc., Mountain View, USA, <http://www.dnanexus.com>), Geneious (Kearse et al., 2012), DNASTAR

(DNASTAR inc.) and basespace (Illumina, inc., <https://illumina.com>). However, these systems are often bound by relatively rigid pipelines.

The importance of bioinformatic streamlining in modern biological sciences can be seen in the literature. In the field of metabolomics and genomics approximately 96% and 75% respectively, of all bioinformatic related papers ever published were published within the last decade alone. The number of results were calculated by searching for “(bioinformatics | metabolomics | metabolome) (<Terms>)” on PubMed (retrieved 20 March 2018) and “(bioinformatics | genome | genomic) (<Terms>)” on PubMed (retrieved 20 March 2018). It is clear that complex life science research, such as microbiome analysis, require fast bioinformatic tools that can streamline analysis of otherwise monumental datasets. This is likewise essential in biomes such as the gut.

This highlights the value of chapter 6 and the novelty of a complete genomic analysis toolbox as an open source installable program. Chapter 6 of this thesis looked to build a streamlined mass sample analysis tool of genomic data, for viral, bacterial and fungal community analysis, as well as whole genome analysis. Importantly this study aimed to provide the same extensive range of command line OSS as well as bridge any gaps between tools. There numerous OSS tools for varying types of genomic analysis most of which provide only a portion of the necessary steps, and function strictly from the command terminal. GGOSS is the first installable genomic analysis GUI program that includes a large variation of OSS tools covering the majority of genomic analysis techniques. Furthermore, it provides mass sample analysis, novel tools, custom setting defaults, and custom tool pipelining. Its user-friendly interface, streamlining methods, progression bars and estimated completion times, allow for easy genomic analysis. This chapter also aimed to simplify installation of all OSS integrated into the program, which is otherwise a difficult and time-consuming task. This chapter was essential in streamlining genomic analysis of the gut microbiome in chapter 7.

Gut microbes can have significant implications on both healthcare and the economy (Musich, MacLeod et al., 2016). The gastrointestinal tract is a complex microbiological environment harbouring a polymicrobial community of fungi, bacteria and viruses. These community structures

have been correlated to biomarkers, and precursors to an array of physical and mental health issues (see section 1.6.3). Necrotising enterocolitis (NEC) and late onset infection remain major causes of morbidity and mortality in preterm infants (Caplan, 2008). The microbiota of preterm neonates has been widely studied, and dysbiosis of the microbial community in the gastrointestinal tract appears key to disease development (Hosny, Cassir et al., 2017), yet how this occurs is poorly understood.

The impact of interactions between microbes in the gut biome on mammalian health is still poorly understood. The gut is a hostile, versatile and constantly changing environment, influenced by a diverse range of variables. Due to the global food market and oral consumption of medicine, the human gut is an extremely diverse melting pot of evolutionary driven pan-kingdom interactions (David, Maurice et al., 2014, Micha, Khatibzadeh et al., 2015, Van Boeckel, Gandra et al., 2014). The gut provides a niche where the pressure of the environment drives adaptation and diversity of the microbial community (Dominguez, Zarazaga et al., 2002, Ley, Peterson et al., 2006), where cross-talk not just between viral, bacterial and fungal life but also mammalian cells is essential. It is thus understandable that so much of mammalian health and lifestyle can be correlated to their gut microbial community (see chapter 1), and why genomic and metabolic research in this field is increasing.

In the literature, the assessment of gut microbial communities predominantly has a bacteria-centric methodological approach, where virome analysis consists of just 1% of gut community publications (based off the PubMed database). Furthermore, dysbiosis has often been linked to disease progression without identifying bacterial cause.

Chapter 7 of this thesis looked to compare the bacterial, viral, and fungal communities in a pan-kingdom analysis approach of the preterm neonate gut. This study observes the improved resolution in microbial dissimilarity between individual infants and twin pairs with the inclusion of lysogenic bacteriophage analysis. These findings were even more apparent with the free viral particle data. This chapter identified that lysogenic communities showed the strongest similarities to the bacterial communities reflecting bacterial viability. We observed that bacterial taxonomic richness

increased over time in all patients. A decrease in viral richness was seen in infants who remained healthy, with a greater patient cohort, we could hypothesise that the increased instability we see with disease states may in fact be driven by phage. This study demonstrates the potential importance of complementary viral community analysis in evaluating the role of microbiota stability and dysbiosis in polymicrobial disease states.

The impact of sample storage has been investigated in relation to bacterial community recovery and analysis (Baumgart & Carding, 2007, Blekhman, Tang et al., 2016, Lauber et al., 2010, Roesch, Casella et al., 2009). This chapter included the novel approach of monitoring the effects of faecal sample storage on viral communities. The study found that samples would ideally be used fresh rather than frozen stored. Though this poses increased difficulties in already challenging to study populations, the lack of therapeutic progress thus far from bacterio-centric studies suggests that this may be a necessary and pertinent step.

The studies presented in this thesis help to highlight the significant role a phage can have on a gut bacterium, the necessity of monitoring viral communities within a gut biome and fresh sample use, as well as the need for user-friendly tools to streamline these complex analyses. In a viral strain related to an enteric pathogenic virus, this study has shown increased growth and metabolic rate from primary and secondary lysogenic infection. Other studies have shown that pathogenic bacteria tend to maintain multiple prophages in the same chromosome (Hargreaves, Otieno et al., 2015, Hayashi et al., 2001, Winstanley et al., 2009). From our research this would suggest that pathogenic bacteria may have a greater potential of increased growth, metabolic rate, and possibly mutagenesis, due to their viral counterparts. A study has recently identified enteric pathogenic bacteria as key triggers/exacerbators of multiple sclerosis (Yadav, Boppana et al., 2017), however the methodology was entirely bacterio-centric, and could be associated to lysogeny of pathogenic bacteria.

The development of gut dysbiosis can lead to activation of the host immune and inflammation response, cause metabolic abnormalities, and disturbed intestinal homeostasis. As such, gut dysbiosis is implemented in many disease progressions (see section 1), however causative agents are commonly

only found to correlate at a phylum level, such as proteobacteria or bacteroidetes (Shin, Whon et al., 2015). NEC and LOS development is the imbalance between pro- and anti-inflammatory mechanisms in the gut leading to increased gut permeability and bacterial translocation or tissue damage. Several large-scale analyses, reviews and meta-analyses have been conducted into NEC (Pammi 2017, Warner 2016, Milani 2017), yet no universal bacterial pathogen or single repeatable ‘pattern’ of gut dysbiosis has been identified. Our study shows that viral analysis of the gut microbiome can improve analytical resolution, potentially allowing identification of more definitive causative agents.

SCFA have been linked to several disease progressions, described in section 1. One such example of fatty acid influence is conjugated linoleic acid, the bacterial metabolic by-product of dietary linoleic acid. Linoleic acid has many putative effects on host functions, that include anti-inflammatory and metabolic pathway regulation (Delzenne & Cani, 2011). Studies have shown that the reduction of bacterial anti-inflammatory SCFA metabolites in the gut result in inflammatory responses (see section 1). Inflammation in the gut is a strong selective pressure on microbes, resistance and sensitivities of bacteria to inflammatory fluctuations can provide methods of regulating the microbial environment. Our study identifies phage subversion of the fatty acid pathways, metabolically, transcriptionally and structurally. It is possible that in a lysogenic state, phage can regulate inflammatory responses in the gut to serve their purpose. For example, our study identifies lysogen driven fatty acid reduction during low stress conditions, and significant lysogen driven increases in fatty acids when stressed, this may play a role in gut survival, tempering inflammatory responses when they become detrimental to bacterial host survival.

This thesis has shown that temperate bacteriophage can have a role in gut bacterial evolution that’s not related to HGT or predator/prey systems, and may be involved in disease progression. Phage may add a layer of intricacy that can ultimately have an effect on the establishment and clinical outcomes of gut dysbiosis. This thesis stresses the importance of investigating the viral community, particularly in relation to temperate phage and lysogeny, and builds on current systems to improve the tools available to achieve this in research.

8.2 Further work

In order to fully elucidate if biotin regulation is a key mechanism behind prophage driven growth observed in the lysogen, it may be essential to carry out biotin gene knockouts in both the naïve host and the lysogen. Of particular interest is the BirA gene due its regulation of biotin and its interaction acetyl-CoA (see Figure 4.22 for illustration of hypothesis). This will not only identify if biotin is the growth regulating factor between infected and non-infected states, but also establish where the phage is influencing this gene structure.

Data mining of another group's transcriptomic data helped support theories toward phage regulation of the biotin and fatty acid pathways. However, transcriptomics were only carried out under standard conditions. To strengthen our findings and to elucidate more toward resistant mechanisms, it would be pertinent to conduct transcriptomic analysis under antimicrobial challenge. Furthermore, transcriptomic analysis through the acquired resistance study would identify the extent of phage subversion in resistance. While phage subversion may in fact be the sole source of acquired resistance mechanisms, it would be amiss to not investigate the potential mutational differences between naïve host and lysogen.

Running whole genome analysis on resistant lysogen cells at each antimicrobial increment, would allow us to identify if genetic mutation played a role in the antimicrobial resistance acquired by the lysogen. Furthermore, monitoring genomic change in both the naïve host and lysogen during incremental antimicrobial challenge would allow us to identify rate of mutation and confirm whether the lysogen is increasing this rate. This study would also help identify if mutational patterns emerged. Where evolution often takes the path of least resistance, it is likely that similar patterns in phage driven mutational change will occur, this could elucidate to novel mechanisms and targets of antimicrobial resistance.

Due to the array of data I've discovered on fatty acid subversion by the prophage (metabolomic, transcriptomic, and targeted structural fatty acids analysis), it would be prudent to run targeted metabolomics of SCFA associated to common disease progression within the gut. If found

significant, this would highlight the need to investigate viral communities in association to these disease progressions, as previously weak correlations may be explained by varying viral communities and their host ranges. This study would be strengthened by investigating an increased neonate cohort when examining disease progression and viral communities. By increasing our sample number we could accept or reject hypothesis toward increased bacterial instability in disease states being driven by bacteriophage.

CRACCD is not a standalone tool for metabolomics, and requires data preparation in other programs. Future work would like to include data alignment, compound identification, calculations of p values and CV%, and an automated system for prior pooling of an unlimited number of datasets. Thereby making it the first metabolomic program feely available and installable for complete metabolomic analysis. To provide improved genomic analysis support, GGOSS is intended to be bolstered to include many prominent OSS tools, as well as furthering unique GGOSS tools. Tools of primary interest of inclusion include RASTtk, trimmomatic, velvet, BASE, GAM_NGS, a selection of k-mer counting tools, REAPR, MEGAnnotator, Mauve, Gepard, and ViromeScan. New sections of GGOSS would be built to include assembly reconciliation and post annotation tools. In keeping with this study, specific tools would be created for the automated accumulation of separate microbial kingdom community analysis i.e. bacterial, viral and fungal, taking into account methods of abundance calculation. Other additions would include automated primer design for Illumina SBS platform with selectable 2 step or 1 step PCR based design, for more user specific community analysis.

Chapter 9. References

- Abdulkadir B, Nelson A, Skeath T, Marrs ECL, Perry JD, Cummings SP (2016) Stool bacterial load in preterm infants with necrotising enterocolitis. *Early Hum Dev* 95
- Agin TS, Zhu C, Johnson LA, Thate TE, Yang Z, Boedeker EC (2005) Protection against hemorrhagic colitis in an animal model by oral immunization with isogeneic rabbit enteropathogenic *Escherichia coli* attenuated by truncating intimin. *Infection and Immunity* 73: 6608-19
- Agol VI (1974) Towards the system of viruses. *Biosystems* 6: 113-132
- Ahlquist P (2006) Parallels among positive-strand RNA viruses, reverse-transcribing viruses and double-stranded RNA viruses. *Nature Reviews Microbiology* 4: 371-382
- Ahmad I, Mirza T, Qadeer K, Nazim U, Vaid FH (2013) Vitamin B6: deficiency diseases and methods of analysis. *Pak J Pharm Sci* 26: 1057-69
- Ahn J, Sinha R, Pei Z, Dominianni C, Wu J, Shi J, Goedert JJ, Hayes RB, Yang L (2013) Human gut microbiome and risk for colorectal cancer. *J Natl Cancer Inst* 105: 1907-11
- Aird D, Ross MG, Chen WS, Danielsson M, Fennell T, Russ C, Jaffe DB, Nusbaum C, Gnirke A (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol* 12: R18
- Aizenman E, Engelberg-Kulka H, Glaser G (1996) An *Escherichia coli* chromosomal "addiction module" regulated by guanosine [corrected] 3',5'-bispyrophosphate: a model for programmed bacterial cell death. *Proc Natl Acad Sci U S A* 93: 6059-63
- Al-Busafi SN, Suliman FEO, Al-Alawi ZR (2014) 8-hydroxyquinoline and its derivatives: Synthesis and applications. *ChemInform* 45
- Alang N, Kelly CR (2015) Weight gain after fecal microbiota transplantation. *Open Forum Infect Dis* 2: ofv004

Alcock J, Maley CC, Aktipis CA (2014) Is eating behavior manipulated by the gastrointestinal microbiota? Evolutionary pressures and potential mechanisms. *BioEssays : news and reviews in molecular, cellular and developmental biology* 36: 940-9

Alexander C, Rietschel ET (2001) Bacterial lipopolysaccharides and innate immunity. *J Endotoxin Res* 7: 167-202

Alhakami H, Mirebrahim H, Lonardi S (2017) A comparative evaluation of genome assembly reconciliation tools. *Genome Biol* 18: 93

Allin B, Long AM, Gupta A, Knight M, Lakhoo K, British Association of Paediatric Surgeons Congenital Anomalies Surveillance System Necrotising Enterocolitis C (2017) A UK wide cohort study describing management and outcomes for infants with surgical Necrotising Enterocolitis. *Scientific reports* 7: 41149

Allison HE (2007) Stx-phages: drivers and mediators of the evolution of STEC and STEC-like pathogens. *Future Microbiol* 2: 165-174

Allison HE, Sergeant MJ, James CE, Saunders JR, Smith DL, Sharp RJ, Marks TS, McCarthy AJ (2003) Immunity profiles of wild-type and recombinant shiga-like toxin-encoding bacteriophages and characterization of novel double lysogens. *Infection and immunity* 71: 3409-18

Altermann E, Lu J, McCulloch A (2017) GAMOLA2, a Comprehensive Software Package for the Annotation and Curation of Draft and Complete Microbial Genomes. *Frontiers in microbiology* 8: 346

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* 215: 403-10

Andersson DI, Hughes D (2011) Persistence of antibiotic resistance in bacterial populations. *FEMS Microbiol Rev* 35: 901-11

Angly FE, Willner D, Prieto-Davo A, Edwards RA, Schmieder R, Vega-Thurber R, Antonopoulos DA, Barott K, Cottrell MT, Desnues C, Dinsdale EA, Furlan M, Haynes M, Henn MR, Hu Y,

Kirchman DL, McDole T, McPherson JD, Meyer F, Miller RM et al. (2009) The GAAS metagenomic tool and its estimations of viral and microbial average genome size in four major biomes. *PLoS Comput Biol* 5: e1000593

Annous BA, Kozempel MF, Kurantz MJ (1999) Changes in membrane fatty acid composition of *Pediococcus* sp strain NRRL B-2354 in response to growth conditions and its effect on thermal resistance. *Applied and environmental microbiology* 65: 2857-2862

Arboleya S, Binetti A, Salazar N, Fernández N, Solís G, Hernández-Barranco A, Margolles A, de los Reyes-Gavilán CG, Gueimonde M (2012) Establishment and development of intestinal microbiota in preterm neonates. *FEMS Microbiology Ecology* 79: 763-772

Aron-Wisnewsky J, Dore J, Clement K (2012) The importance of the gut microbiota after bariatric surgery. *Nat Rev Gastroenterol Hepatol* 9: 590-8

Ascenzi JM (1995) *Handbook of Disinfectants and Antiseptics*. Taylor & Francis,

Baker GC, Smith JJ, Cowan DA (2003) Review and re-analysis of domain-specific 16S primers. *J Microbiol Meth* 55: 541-555

Bakhshinejad B, Ghiasvand S (2017) Bacteriophages in the human gut: Our fellow travelers throughout life and potential biomarkers of health or disease. *Virus Res* 240: 47-55

Bakken JS, Borody T, Brandt LJ, Brill JV, Demarco DC, Franzos MA, Kelly C, Khoruts A, Louie T, Martinelli LP, Moore TA, Russell G, Surawicz C (2011) Treating *Clostridium difficile* Infection With Fecal Microbiota Transplantation. *Clinical Gastroenterology and Hepatology* 9: 1044-1049

Baltimore D (1971) Expression of animal virus genomes. *Bacteriol Rev* 35: 235-41

Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, Pyshkin AV, Sirotkin AV, Vyahhi N, Tesler G, Alekseyev MA, Pevzner PA (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J Comput Biol* 19: 455-77

- Baothman OA, Zamzami MA, Taher I, Abubaker J, Abu-Farha M (2016) The role of Gut Microbiota in the development of obesity and Diabetes. *Lipids Health Dis* 15
- Barak I, Muchova K (2013) The Role of Lipid Domains in Bacterial Cell Processes. *Int J Mol Sci* 14: 4050-4065
- Barondess JJ, Beckwith J (1990) A bacterial virulence determinant encoded by lysogenic coliphage lambda. *Nature* 346: 871-4
- Barondess JJ, Beckwith J (1995) bor gene of phage lambda, involved in serum resistance, encodes a widely conserved outer membrane lipoprotein. *J Bacteriol* 177: 1247-53
- Baumgart DC, Carding SR (2007) Inflammatory bowel disease: cause and immunobiology. *Lancet* 369: 1627-40
- Beach RC, Menzies IS, Clayden GS, Scopes JW (1982) Gastrointestinal permeability changes in the preterm neonate. *Archives of Disease in Childhood* 57: 141-145
- Bercik P, Denou E, Collins J, Jackson W, Lu J, Jury J, Deng Y, Blennerhassett P, Macri J, McCoy KD, Verdu EF, Collins SM (2011) The intestinal microbiota affect central levels of brain-derived neurotrophic factor and behavior in mice. *Gastroenterology* 141: 599-609, 609 e1-3
- Berer K, Gerdes LA, Cekanaviciute E, Jia XM, Xiao L, Xia Z, Liu C, Klotz L, Stauffer U, Baranzini SE, Kumpfel T, Hohlfeld R, Krishnamoorthy G, Wekerle H (2017) Gut microbiota from multiple sclerosis patients enables spontaneous autoimmune encephalomyelitis in mice. *Proceedings of the National Academy of Sciences of the United States of America* 114: 10719-10724
- Berger AK, Mainou BA (2018) Interactions between Enteric Bacteria and Eukaryotic Viruses Impact the Outcome of Infection. *Viruses-Basel* 10
- Berger AK, Yi H, Kearns DB, Mainou BA (2017) Bacteria and bacterial envelope components enhance mammalian reovirus thermostability. *PLoS Pathog* 13: e1006768

Berlin K, Koren S, Chin CS, Drake JP, Landolin JM, Phillippy AM (2015) Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* 33: 623-30

Bhimma R, Rollins NC, Coovadia HM, Adhikari M (1997) Post-dysenteric hemolytic uremic syndrome in children during an epidemic of *Shigella* dysentery in Kwazulu/Natal. *Pediatric Nephrology* 11: 560-564

Bik EM, Bunschoten AE, Gouw RD, Mooi FR (1995) Genesis of the novel epidemic *Vibrio cholerae* O139 strain: evidence for horizontal transfer of genes involved in polysaccharide synthesis. *Embo J* 14: 209-16

Bik EM, Eckburg PB, Gill SR, Nelson KE, Purdom EA, Francois F, Perez-Perez G, Blaser MJ, Relman DA (2006) Molecular analysis of the bacterial microbiota in the human stomach. *Proceedings of the National Academy of Sciences of the United States of America* 103: 732-7

Bjarnsholt T, Jensen PO, Burmolle M, Hentzer M, Haagensen JAJ, Hougen HP, Calum H, Madsen KG, Moser C, Molin S, Hoiby N, Givskov M (2005) *Pseudomonas aeruginosa* tolerance to tobramycin, hydrogen peroxide and polymorphonuclear leukocytes is quorum-sensing dependent. *Microbiol-Sgm* 151: 373-383

Blair JM, Webber MA, Baylay AJ, Ogbolu DO, Piddock LJ (2015) Molecular mechanisms of antibiotic resistance. *Nat Rev Microbiol* 13: 42-51

Blekhman R, Tang K, Archie EA, Barreiro LB, Johnson ZP, Wilson ME, Kohn J, Yuan ML, Gesquiere L, Grieneisen LE, Tung J (2016) Common methods for fecal sample storage in field studies yield consistent signatures of individual identity in microbiome sequencing data. *Scientific reports* 6: 31519

Blower TR, Short FL, Rao F, Mizuguchi K, Pei XY, Fineran PC, Luisi BF, Salmond GPC (2012) Identification and classification of bacterial Type III toxin-antitoxin systems encoded in chromosomal and plasmid genomes. *Nucleic Acids Research* 40: 6158-6173

Bohannan BJM, Lenski RE (2000) Linking genetic change to community evolution: insights from studies of bacteria and bacteriophage. *Ecol Lett* 3: 362-377

Bohn E, Bechtold O, Zahir N, Frick JS, Reimann J, Jilge B, Autenrieth IB (2006) Host gene expression in the colon of gnotobiotic interleukin-2-deficient mice colonized with commensal colitogenic or noncolitogenic bacterial strains: common patterns and bacteria strain specific signatures. *Inflamm Bowel Dis* 12: 853-62

Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30: 2114-20

Bollback JP, Huelsenbeck JP (2001) Phylogeny, genome evolution, and host specificity of single-stranded RNA bacteriophage (family Leviviridae). *J Mol Evol* 52: 117-128

Bonanno L, Petit MA, Loukiadis E, Michel V, Auvray F (2016) Heterogeneity in Induction Level, Infection Ability, and Morphology of Shiga Toxin-Encoding Phages (Stx Phages) from Dairy and Human Shiga Toxin-Producing *Escherichia coli* O26:H11 Isolates. *Applied and environmental microbiology* 82: 2177-86

Bondy-Denomy J, Qian J, Westra ER, Buckling A, Guttman DS, Davidson AR, Maxwell KL (2016) Prophages mediate defense against phage infection through diverse mechanisms. *Isme J* 10: 2854-2866

Boroni Moreira AP, Fiche Salles Teixeira T, do CGPM, de Cassia Goncalves Alfenas R (2012) Gut microbiota and the development of obesity. *Nutr Hosp* 27: 1408-14

Bosi E, Donati B, Galardini M, Brunetti S, Sagot MF, Lio P, Crescenzi P, Fani R, Fondi M (2015) MeDuSa: a multi-draft based scaffolder. *Bioinformatics* 31: 2443-51

Botticelli A, Zizzari I, Mazzuca F, Ascierio PA, Putignani L, Marchetti L, Napoletano C, Nuti M, Marchetti P (2017) Cross-talk between microbiota and immune fitness to steer and control response to anti PD-1/PDL-1 treatment. *Oncotarget* 8: 8890-8899

Brantl S (2012) Bacterial type I toxin-antitoxin systems. *Rna Biol* 9: 1488-1490

Breitbart M, Haynes M, Kelley S, Angly F, Edwards RA, Felts B, Mahaffy JM, Mueller J, Nulton J, Rayhawk S, Rodriguez-Brito B, Salamon P, Rohwer F (2008) Viral diversity and dynamics in an infant gut. *Res Microbiol* 159: 367-73

Brettin T, Davis JJ, Disz T, Edwards RA, Gerdes S, Olsen GJ, Olson R, Overbeek R, Parrello B, Pusch GD, Shukla M, Thomason JA, 3rd, Stevens R, Vonstein V, Wattam AR, Xia F (2015) RASTtk: a modular and extensible implementation of the RAST algorithm for building custom annotation pipelines and annotating batches of genomes. *Scientific reports* 5: 8365

Brown S, Santa Maria JP, Jr., Walker S (2013) Wall teichoic acids of gram-positive bacteria. *Annu Rev Microbiol* 67: 313-36

Bruce-Keller AJ, Salbaum JM, Luo M, Blanchard Et, Taylor CM, Welsh DA, Berthoud HR (2015) Obese-type gut microbiota induce neurobehavioral changes in the absence of obesity. *Biol Psychiatry* 77: 607-15

Brudno M, Do CB, Cooper GM, Kim MF, Davydov E, Program NCS, Green ED, Sidow A, Batzoglou S (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res* 13: 721-31

Brunham, R.C., Shen, C., Gill, S.R., Heidelberg, J.F., White, O., Hickey, E.K., Peterson, J., Utterback, T., Berry, K., Bass, S., Linher, K., Weidman, J., Khouri, H., Craven, B., Bowman, C., Dodson, R., Gwinn, M., Nelson, W., DeBoy, R., Kolonay, J., McClarty, G., Salzberg, S.L., Eisen, J., Fraser, C.M. (2000). Genome sequences of *Chlamydia trachomatis* MoPn and *Chlamydia pneumoniae* AR39. *Nucleic Acids Res.* 28: 1397-1406

Brussow H, Canchaya C, Hardt WD (2004) Phages and the evolution of bacterial pathogens: From genomic rearrangements to lysogenic conversion. *Microbiol Mol Biol R* 68: 560-+

Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. *Nature Methods* 12: 59-60

Bueter M, Abegg K, Seyfried F, Lutz TA, le Roux CW (2012) Roux-en-Y gastric bypass operation in rats. *J Vis Exp*: e3940

Buffie CG, Bucci V, Stein RR, McKenney PT, Ling L, Gobourne A, No D, Liu H, Kinnebrew M, Viale A, Littmann E, van den Brink MR, Jenq RR, Taur Y, Sander C, Cross JR, Toussaint NC, Xavier JB, Pamer EG (2015) Precision microbiome reconstitution restores bile acid mediated resistance to *Clostridium difficile*. *Nature* 517: 205-8

Busby B, Kristensen DM, Koonin EV (2013) Contribution of phage-derived genomic islands to the virulence of facultative bacterial pathogens. *Environ Microbiol* 15: 307-312

Byrdwell WC (2001) Atmospheric pressure chemical ionization mass spectrometry for analysis of lipids. *Lipids* 36: 327-346

Cadwell K (2015) Expanding the role of the virome: commensalism in the gut. *Journal of virology* 89: 1951-3

Canchaya C, Fournous G, Brussow H (2004) The impact of prophages on bacterial chromosomes. *Mol Microbiol* 53: 9-18

Cani PD, Bibiloni R, Knauf C, Waget A, Neyrinck AM, Delzenne NM, Burcelin R (2008) Changes in gut microbiota control metabolic endotoxemia-induced inflammation in high-fat diet-induced obesity and diabetes in mice. *Diabetes* 57: 1470-81

Cani PD, Delzenne NM (2010) Involvement of the gut microbiota in the development of low grade inflammation associated with obesity : focus on this neglected partner. *Acta Gastro-Ent Belg* 73: 267-269

Cani PD, Osto M, Geurts L, Everard A (2012) Involvement of gut microbiota in the development of low-grade inflammation and type 2 diabetes associated with obesity. *Gut Microbes* 3: 279-88

Cantarel BL, Korf I, Robb SMC, Parra G, Ross E, Moore B, Holt C, Alvarado AS, Yandell M (2008) MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Res* 18: 188-196

Caplan MS (2008) Introduction. *Seminars in Perinatology* 32: 69

Caplan MS, Jilling T (2001) New concepts in necrotizing enterocolitis. *Curr Opin Pediatr* 13: 111-115

Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI, Huttley GA, Kelley ST, Knights D, Koenig JE, Ley RE, Lozupone CA, McDonald D, Muegge BD, Pirrung M, Reeder J et al. (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat Methods* 7: 335-6

Carlet J (2012) The gut is the epicentre of antibiotic resistance. *Antimicrob Resist In* 1

Carver T, Harris SR, Berriman M, Parkhill J, McQuillan JA (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics* 28: 464-9

Casas V, Magbanua J, Sobrepena G, Kelley ST, Maloy SR (2010) Reservoir of bacterial exotoxin genes in the environment. *Int J Microbiol* 2010: 754368

Casas V, Sobrepena G, Rodriguez-Mueller B, Ahtye J, Maloy SR (2011) Bacteriophage-encoded shiga toxin gene in atypical bacterial host. *Gut Pathog* 3: 10

Casjens SR, Hendrix RW (2015) Bacteriophage lambda: Early pioneer and still relevant. *Virology* 479: 310-330

Catalano CE, Cue D, Feiss M (1995) Virus-DNA Packaging - the Strategy Used by Phage-Lambda. *Mol Microbiol* 16: 1075-1086

Cenit MC, Sanz Y, Codoner-Franch P (2017) Influence of gut microbiota on neuropsychiatric disorders. *World J Gastroentero* 23: 5486-5498

Chafee ME, Zecher CN, Gourley ML, Schmidt VT, Chen JH, Bordenstein SR, Clark ME (2011) Decoupling of host-symbiont-phage coadaptations following transfer between insect species. *Genetics* 187: 203-15

Chaisson MJP, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F, Antonacci F, Surti U, Sandstrom R, Boitano M, Landolin JM, Stamatoyannopoulos JA, Hunkapiller MW, Korlach J, Eichler EE (2015) Resolving the complexity of the human genome using single-molecule sequencing. *Nature* 517: 608-U163

Chambers ES, Viardot A, Psichas A, Morrison DJ, Murphy KG, Zac-Varghese SE, MacDougall K, Preston T, Tedford C, Finlayson GS, Blundell JE, Bell JD, Thomas EL, Mt-Isa S, Ashby D, Gibson GR, Kolida S, Dhillo WS, Bloom SR, Morley W et al. (2015) Effects of targeted delivery of propionate to the human colon on appetite regulation, body weight maintenance and adiposity in overweight adults. *Gut* 64: 1744-54

Chandler M, de la Cruz F, Dyda F, Hickman AB, Moncalian G, Ton-Hoang B (2013) Breaking and joining single-stranded DNA: the HUH endonuclease superfamily. *Nature Reviews Microbiology* 11: 525-538

Chany C, Moscovici O, Lebon P, Rousset S (1982) Association of coronavirus infection with neonatal necrotizing enterocolitis. *Pediatrics* 69: 209-14

Cheeke PR (1995) Endogenous Toxins and Mycotoxins in Forage Grasses and Their Effects on Livestock. *J Anim Sci* 73: 909-918

Chen KT, Chen CJ, Shen HT, Liu CL, Huang SH, Lu CL (2016) Multi-CAR: a tool of contig scaffolding using multiple references. *Bmc Bioinformatics* 17

Chen YC, Liu T, Yu CH, Chiang TY, Hwang CC (2013) Effects of GC bias in next-generation-sequencing data on de novo genome assembly. *Plos One* 8: e62856

Cherny I, Gazit E (2004) The YefM antitoxin defines a family of natively unfolded proteins - Implications as a novel antibacterial target. *J Biol Chem* 279: 8252-8261

Chevallereau A, Blasdel BG, De Smet J, Monot M, Zimmermann M, Kogadeeva M, Sauer U, Jorth P, Whiteley M, Debarbieux L, Lavigne R (2016) Next-Generation "-omics" Approaches Reveal a Massive Alteration of Host RNA Metabolism during Bacteriophage Infection of *Pseudomonas aeruginosa*. *PLoS Genet* 12: e1006134

Choi Y, Chan AP (2015) PROVEAN web server: a tool to predict the functional effect of amino acid substitutions and indels. *Bioinformatics* 31: 2745-2747

Chopin MC, Chopin A, Bidnenko E (2005) Phage abortive infection in lactococci: variations on a theme. *Curr Opin Microbiol* 8: 473-9

Chou HH, Berthet J, Marx CJ (2009) Fast growth increases the selective advantage of a mutation arising recurrently during evolution under metal limitation. *PLoS genetics* 5: e1000652

Christensen SK, Maenhaut-Michel G, Mine N, Gottesman S, Gerdes K, Van Melderen L (2004) Overproduction of the Lon protease triggers inhibition of translation in *Escherichia coli*: involvement of the yefM-yoeB toxin-antitoxin system. *Mol Microbiol* 51: 1705-17

Christensen SK, Mikkelsen M, Pedersen K, Gerdes K (2001) ReIE, a global inhibitor of translation, is activated during nutritional stress. *P Natl Acad Sci USA* 98: 14328-14333

Cimolai N, Morrison BJ, Carter JE (1992) Risk-Factors for the Central-Nervous-System Manifestations of Gastroenteritis-Associated Hemolytic-Uremic Syndrome. *Pediatrics* 90: 616-621

Claesson MJ, Cusack S, O'Sullivan O, Greene-Diniz R, de Weerd H, Flannery E, Marchesi JR, Falush D, Dinan T, Fitzgerald G, Stanton C, van Sinderen D, O'Connor M, Harnedy N, O'Connor K, Henry C, O'Mahony D, Fitzgerald AP, Shanahan F, Twomey C et al. (2011) Composition, variability, and temporal stability of the intestinal microbiota of the elderly. *Proceedings of the National Academy of Sciences of the United States of America* 108 Suppl 1: 4586-91

- Clarke J, Wu HC, Jayasinghe L, Patel A, Reid S, Bayley H (2009) Continuous base identification for single-molecule nanopore DNA sequencing. *Nat Nanotechnol* 4: 265-270
- Clemente JC, Ursell LK, Parfrey LW, Knight R (2012) The impact of the gut microbiota on human health: an integrative view. *Cell* 148: 1258-70
- Collado MC, Rautava S, Aakko J, Isolauri E, Salminen S (2016) Human gut colonisation may be initiated in utero by distinct microbial communities in the placenta and amniotic fluid. *Scientific reports* 6
- Collins FS, Morgan M, Patrinos A (2003) The Human Genome Project: Lessons from Large-Scale Biology. *Science* 300: 286-290
- Collins JJ, Alder CR, Fernandez-Pol JA, Court D, Johnson GS (1979) Transient growth inhibition of *Escherichia coli* K-12 by ion chelators: "in vivo" inhibition of ribonucleic acid synthesis. *J Bacteriol* 138: 923-32
- Colomer-Lluch M, Jofre J, Muniesa M (2011) Antibiotic Resistance Genes in the Bacteriophage DNA Fraction of Environmental Samples. *Plos One* 6
- Colomer-Lluch M, Jofre J, Muniesa M (2014) Quinolone resistance genes (*qnrA* and *qnrS*) in bacteriophage particles from wastewater samples and the effect of inducing agents on packaged antibiotic resistance genes. *J Antimicrob Chemother* 69: 1265-74
- Colon M, Chakraborty D, Pevzner Y, Koudelka G (2016) Mechanisms that Determine the Differential Stability of *Stx*⁺ and *Stx*⁻ Lysogens. *Toxins* 8: 96
- Colpitts SL, Kasper LH (2017) Influence of the Gut Microbiome on Autoimmunity in the Central Nervous System. *J Immunol* 198: 596-604
- Colson P, Fancello L, Gimenez G, Armougom F, Desnues C, Fournous G, Yoosuf N, Million M, La Scola B, Raoult D (2013) Evidence of the megavirome in humans. *J Clin Virol* 57: 191-200

Cong X, Xu W, Janton S, Henderson WA, Matson A, McGrath JM, Maas K, Graf J (2016) Gut Microbiome Developmental Patterns in Early Life of Preterm Infants: Impacts of Feeding and Gender. *Plos One* 11: e0152751

Conterno L, Fava F, Viola R, Tuohy KM (2011) Obesity and the gut microbiota: does up-regulating colonic fermentation protect against obesity and metabolic disease? *Genes Nutr* 6: 241-260

Coordinators NR (2016) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 44: D7-19

Coren JS, Pierce JC, Sternberg N (1995) Headful Packaging Revisited - the Packaging of More Than One DNA Molecule into a Bacteriophage-P1 Head. *Journal of Molecular Biology* 249: 176-184

Cox AJ, West NP, Cripps AW (2015) Obesity, inflammation, and the gut microbiota. *Lancet Diabetes Endocrinol* 3: 207-15

Cox LM, Blaser MJ (2013) Pathways in microbe-induced obesity. *Cell Metab* 17: 883-94

Criscuolo A, Brisse S (2013) AlienTrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. *Genomics* 102: 500-6

Crusoe MR, Alameldin HF, Awad S, Boucher E, Caldwell A, Cartwright R, Charbonneau A, Constantinides B, Edverson G, Fay S, Fenton J, Fenzl T, Fish J, Garcia-Gutierrez L, Garland P, Gluck J, González I, Guermond S, Guo J, Gupta A et al. (2015) The khmer software package: enabling efficient nucleotide sequence analysis. *F1000Research* 4: 900

Cruz JR, Caceres P, Cano F, Flores J, Bartlett A, Torun B (1990) Adenovirus types 40 and 41 and rotaviruses associated with diarrhea in children from Guatemala. *J Clin Microbiol* 28: 1780-4

Cue D, Feiss M (1998) Termination of packaging of the bacteriophage lambda chromosome: cosQ is required for nicking the bottom strand of cosN. *Journal of Molecular Biology* 280: 11-29

Cue D, Feiss M (2001) Bacteriophage lambda DNA packaging: DNA site requirements for termination and processivity. *J Mol Biol* 311: 233-40

Cui L, Murchland I, Dodd IB, Shearwin KE (2013) Bacteriophage lambda repressor mediates the formation of a complex enhancer-like structure. *Transcription* 4: 201-5

Dahm R (2008) Discovering DNA: Friedrich Miescher and the early years of nucleic acid research. *Hum Genet* 122: 565-81

Dakshinamurti K (2005) Biotin - a regulator of gene expression. *J Nutr Biochem* 16: 419-423

Dalgliesh CE, Horning EC, Horning MG, Knox KL, Yarger K (1966) A gas-liquid-chromatographic procedure for separating a wide range of metabolites occurring in urine or tissue extracts. *Biochem J* 101: 792-810

Danese S, Malesci A, Vetrano S (2011) Colitis-associated cancer: the dark side of inflammatory bowel disease. *Gut* 60: 1609-10

David LA, Maurice CF, Carmody RN, Gootenberg DB, Button JE, Wolfe BE, Ling AV, Devlin AS, Varma Y, Fischbach MA, Biddinger SB, Dutton RJ, Turnbaugh PJ (2014) Diet rapidly and reproducibly alters the human gut microbiome. *Nature* 505: 559-63

Davis BM, Moyer KE, Boyd EF, Waldor MK (2000) CTX prophages in classical biotype *Vibrio cholerae*: Functional phage genes but dysfunctional phage genomes. *J Bacteriol* 182: 6992-6998

Davison S, Couture-Tosi E, Candela T, Mock M, Fouet A (2005) Identification of the *Bacillus anthracis* (gamma) phage receptor. *J Bacteriol* 187: 6742-9

Davison WC (1922) The bacteriolysant therapy of bacillary dysentery in children: Therapeutic application of bacteriolysants; d'herelle's phenomenon. *American Journal of Diseases of Children* 23: 531-534

De E, Basle A, Jaquinod M, Saint N, Mallea M, Molle G, Pages JM (2001) A new mechanism of antibiotic resistance in *Enterobacteriaceae* induced by a structural modification of the major porin. *Mol Microbiol* 41: 189-98

de la Cochetiere MF, Piloquet H, des Robert C, Darmaun D, Galmiche JP, Roze JC (2004) Early intestinal bacterial colonization and necrotizing enterocolitis in premature infants: the putative role of Clostridium. *Pediatric research* 56: 366-70

De Smet J, Zimmermann M, Kogadeeva M, Ceysens PJ, Vermaelen W, Blasdel B, Bin Jang H, Sauer U, Lavigne R (2016) High coverage metabolomics analysis reveals phage-specific alterations to *Pseudomonas aeruginosa* physiology during infection. *ISME J* 10: 1823-35

De Vadder F, Kovatcheva-Datchary P, Goncalves D, Vinera J, Zitoun C, Duchamp A, Backhed F, Mithieux G (2014) Microbiota-generated metabolites promote metabolic benefits via gut-brain neural circuits. *Cell* 156: 84-96

Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL (1999) Alignment of whole genomes. *Nucleic Acids Res* 27: 2369-76

Delcher AL, Phillippy A, Carlton J, Salzberg SL (2002) Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res* 30: 2478-83

Delzenne NM, Cani PD (2011) Interaction Between Obesity and the Gut Microbiota: Relevance in Nutrition. *Annual Review of Nutrition*, Vol 31 31: 15-31

Delzenne NM, Neyrinck AM, Backhed F, Cani PD (2011) Targeting gut microbiota in obesity: effects of prebiotics and probiotics. *Nat Rev Endocrinol* 7: 639-46

DeMarini DM, Lawrence BK (1992) Prophage induction by DNA topoisomerase II poisons and reactive-oxygen species: role of DNA breaks. *Mutat Res* 267: 1-17

den Besten G, Bleeker A, Gerding A, van Eunen K, Havinga R, van Dijk TH, Oosterveer MH, Jonker JW, Groen AK, Reijngoud DJ, Bakker BM (2015) Short-Chain Fatty Acids Protect Against High-Fat Diet-Induced Obesity via a PPARgamma-Dependent Switch From Lipogenesis to Fat Oxidation. *Diabetes* 64: 2398-408

Deorowicz S, Debudaj-Grabysz A, Grabowski S (2013) Disk-based k-mer counting on a PC. *BMC Bioinformatics* 14: 160

Deorowicz S, Kokot M, Grabowski S, Debudaj-Grabysz A (2015) KMC 2: fast and resource-frugal k-mer counting. *Bioinformatics* 31: 1569-76

Derrien M, Vlieg JETV (2015) Fate, activity, and impact of ingested bacteria within the human gut microbiota. *Trends Microbiol* 23: 354-366

Deshpande G, Rao S, Patole S, Bulsara M (2010) Updated meta-analysis of probiotics for preventing necrotizing enterocolitis in preterm neonates. *Pediatrics* 125: 921-30

Desnues B, Cuny C, Gregori G, Dukan S, Aguilaniu H, Nystrom T (2003) Differential oxidative damage and expression of stress defence regulons in culturable and non-culturable *Escherichia coli* cells. *EMBO Rep* 4: 400-4

Dethlefsen L, Relman DA (2011) Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. *Proc Natl Acad Sci U S A* 108 Suppl 1: 4554-61

DeVuyst L, Callewaert R, Crabbe K (1996) Primary metabolite kinetics of bacteriocin biosynthesis by *Lactobacillus amylovorus* and evidence for stimulation of bacteriocin production under unfavourable growth conditions. *Microbiol-Uk* 142: 817-827

d'Hérelle F. (1917). Sur un microbe invisible antagoniste des bacilles dysentériques. *C R Acad Sci Ser D*. 165: 373–375.

Di Pasqua R, Hoskins N, Betts G, Mauriello G (2006) Changes in membrane fatty acids composition of microbial cells induced by addition of thymol, carvacrol, limonene, cinnamaldehyde, and eugenol in the growing media. *J Agr Food Chem* 54: 2745-2749

Diago-Navarro E, Hernandez-Arriaga AM, Kubik S, Konieczny I, Diaz-Orejas R (2013) Cleavage of the antitoxin of the parD toxin-antitoxin system is determined by the ClpAP protease and is modulated by the relative ratio of the toxin and the antitoxin. *Plasmid* 70: 78-85

Diemer GS, Stedman KM (2012) A novel virus genome discovered in an extreme environment suggests recombination between unrelated groups of RNA and DNA viruses. *Biol Direct* 7

Dittmar E, Beyer P, Fischer D, Schafer V, Schoepe H, Bauer K, Schlosser R (2008) Necrotizing enterocolitis of the neonate with *Clostridium perfringens*: diagnosis, clinical course, and role of alpha toxin. *Eur J Pediatr* 167: 891-5

Dohm JC, Lottaz C, Borodina T, Himmelbauer H (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res* 36: e105

Doll K, Riepl H, Eichhorn W, Dirksen G (1999) [Bile acid concentrations in serum, bile and feces of healthy calves and calves with diarrhea]. *Dtsch Tierarztl Wochenschr* 106: 35-40

Domingues S, Chopin A, Ehrlich SD, Chopin MC (2004) The lactococcal abortive phage infection system AbiP prevents both phage DNA replication and temporal transcription switch. *J Bacteriol* 186: 713-721

Dominguez E, Zarazaga M, Saenz Y, Brinas L, Torres C (2002) Mechanisms of antibiotic resistance in *Escherichia coli* isolates obtained from healthy children in Spain. *Microb Drug Resist* 8: 321-327

Dopkins N, Nagarkatti PS, Nagarkatti M (2018) The Role of Gut Microbiome and Associated Metabolome in the Regulation of Neuroinflammation in Multiple Sclerosis and Its Implications in Attenuating Chronic Inflammation in Other Inflammatory and Autoimmune Disorders. *Immunology*

Doyle MP, Erickson MC (2006) Reducing the carriage of foodborne pathogens in livestock and poultry. *Poultry Sci* 85: 960-973

Driffield K, Miller K, Bostock JM, O'Neill AJ, Chopra I (2008) Increased mutability of *Pseudomonas aeruginosa* in biofilms. *J Antimicrob Chemoth* 61: 1053-1056

Duda-Chodak A, Tarko T, Satora P, Sroka P (2015) Interaction of dietary compounds, especially polyphenols, with the intestinal microbiota: a review. *Eur J Nutr* 54: 325-341

Duerkop BA, Hooper LV (2013) Resident viruses and their interactions with the immune system. *Nature Immunology* 14: 654-659

Duffy C, Feiss M (2002) The large subunit of bacteriophage lambda's terminase plays a role in DNA translocation and packaging termination. *J Mol Biol* 316: 547-61

Dy RL, Przybilski R, Semeijn K, Salmond GPC, Fineran PC (2014) A widespread bacteriophage abortive infection system functions through a Type IV toxin-antitoxin mechanism. *Nucleic Acids Research* 42: 4590-4605

Echols H, Green L (1971) Establishment and maintenance of repression by bacteriophage lambda: the role of the cI, cII, and c3 proteins. *Proceedings of the National Academy of Sciences of the United States of America* 68: 2190-4

Edgar R, Rokney A, Feeney M, Semsey S, Kessel M, Goldberg MB, Adhya S, Oppenheim AB (2008) Bacteriophage infection is targeted to cellular poles. *Mol Microbiol* 68: 1107-1116

Edlin G, Lin L, Bitner R (1977) Reproductive fitness of P1, P2, and Mu lysogens of *Escherichia coli*. *J Virol* 21: 560-4

Edlin G, Lin L, Kudrna R (1975a) Lambda lysogens of *E. coli* reproduce more rapidly than non-lysogens. *Nature* 255: 735-7

Edlin G, Lin LEO, Kudrna R (1975b) λ Lysogens of *E. coli* reproduce more rapidly than non-lysogens. *Nature* 255: 735

Edwards DJ, Holt KE (2013) Beginner's guide to comparative bacterial genome analysis using next-generation sequence data. *Microbial Informatics and Experimentation* 3: 2-2

Eklom R, Wolf JB (2014) A field guide to whole-genome sequencing, assembly and annotation. *Evol Appl* 7: 1026-42

El Omari K, Sutton G, Ravantti JJ, Zhang HW, Walter TS, Grimes JM, Bamford DH, Stuart DI, Mancini EJ (2013) Plate Tectonics of Virus Shell Assembly and Reorganization in Phage Phi 8, a Distant Relative of Mammalian Reoviruses. *Structure* 21: 1384-1395

Elinav E, Nowarski R, Thaiss CA, Hu B, Jin CC, Flavell RA (2013) Inflammation-induced cancer: crosstalk between tumours, immune cells and microorganisms. *Nature Reviews Cancer* 13: 759-771

Enault F, Briet A, Bouteille L, Roux S, Sullivan MB, Petit MA (2016) Phages rarely encode antibiotic resistance genes: a cautionary tale for virome analyses. *ISME J*

Eppinger M, Mammel MK, Leclerc JE, Ravel J, Cebula TA (2011) Genomic anatomy of *Escherichia coli* O157:H7 outbreaks. *P Natl Acad Sci USA* 108: 20142-7

Erridge C, Bennett-Guerrero E, Poxton IR (2002) Structure and function of lipopolysaccharides. *Microbes Infect* 4: 837-851

Evrensel A, Ceylan ME (2016) Fecal Microbiota Transplantation and Its Usage in Neuropsychiatric Disorders. *Clin Psychopharmacol Neurosci* 14: 231-7

Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* 8: 186-194

Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res* 8: 175-185

Feng Q, Liang S, Jia H, Stadlmayr A, Tang L, Lan Z, Zhang D, Xia H, Xu X, Jie Z, Su L, Li X, Li X, Li J, Xiao L, Huber-Schonauer U, Niederseer D, Xu X, Al-Aama JY, Yang H et al. (2015) Gut microbiome development along the colorectal adenoma-carcinoma sequence. *Nature communications* 6: 6528

Ferenci T, Schwentorat M, Ullrich S, Vilmart J (1980) Lambda-Receptor in the Outer-Membrane of *Escherichia-Coli* as a Binding-Protein for Maltodextrins and Starch Polysaccharides. *J Bacteriol* 142: 521-526

Fernandez De Henestrosa AR, Ogi T, Aoyagi S, Chafin D, Hayes JJ, Ohmori H, Woodgate R (2000) Identification of additional genes belonging to the LexA regulon in *Escherichia coli*. *Mol Microbiol* 35: 1560-72

Fetissov SO, Hallman J, Orelund L, Af Klinteberg B, Grenback E, Hulting AL, Hokfelt T (2002) Autoantibodies against alpha -MSH, ACTH, and LHRH in anorexia and bulimia nervosa patients. *Proceedings of the National Academy of Sciences of the United States of America* 99: 17155-60

Fetissov SO, Harro J, Jaanisk M, Jarv A, Podar I, Allik J, Nilsson I, Sakthivel P, Lefvert AK, Hokfelt T (2005) Autoantibodies against neuropeptides are associated with psychological traits in eating disorders. *Proceedings of the National Academy of Sciences of the United States of America* 102: 14865-70

Filee J (2013) Route of NCLDV evolution: the genomic accordion. *Curr Opin Virol* 3: 595-599

Fineran PC, Blower TR, Foulds IJ, Humphreys DP, Lilley KS, Salmond GPC (2009) The phage abortive infection system, ToxIN, functions as a protein-RNA toxin-antitoxin pair. *P Natl Acad Sci USA* 106: 894-899

Firmesse O, Mogenet A, Bresson JL, Corthier G, Furet JP (2008) *Lactobacillus rhamnosus* R11 consumed in a food supplement survived human digestive transit without modifying microbiota equilibrium as assessed by real-time polymerase chain reaction. *J Mol Microb Biotech* 14: 90-99

Fischbach MA, Segre JA (2016) Signaling in Host-Associated Microbial Communities. *Cell* 164: 1288-1300

Flint HJ (2011) Obesity and the gut microbiota. *J Clin Gastroenterol* 45 Suppl: S128-32

Flint HJ, Scott KP, Duncan SH, Louis P, Forano E (2012a) Microbial degradation of complex carbohydrates in the gut. *Gut Microbes* 3: 289-306

Flint HJ, Scott KP, Louis P, Duncan SH (2012b) The role of the gut microbiota in nutrition and health. *Nat Rev Gastroenterol Hepatol* 9: 577-89

Fogg PC, Allison HE, Saunders JR, McCarthy AJ (2010) Bacteriophage lambda: a paradigm revisited. *Journal of virology* 84: 6876-9

Fogg PC, Gossage SM, Smith DL, Saunders JR, McCarthy AJ, Allison HE (2007) Identification of multiple integration sites for Stx-phage Phi24B in the *Escherichia coli* genome, description of a novel integrase and evidence for a functional anti-repressor. *Microbiology* 153: 4098-110

Fogg PC, Saunders JR, McCarthy AJ, Allison HE (2012) Cumulative effect of prophage burden on Shiga toxin production in *Escherichia coli*. *Microbiology* 158: 488-97

Fogg PCM, Rigden DJ, Saunders JR, McCarthy AJ, Allison HE (2011) Characterization of the relationship between integrase, excisionase and antirepressor activities associated with a superinfecting Shiga toxin encoding bacteriophage. *Nucleic Acids Research* 39: 2116-2129

Fong SS, Marciniak JY, Palsson BO (2003) Description and interpretation of adaptive evolution of *Escherichia coli* K-12 MG1655 by using a genome-scale in silico metabolic model. *J Bacteriol* 185: 6400-6408

Forterre P (2005) The two ages of the RNA world, and the transition to the DNA world: a story of viruses and cells. *Biochimie* 87: 793-803

Fozo EM (2012) New type I toxin-antitoxin families from "wild" and laboratory strains of *E-coli* Ibs-Sib, ShoB-OhsC and Zor-Orz. *Rna Biol* 9: 1504-1512

Fozo EM, Hemm MR, Storz G (2008) Small Toxic Proteins and the Antisense RNAs That Repress Them. *Microbiol Mol Biol R* 72: 579-589

Francino MP (2015) Antibiotics and the Human Gut Microbiome: Dysbioses and Accumulation of Resistances. *Frontiers in microbiology* 6: 1543

Fraser PD, Enfissi EMA, Goodfellow M, Eguchi T, Bramley PM (2007) Metabolite profiling of plant carotenoids using the matrix-assisted laser desorption ionization time-of-flight mass spectrometry. *Plant J* 49: 552-564

Fraser RS, Creanor J (1975) The mechanism of inhibition of ribonucleic acid synthesis by 8-hydroxyquinoline and the antibiotic lomofungin. *Biochem J* 147: 401-10

Friedman S, Gots JS (1953) The purine and pyrimidine metabolism of normal and phage-infected *Escherichia coli*. *J Biol Chem* 201: 125-35

Frost G, Sleeth ML, Sahuri-Arisoylu M, Lizarbe B, Cerdan S, Brody L, Anastasovska J, Ghourab S, Hankir M, Zhang S, Carling D, Swann JR, Gibson G, Viardot A, Morrison D, Louise Thomas E, Bell JD (2014) The short-chain fatty acid acetate reduces appetite via a central homeostatic mechanism. *Nature communications* 5: 3611

Fuhrman JA (1999) Marine viruses and their biogeochemical and ecological effects. *Nature* 399: 541-548

Fujimoto J, Matsuki T, Sasamoto M, Tomii Y, Watanabe K (2008) Identification and quantification of *Lactobacillus casei* strain Shirota in human feces with strain-specific primers derived from randomly amplified polymorphic DNA. *Int J Food Microbiol* 126: 210-215

Fujisawa H, Morita M (1997) Phage DNA packaging. *Genes Cells* 2: 537-545

Fuller R (1989) Probiotics in man and animals. *J Appl Bacteriol* 66: 365-78

Futerman AH, Riezman H (2005) The ins and outs of sphingolipid synthesis. *Trends Cell Biol* 15: 312-8

Gao Z, Yin J, Zhang J, Ward RE, Martin RJ, Lefevre M, Cefalu WT, Ye J (2009) Butyrate improves insulin sensitivity and increases energy expenditure in mice. *Diabetes* 58: 1509-17

Garcia DE, Baidoo EE, Benke PI, Pingitore F, Tang YJ, Villa S, Keasling JD (2008) Separation and mass spectrometry in microbial metabolomics. *Curr Opin Microbiol* 11: 233-9

Garcia E, Elliott JM, Ramanculov E, Chain PS, Chu MC, Molineux IJ (2003) The genome sequence of *Yersinia pestis* bacteriophage phiA1122 reveals an intimate history with the coliphage T3 and T7 genomes. *J Bacteriol* 185: 5248-62

Gardes M, Bruns TD (1993) ITS primers with enhanced specificity for basidiomycetes--application to the identification of mycorrhizae and rusts. *Mol Ecol* 2: 113-8

Gephart SM, McGrath JM, Effken JA, Halpern MD (2012) Necrotizing Enterocolitis Risk: State of the Science. *Advances in Neonatal Care* 12: 77-89

Gerna G, Passarani N, Battaglia M, Rondanelli EG (1985) Human enteric coronaviruses: antigenic relatedness to human coronavirus OC43 and possible etiologic role in viral gastroenteritis. *J Infect Dis* 151: 796-803

Ghazalpour, A., Cespedes, I., Bennett, B. J., & Allayee, H. (2016). Expanding role of gut microbiota in lipid metabolism. *Current opinion in lipidology*. 27: 141-147

Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A (2005) Galaxy: A platform for interactive large-scale genome analysis. *Genome Res* 15: 1451-1455

Gibbs KA, Isaac DD, Xu J, Hendrix RW, Silhavy TJ, Theriot JA (2004) Complex spatial distribution and dynamics of an abundant *Escherichia coli* outer membrane protein, LamB. *Mol Microbiol* 53: 1771-83

Gibson GR, Roberfroid MB (1995) Dietary modulation of the human colonic microbiota: introducing the concept of prebiotics. *J Nutr* 125: 1401-12

Gibson MK, Forsberg KJ, Dantas G (2015) Improved annotation of antibiotic resistance determinants reveals microbial resistomes cluster by ecology. *Isme Journal* 9: 207-216

Gill CIR, Rowland IR (2002) Diet and cancer: assessing the risk. *Brit J Nutr* 88: S73-S87

Gill SR, Pop M, Deboy RT, Eckburg PB, Turnbaugh PJ, Samuel BS, Gordon JI, Relman DA, Fraser-Liggett CM, Nelson KE (2006) Metagenomic analysis of the human distal gut microbiome. *Science* 312: 1355-9

Giwerzman B, Lambert PA, Rosdahl VT, Shand GH, Hoiby N (1990) Rapid Emergence of Resistance in *Pseudomonas-Aeruginosa* in Cystic-Fibrosis Patients Due to In vivo Selection of Stable Partially Derepressed Beta-Lactamase Producing Strains. *J Antimicrob Chemoth* 26: 247-259

Glass RI, Noel J, Ando T, Fankhauser R, Belliot G, Mounts A, Parashar UD, Bresee JS, Monroe SS (2000) The epidemiology of enteric caliciviruses from humans: A reassessment using new diagnostics. *J Infect Dis* 181: S254-S261

Golais F, Holly J, Vitkovska J (2013) Coevolution of bacteria and their viruses. *Folia Microbiol* 58: 177-186

Gottesma.Mm, Gottesma.Me, Gottesma.S, Gellert M (1974) Characterization of Bacteriophage-Lambda Reverse as an *Escherichia-Coli* Phage Carrying a Unique Set of Host-Derived Recombination Functions. *Journal of Molecular Biology* 88: 471-&

Gowen B, Bamford JKH, Bamford DH, Fuller SD (2003) The tailless icosahedral membrane virus PRD1 localizes the proteins involved in genome packaging and injection at a unique vertex. *Journal of virology* 77: 7863-7871

Graessler J, Qin Y, Zhong H, Zhang J, Licinio J, Wong ML, Xu A, Chavakis T, Bornstein AB, Ehrhart-Bornstein M, Lamounier-Zepter V, Lohmann T, Wolf T, Bornstein SR (2013) Metagenomic sequencing of the human gut microbiome before and after bariatric surgery in obese patients with type 2 diabetes: correlation with inflammatory and metabolic parameters. *Pharmacogenomics J* 13: 514-522

Grayson P, Molineux IJ (2007) Is phage DNA 'injected' into cells-biologists and physicists can agree. *Curr Opin Microbiol* 10: 401-409

Graziewicz MA, Zastawny TH, Olinski R, Speina E, Siedlecki J, Tudek B (2000) Fapyadenine is a moderately efficient chain terminator for prokaryotic DNA polymerases. *Free Radical Bio Med* 28: 75-83

Greenblum S, Turnbaugh PJ, Borenstein E (2012) Metagenomic systems biology of the human gut microbiome reveals topological shifts associated with obesity and inflammatory bowel disease. *Proceedings of the National Academy of Sciences of the United States of America* 109: 594-9

Guerrero-Preston R, Godoy-Vitorino F, Jedlicka A, Rodriguez-Hilario A, Gonzalez H, Bondy J, Lawson F, Folawiyo O, Michailidi C, Dziedzic A, Thangavel R, Hadar T, Noordhuis MG, Westra W, Koch W, Sidransky D (2016) 16S rRNA amplicon sequencing identifies microbiota associated with oral cancer, human papilloma virus infection and surgical treatment. *Oncotarget* 7: 51320-51334

Guerrero ML, Noel JS, Mitchell DK, Calva JJ, Morrow AL, Martinez J, Rosales G, Velazquez FR, Monroe SS, Glass RI, Pickering LK, Ruiz-Palacios GM (1998) A prospective study of astrovirus diarrhea of infancy in Mexico City. *Pediatr Infect Dis J* 17: 723-7

Guo L, Hua X, Zhang W, Yang S, Shen Q, Hu H, Li J, Liu Z, Wang X, Wang H, Zhou C, Cui L (2017) Viral metagenomics analysis of feces from coronary heart disease patients reveals the genetic diversity of the Microviridae. *Virol Sin* 32: 130-138

Gurevich A, Saveliev V, Vyahhi N, Tesler G (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics* 29: 1072-5

Gweon HS, Oliver A, Taylor J, Booth T, Gibbs M, Read DS, Griffiths RI, Schonrogge K (2015) PIPITS: an automated pipeline for analyses of fungal internal transcribed spacer sequences from the Illumina sequencing platform. *Methods Ecol Evol* 6: 973-980

Hack M, Horbar JD, Malloy MH, Tyson JE, Wright E, Wright L (1991) Very low birth weight outcomes of the National Institute of Child Health and Human Development Neonatal Network. *Pediatrics* 87: 587-97

Hague A, Elder DJ, Hicks DJ, Paraskeva C (1995) Apoptosis in colorectal tumour cells: induction by the short chain fatty acids butyrate, propionate and acetate and by the bile salt deoxycholate. *Int J Cancer* 60: 400-6

Hankin ME. (1896). L'action bactéricide des eaux de la Jumna et du Gange sur le vibron du cholera. *Ann Inst Pasteur (Paris)*. 10: 511–523.

Hallstrom M, Eerola E, Vuento R, Janas M, Tammela O (2004) Effects of mode of delivery and necrotising enterocolitis on the intestinal microflora in preterm infants. *Eur J Clin Microbiol Infect Dis* 23: 463-70

Hamer HM, De Preter V, Windey K, Verbeke K (2012) Functional analysis of colonic bacterial metabolism: relevant to health? *Am J Physiol Gastrointest Liver Physiol* 302: G1-9

Handa N, Kobayashi I (2005) Type III restriction is alleviated by bacteriophage (RecE) homologous recombination function but enhanced by bacterial (RecBCD) function. *J Bacteriol* 187: 7362-7373

Hang JQ, Catalano CE, Feiss M (2001) The functional asymmetry of cosN, the nicking site for bacteriophage lambda DNA packaging, is dependent on the terminase binding site, cosB. *Biochemistry-U S A* 40: 13370-13377

Hansen-Hagge T, Lehmann V, Seydel U, Lindner B, Zähringer U (1985) Isolation and structural analysis of two lipid A precursors from a KDO deficient mutant of *Salmonella typhimurium* differing in their hexadecanoic acid content. *Archives of Microbiology* 141: 353-358

Hansen R, Russell RK, Reiff C, Louis P, McIntosh F, Berry SH, Mukhopadhyaya I, Bisset WM, Barclay AR, Bishop J (2012) Microbiota of de-novo pediatric IBD: increased *Faecalibacterium prausnitzii* and reduced bacterial diversity in Crohn's but not in ulcerative colitis. *The American journal of gastroenterology* 107: 1913

Haque QM, Sugiyama A, Iwade Y, Midorikawa Y, Yamauchi T (1996) Diarrheal and environmental isolates of *Aeromonas* spp. produce a toxin similar to Shiga-like toxin 1. *Curr Microbiol* 32: 239-45

Hara H, Haga S, Aoyama Y, Kiriya S (1999) Short-chain fatty acids suppress cholesterol synthesis in rat liver and intestine. *J Nutr* 129: 942-948

- Hardman M, Makarov AA (2003) Interfacing the orbitrap mass analyzer to an electrospray ion source. *Anal Chem* 75: 1699-705
- Hargreaves KR, Otieno JR, Thanki A, Blades MJ, Millard AD, Browne HP, Lawley TD, Clokie MR (2015) As Clear as Mud? Determining the Diversity and Prevalence of Prophages in the Draft Genomes of Estuarine Isolates of *Clostridium difficile*. *Genome biology and evolution* 7: 1842-55
- Hashemolhosseini S, Holmes Z, Mutschler B, Henning U (1994) Alterations of Receptor Specificities of Coliphages of the T2 Family. *Journal of Molecular Biology* 240: 105-110
- Hayashi T, Makino K, Ohnishi M, Kurokawa K, Ishii K, Yokoyama K, Han CG, Ohtsubo E, Nakayama K, Murata T, Tanaka M, Tobe T, Iida T, Takami H, Honda T, Sasakawa C, Ogasawara N, Yasunaga T, Kuhara S, Shiba T et al. (2001) Complete genome sequence of enterohemorrhagic *Escherichia coli* O157:H7 and genomic comparison with a laboratory strain K-12. *DNA research : an international journal for rapid publication of reports on genes and genomes* 8: 11-22
- Heida FH, van Zoonen AGJF, Hulscher JBF, te Kiefte BJC, Wessels R, Kooi EMW, Bos AF, Harmsen HJM, de Goffau MC (2016) A Necrotizing Enterocolitis-Associated Gut Microbiota Is Present in the Meconium: Results of a Prospective Study. *Clin Infect Dis* 62: 863-870
- Heimann E, Nyman M, Palbrink AK, Lindkvist-Petersson K, Degerman E (2016) Branched short-chain fatty acids modulate glucose and lipid metabolism in primary adipocytes. *Adipocyte* 5: 359-368
- Helander IM, Lindner B, Seydel U, Vaara M (1993) Defective biosynthesis of the lipid A component of temperature-sensitive *firA* (*omsA*) mutant of *Escherichia coli*. *Eur J Biochem* 212: 363-9
- Hendrix RW (2003) Bacteriophage genomics. *Current Opinion in Microbiology* 6: 506-511
- Hendrix RW, Duda RL (1998) Bacteriophage HK97 head assembly: a protein ballet. *Adv Virus Res* 50: 235-88
- Hiramatsu K (2001) Vancomycin-resistant *Staphylococcus aureus*: a new model of antibiotic resistance. *Lancet Infect Dis* 1: 147-55

Ho TD, Slauch JM (2001) OmpC is the receptor for Gifsy-1 and Gifsy-2 bacteriophages of Salmonella. *J Bacteriol* 183: 1495-1498

Hoffmann C., Dollive S., Grunberg S., Chen J., Li H., Wu G.D., Lewis J.D., Bushman F.D. (2013) Archaea and Fungi of the Human Gut Microbiome: Correlations with Diet and Bacterial Residents. *PLOS ONE* 8(6)

Hohl M, Kurtz S, Ohlebusch E (2002) Efficient multiple genome alignment. *Bioinformatics* 18 Suppl 1: S312-20

Hoiby N, Bjarnsholt T, Givskov M, Molin S, Ciofu O (2010) Antibiotic resistance of bacterial biofilms. *Int J Antimicrob Agents* 35: 322-32

Holt GS, Lodge JK, McCarthy AJ, Graham AK, Young G, Bridge SH, Brown AK, Veses-Garcia M, Lanyon CV, Sails A, Allison HE, Smith DL (2017) Shigatoxin encoding Bacteriophage phi24B modulates bacterial metabolism to raise antimicrobial tolerance. *Scientific reports* 7: 40424

Holtz LR, Finkbeiner SR, Kirkwood CD, Wang D (2008) Identification of a novel picornavirus related to cosaviruses in a child with acute diarrhea. *Virol J* 5: 159

Honeyman MC, Coulson BS, Stone NL, Gellert SA, Goldwater PN, Steele CE, Couper JJ, Tait BD, Colman PG, Harrison LC (2000) Association between rotavirus infection and pancreatic islet autoimmunity in children at risk of developing type 1 diabetes. *Diabetes* 49: 1319-24

Horning EC, Horning MG (1971) Metabolic profiles: gas-phase methods for analysis of metabolites. *Clin Chem* 17: 802-9

Hosny M, Cassir N, La Scola B (2017) Updating on gut microbiota and its relationship with the occurrence of necrotizing enterocolitis. *Human Microbiome Journal* 4: 14-19

Hsiao A, Ahmed AM, Subramanian S, Griffin NW, Drewry LL, Petri WA, Jr., Haque R, Ahmed T, Gordon JI (2014) Members of the human gut microbiota involved in recovery from *Vibrio cholerae* infection. *Nature* 515: 423-6

Hsueh W, Caplan MS, Qu XW, Tan XD, De Plaen IG, Gonzalez-Crussi F (2003) Neonatal necrotizing enterocolitis: clinical considerations and pathogenetic concepts. *Pediatr Dev Pathol* 6: 6-23

Hugonnet JE, Haddache N, Veckerie C, Dubost L, Marie A, Shikura N, Mainardi JL, Rice LB, Arthur M (2014) Peptidoglycan Cross-Linking in Glycopeptide-Resistant Actinomycetales. *Antimicrob Agents Ch* 58: 1749-1756

Hulo C, Masson P, de Castro E, Auchincloss AH, Foulger R, Poux S, Lomax J, Bougueleret L, Xenarios I, Le Mercier P (2017) The ins and outs of eukaryotic viruses: Knowledge base and ontology of a viral infection. *Plos One* 12

Hunt M, Kikuchi T, Sanders M, Newbold C, Berriman M, Otto TD (2013) REAPR: a universal tool for genome assembly evaluation. *Genome Biol* 14: R47

Huson DH, Auch AF, Qi J, Schuster SC (2007a) MEGAN analysis of metagenomic data. *Genome Research* 17: 377-386

Huson DH, Auch AF, Qi J, Schuster SC (2007b) MEGAN analysis of metagenomic data. *Genome Res* 17: 377-86

Hwang Y, Feiss M (1995) A defined system for in vitro lambda DNA packaging. *Virology* 211: 367-76

Ikeda M, Hosotani T, Kurimoto K, Mori T, Ueda T, Kotake Y, Sakakibara B (1979) The differences of the metabolism related to vitamin B6-dependent enzymes among vitamin B6-deficient germ-free and conventional rats. *J Nutr Sci Vitaminol (Tokyo)* 25: 131-9

Illumina (2010) Illumina Sequencing Technology: Highest data accuracy, simple workflow, and a broad range of applications. In *Technology Spotlight: Illumina Sequencing*, Illumina, Inc

Issam Raad HAH, Gassan Chaiban (2004) Methods for coating and impregnating medical devices with antiseptic compositions. In *University of Texas System*,

J White T, Bruns T, Lee S, Taylor J, A Innis M, H Gelfand D, Sninsky J (1990) Amplification and Direct Sequencing of Fungal Ribosomal RNA Genes for Phylogenetics.

Jaitly N, Monroe ME, Petyuk VA, Clauss TRW, Adkins JN, Smith RD (2006) Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline. *Anal Chem* 78: 7397-7409

Jakobsson HE, Jernberg C, Andersson AF, Sjolund-Karlsson M, Jansson JK, Engstrand L (2010) Short-Term Antibiotic Treatment Has Differing Long-Term Impacts on the Human Throat and Gut Microbiome. *Plos One* 5

Jameel S, Durgapal H, Habibullah CM, Khuroo MS, Panda SK (1992) Enteric non-A, non-B hepatitis: epidemics, animal transmission, and hepatitis E virus detection by the polymerase chain reaction. *J Med Virol* 37: 263-70

James CE, Stanley KN, Allison HE, Flint HJ, Stewart CS, Sharp RJ, Saunders JR, McCarthy AJ (2001) Lytic and lysogenic infection of diverse *Escherichia coli* and *Shigella* strains with a verocytotoxigenic bacteriophage. *Applied and environmental microbiology* 67: 4335-7

Jan G, Belzacq AS, Haouzi D, Rouault A, Metivier D, Kroemer G, Brenner C (2002) Propionibacteria induce apoptosis of colorectal carcinoma cells via short-chain fatty acids acting on mitochondria. *Cell Death Differ* 9: 179-88

Jemal A, Bray F, Center MM, Ferlay J, Ward E, Forman D (2011) Global cancer statistics. *CA: a cancer journal for clinicians* 61: 69-90

Jernberg C, Lofmark S, Edlund C, Jansson JK (2007) Long-term ecological impacts of antibiotic administration on the human intestinal microbiota. *Isme J* 1: 56-66

Jess T, Gomborg M, Matzen P, Munkholm P, Sorensen TI (2005) Increased risk of intestinal cancer in Crohn's disease: a meta-analysis of population-based cohort studies. *Am J Gastroenterol* 100: 2724-9

- Jiang H, Lei R, Ding SW, Zhu S (2014) Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics* 15: 182
- Jimenez A, Velazquez JB, Rodriguez J, Tinajas A, Villa TG (1994) Prevalence of fluoroquinolone resistance in clinical strains of *Campylobacter jejuni* isolated in Spain. *The Journal of antimicrobial chemotherapy* 33: 188-90
- Johannes L, Mayor S (2010) Induced domain formation in endocytic invagination, lipid sorting, and scission. *Cell* 142: 507-10
- Johannes L, Romer W (2010) Shiga toxins--from cell biology to biomedical applications. *Nature reviews Microbiology* 8: 105-16
- Johannessen GS, James CE, Allison HE, Smith DL, Saunders JR, McCarthy AJ (2005) Survival of a Shiga toxin-encoding bacteriophage in a compost model. *FEMS Microbiol Lett* 245: 369-75
- Johansen BK, Wasteson Y, Granum PE, Brynstad S (2001) Mosaic structure of Shiga-toxin-2-encoding phages isolated from *Escherichia coli* O157 : H7 indicates frequent gene exchange between lambdoid phage genomes. *Microbiol-Sgm* 147: 1929-1936
- Johnsen PH, Hilpusch F, Cavanagh JP, Leikanger IS, Kolstad C, Valle PC, Goll R (2018) Faecal microbiota transplantation versus placebo for moderate-to-severe irritable bowel syndrome: a double-blind, randomised, placebo-controlled, parallel-group, single-centre trial. *Lancet Gastroenterol Hepatol* 3: 17-24
- Johnson AD, Poteete AR, Lauer G, Sauer RT, Ackers GK, Ptashne M (1981) Lambda-Repressor and Cro - Components of an Efficient Molecular Switch. *Nature* 294: 217-223
- Jones JB, Vallad GE, Iriarte FB, Obradović A, Wernsing MH, Jackson LE, Balogh B, Hong JC, Momol M (2012) Considerations for using bacteriophages for plant disease control. *Bacteriophage* 2: 208-214

Jones MK, Watanabe M, Zhu S, Graves CL, Keyes LR, Grau KR, Gonzalez-Hernandez MB, Iovine NM, Wobus CE, Vinje J, Tibbetts SA, Wallet SM, Karst SM (2014) Enteric bacteria promote human and mouse norovirus infection of B cells. *Science* 346: 755-9

Joshi NA FJ (2011) Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files.

KA W (2017) DNA Sequencing Costs: Data. In Large-Scale Genome Sequencing and Analysis Centers (LSAC): National Human Genome Research Institute

Kadaja M, Silla T, Ustav E, Ustav M (2009) Papillomavirus DNA replication — From initiation to genomic instability. *Virology* 384: 360-368

Kaiser D, Dworkin M (1975) Gene transfer to myxobacterium by *Escherichia coli* phage P1. *Science* 187: 653-4

Kallus SJ, Brandt LJ (2012) The intestinal microbiota and obesity. *J Clin Gastroenterol* 46: 16-24

Kamada N, Chen GY, Inohara N, Nunez G (2013) Control of pathogens and pathobionts by the gut microbiota. *Nat Immunol* 14: 685-90

Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 45: D353-D361

Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* 28: 27-30

Kaper JB (1998) *Escherichia coli* O157:H7 and other shiga toxin producing *E. coli* strains. ASM Press, Washington, DC

Karaki S, Tazoe H, Hayashi H, Kashiwabara H, Tooyama K, Suzuki Y, Kuwahara A (2008) Expression of the short-chain fatty acid receptor, GPR43, in the human colon. *J Mol Histol* 39: 135-42

Karłowicz MG (1993) Risk-Factors Associated with Fungal Peritonitis in Very-Low-Birth-Weight Neonates with Severe Necrotizing Enterocolitis - a Case-Control Study. *Pediatr Infect Dis J* 12: 574-577

Katajamaa M, Miettinen J, Oresic M (2006) MZmine: toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics* 22: 634-636

Kawano K, Okada M, Haga T, Maeda K, Goto Y (2008) Relationship between pathogenicity for humans and stx genotype in Shiga toxin-producing *Escherichia coli* serotype O157. *Eur J Clin Microbiol* 27: 227-232

Kawano M, Aravind L, Storz G (2007) An antisense RNA controls synthesis of an SOS-induced toxin evolved from an antitoxin. *Mol Microbiol* 64: 738-54

Kearse M, Moir R, Wilson A, Stones-Havas S, Cheung M, Sturrock S, Buxton S, Cooper A, Markowitz S, Duran C, Thierer T, Ashton B, Meintjes P, Drummond A (2012) Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* 28: 1647-1649

Kekarainen T, Segales J (2009) Torque teno virus infection in the pig and its potential role as a model of human infection. *Vet J* 180: 163-168

Kernbauer E, Ding Y, Cadwell K (2014) An enteric virus can replace the beneficial function of commensal bacteria. *Nature* 516: 94-U223

Kim MS, Bae JW (2018) Lysogeny is prevalent and widely distributed in the murine gut microbiota. *ISME J*

Kind T, Fiehn O (2006) Metabolomic database annotations via query of elemental compositions: Mass accuracy is insufficient even at less than 1 ppm. *Bmc Bioinformatics* 7

Koek MM, Muilwijk B, van der Werf MJ, Hankemeier T (2006) Microbial metabolomics with gas chromatography/mass spectrometry. *Anal Chem* 78: 1272-1281

- Kokot M, Dlugosz M, Deorowicz S (2017) KMC 3: counting and manipulating k-mer statistics. *Bioinformatics* 33: 2759-2761
- Koley D, Ramsey MM, Bard AJ, Whiteley M (2011) Discovery of a biofilm electroline using real-time 3D metabolite analysis. *Proceedings of the National Academy of Sciences of the United States of America* 108: 19996-20001
- Koljalg U, Larsson KH, Abarenkov K, Nilsson RH, Alexander IJ, Eberhardt U, Erland S, Hoiland K, Kjöller R, Larsson E, Pennanen T, Sen R, Taylor AF, Tedersoo L, Vralstad T, Ursing BM (2005) UNITE: a database providing web-based methods for the molecular identification of ectomycorrhizal fungi. *New Phytol* 166: 1063-8
- Kolmogorov M, Raney B, Paten B, Pham S (2014) Ragout-a reference-assisted assembly tool for bacterial genomes. *Bioinformatics* 30: i302-9
- Kommineni S, Bretl DJ, Lam V, Chakraborty R, Hayward M, Simpson P, Cao Y, Bousounis P, Kristich CJ, Salzman NH (2015) Bacteriocin production augments niche competition by enterococci in the mammalian gastrointestinal tract. *Nature* 526: 719-22
- Kong LC, Tap J, Aron-Wisnewsky J, Pelloux V, Basdevant A, Bouillot JL, Zucker JD, Dore J, Clement K (2013) Gut microbiota after gastric bypass in human obesity: increased richness and associations of bacterial genera with adipose tissue genes. *Am J Clin Nutr* 98: 16-24
- Koonin EV, Gorbalenya AE, Chumakov KM (1989) Tentative Identification of Rna-Dependent Rna-Polymerases of Dsrna Viruses and Their Relationship to Positive Strand Rna Viral Polymerases. *Febs Lett* 252: 42-46
- Koonin EV, Ilyina TV (1992) Geminivirus Replication Proteins Are Related to Prokaryotic Plasmid Rolling Circle DNA-Replication Initiator Proteins. *J Gen Virol* 73: 2763-2766

Koonin, E. V., Krupovic, M., & Yutin, N. (2015). Evolution of double-stranded DNA viruses of eukaryotes: from bacteriophages to transposons to giant viruses. *Annals of the New York Academy of Sciences*. 1341: 10-24.

Koskella B, Brockhurst MA (2014) Bacteria-phage coevolution as a driver of ecological and evolutionary processes in microbial communities. *Fems Microbiology Reviews* 38: 916-931

Kostic AD, Chun E, Robertson L, Glickman JN, Gallini CA, Michaud M, Clancy TE, Chung DC, Lochhead P, Hold GL, El-Omar EM, Brenner D, Fuchs CS, Meyerson M, Garrett WS (2013) *Fusobacterium nucleatum* potentiates intestinal tumorigenesis and modulates the tumor-immune microenvironment. *Cell Host Microbe* 14: 207-15

Kozich JJ, Westcott SL, Baxter NT, Highlander SK, Schloss PD (2013) Development of a Dual-Index Sequencing Strategy and Curation Pipeline for Analyzing Amplicon Sequence Data on the MiSeq Illumina Sequencing Platform. *Applied and Environmental Microbiology* 79: 5112-5120

Krajmalnik-Brown R, Ilhan ZE, Kang DW, DiBaise JK (2012) Effects of gut microbes on nutrient absorption and energy regulation. *Nutr Clin Pract* 27: 201-14

Krishnamurthy SR, Janowski AB, Zhao GY, Barouch D, Wang D (2016) Hyperexpansion of RNA Bacteriophage Diversity. *PLoS Biol* 14

Kristensen DM, Mushegian AR, Dolja VV, Koonin EV (2010) New dimensions of the virus world discovered through metagenomics. *Trends Microbiol* 18: 11-19

Kristensen DM, Waller AS, Yamada T, Bork P, Mushegian AR, Koonin EV (2013) Orthologous Gene Clusters and Taxon Signature Genes for Viruses of Prokaryotes. *J Bacteriol* 195: 941-950

Krupovic M (2013) Networks of evolutionary interactions underlying the polyphyletic origin of ssDNA viruses. *Curr Opin Virol* 3: 578-586

Krupovic M, Bamford DH (2009) Does the evolution of viral polymerases reflect the origin and evolution of viruses? *Nature Reviews Microbiology* 7: 250-250

Krupovic M, Forterre P (2015) Single-stranded DNA viruses employ a variety of mechanisms for integration into host genomes. *Ann Ny Acad Sci* 1341: 41-53

Krupovic M, Prangishvili D, Hendrix RW, Bamford DH (2011) Genomics of Bacterial and Archaeal Viruses: Dynamics within the Prokaryotic Virosphere. *Microbiol Mol Biol R* 75: 610-+

Krupovic M, Ravantti JJ, Bamford DH (2009) Geminiviruses: a tale of a plasmid becoming a virus. *Bmc Evol Biol* 9

Kugelberg E (2013) Surgery: Altered gut microbiota trigger weight loss. *Nat Rev Endocrinol* 9: 314

Kuhnau J (1976) The flavonoids. A class of semi-essential food components: their role in human nutrition. *World Rev Nutr Diet* 24: 117-91

Kurioka T, Yunou Y, Kita E (1998) Enhancement of susceptibility to shiga toxin-producing *Escherichia coli* O157 : H7 by protein calorie malnutrition in mice. *Infection and Immunity* 66: 1726-1734

Kurtz S, Narechania A, Stein JC, Ware D (2008) A new method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. *BMC genomics* 9: 517

Kyrpides NC, Ouzounis CA (1999) Whole-genome sequence annotation: 'Going wrong with confidence'. *Mol Microbiol* 32: 886-887

La Rosa PS, Warner BB, Zhou Y, Weinstock GM, Sodergren E, Hall-Moore CM, Stevens HJ, Bennett WE, Jr., Shaikh N, Linneman LA, Hoffmann JA, Hamvas A, Deych E, Shands BA, Shannon WD, Tarr PI (2014) Patterned progression of bacterial populations in the premature infant gut. *Proceedings of the National Academy of Sciences of the United States of America* 111: 12522-7

Labrie SJ, Samson JE, Moineau S (2010) Bacteriophage resistance mechanisms. *Nature reviews Microbiology* 8: 317-27

Lafont F, Tran Van Nhieu G, Hanada K, Sansonetti P, van der Goot FG (2002) Initial steps of Shigella infection depend on the cholesterol/sphingolipid raft-mediated CD44-IpaB interaction. *Embo J* 21: 4449-57

Lam YY, Maguire S, Palacios T, Caterson ID (2017) Are the Gut Bacteria Telling Us to Eat or Not to Eat? Reviewing the Role of Gut Microbiota in the Etiology, Disease Progression and Treatment of Eating Disorders. *Nutrients* 9

Landy J, Al-Hassi HO, McLaughlin SD, Walker AW, Ciclitira PJ, Nicholls RJ, Clark SK, Hart AL (2011) Review article: faecal transplantation therapy for gastrointestinal disease. *Aliment Pharmacol Ther* 34: 409-15

Langmead B, Nellore A (2018) Cloud computing for genomic data analysis and collaboration. *Nat Rev Genet* 19: 208-219

Laparra JM, Sanz Y (2010) Interactions of gut microbiota with functional food components and nutraceuticals. *Pharmacol Res* 61: 219-25

Lathe R, Lecocq JP (1977) The *firA* gene, a locus involved in the expression of rifampicin resistance in *Escherichia coli*. I. Characterisation of λ *firA* transducing phages constructed in vitro. *Molecular & general genetics* : MGG 154: 43-51

Lauber CL, Zhou N, Gordon JI, Knight R, Fierer N (2010) Effect of storage conditions on the assessment of bacterial community structure in soil and human-associated samples. *FEMS Microbiol Lett* 307: 80-6

LeBlanc JG, Milani C, de Giori GS, Sesma F, van Sinderen D, Ventura M (2013) Bacteria as vitamin suppliers to their host: a gut microbiota perspective. *Curr Opin Biotech* 24: 160-168

Lee HC, Jenner AM, Low CS, Lee YK (2006) Effect of tea phenolics and their aromatic fecal bacterial metabolites on intestinal microbiota. *Res Microbiol* 157: 876-884

Lefkowitz EJ, Dempsey DM, Hendrickson RC, Orton RJ, Siddell SG, Smith DB (2018) Virus taxonomy: the database of the International Committee on Taxonomy of Viruses (ICTV). *Nucleic Acids Res* 46: D708-D717

Lehnherr H, Yarmolinsky MB (1995) Addition Protein Phd of Plasmid Prophage P1 Is a Substrate of the Clpxp Serine-Protease of Escherichia-Coli. *P Natl Acad Sci USA* 92: 3274-3277

Lehri B, Seddon AM, Karlyshev AV (2017) The hidden perils of read mapping as a quality assessment tool in genome sequencing. *Scientific reports* 7

Levy O (2007) Innate immunity of the newborn: basic mechanisms and clinical correlates. *Nat Rev Immunol* 7: 379-90

Lewis DEA, Gussin GN, Adhya S (2016) New Insights into the Phage Genetic Switch: Effects of Bacteriophage Lambda Operator Mutations on DNA Looping and Regulation of PR, PL, and PRM. *J Mol Biol* 428: 4438-4456

Ley RE, Peterson DA, Gordon JI (2006) Ecological and evolutionary forces shaping microbial diversity in the human intestine. *Cell* 124: 837-48

Li D, Breiman A, le Pendu J, Uyttendaele M (2015a) Binding to histo-blood group antigen-expressing bacteria protects human norovirus from acute heat stress. *Frontiers in microbiology* 6: 659

Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25: 1754-60

Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25: 1966-7

Li YL, Weng JC, Hsiao CC, Chou MT, Tseng CW, Hung JH (2015b) PEAT: an intelligent and efficient paired-end sequencing adapter trimming algorithm. *BMC Bioinformatics* 16 Suppl 1: S2

Liengme B (2002) *A Guide to Microsoft Excel 2002 for Scientists and Engineers*. Butterworth-Heinemann, London

Lim ES, Zhou Y, Zhao G, Bauer IK, Droit L, Ndao IM, Warner BB, Tarr PI, Wang D, Holtz LR (2015a) Early life dynamics of the human gut virome and bacterial microbiome in infants. *Nat Med* 21: 1228-34

Lim ES, Zhou Y, Zhao G, Bauer IK, Droit L, Ndao IM, Warner BB, Tarr PI, Wang D, Holtz LR (2015b) Early life dynamics of the human gut virome and bacterial microbiome in infants. *Nature medicine* 21: 1228-1234

Lin HV, Frassetto A, Kowalik EJ, Jr., Nawrocki AR, Lu MM, Kosinski JR, Hubert JA, Szeto D, Yao X, Forrest G, Marsh DJ (2012) Butyrate and propionate protect against diet-induced obesity and regulate gut hormones via free fatty acid receptor 3-independent mechanisms. *Plos One* 7: e35240

Lin J, Nishino K, Roberts MC, Tolmasky M, Aminov RI, Zhang L (2015) Mechanisms of antibiotic resistance. *Frontiers in microbiology* 6

Lin L, Bitner R, Edlin G (1977) Increased reproductive fitness of *Escherichia coli* lambda lysogens. *Journal of virology* 21: 554-9

Lin PW, Stoll BJ (2006) Necrotising enterocolitis. *Lancet* 368: 1271-83

Lin S, Hanson RE, Cronan JE (2010) Biotin synthesis begins by hijacking the fatty acid synthetic pathway. *Nat Chem Biol* 6: 682-8

Liou AP, Paziuk M, Luevano JM, Machineni S, Turnbaugh PJ, Kaplan LM (2013) Conserved Shifts in the Gut Microbiota Due to Gastric Bypass Reduce Host Weight and Adiposity. *Sci Transl Med* 5

Lischer HEL, Shimizu KK (2017) Reference-guided de novo assembly approach improves genome reconstruction for related species. *BMC Bioinformatics* 18: 474

Liste MB, Natera I, Suarez JA, Pujol FH, Liprandi F, Ludert JE (2000) Enteric virus infections and diarrhea in healthy and human immunodeficiency virus-infected children. *J Clin Microbiol* 38: 2873-7

Liu BH, Liu CM, Li DH, Li YR, Ting HF, Yiu SM, Luo RB, Lam TW (2016) BASE: a practical de novo assembler for large genomes using long NGS reads. *BMC genomics* 17

Liu F, Huang J, Sadler JE (2011) Shiga toxin (Stx)1B and Stx2B induce von Willebrand factor secretion from human umbilical vein endothelial cells through different signaling pathways. *Blood* 118: 3392-3398

Liu H, Naismith JH, Hay RT (2003) Adenovirus DNA replication. *Current topics in microbiology and immunology* 272: 131-64

Liu SX, Li YH, Dai WK, Li XS, Qiu CZ, Ruan ML, Zou B, Dong C, Liu YH, He JY, Huang ZH, Shu SN (2017) Fecal microbiota transplantation induces remission of infantile allergic colitis through gut microbiota re-establishment. *World J Gastroentero* 23: 8570-8581

Livny J, Friedman DI (2004) Characterizing spontaneous induction of Stx encoding phages using a selectable reporter system. *Mol Microbiol* 51: 1691-1704

Llosa M, Gomis-Ruth FX, Coll M, de la Cruz Fd F (2002) Bacterial conjugation: a two-step mechanism for DNA transport. *Mol Microbiol* 45: 1-8

Lloyd-Price J, Abu-Ali G, Huttenhower C (2016) The healthy human microbiome. *Genome Med* 8: 51

Loman NJ, Quick J, Simpson JT (2015) A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods* 12: 733-U51

Lommen A (2009) MetAlign: Interface-Driven, Versatile Metabolomics Tool for Hyphenated Full-Scan Mass Spectrometry Data Preprocessing. *Anal Chem* 81: 3079-3086

Lopman B, Vennema H, Kohli E, Pothier P, Sanchez A, Negrodo A, Buesa J, Schreier E, Reacher M, Brown D, Gray J, Iturriza M, Gallimore C, Bottiger B, Hedlund KO, Torven M, von Bonsdorff CH, Maunula L, Poljsak-Prijatelj M, Zimsek J et al. (2004) Increase in viral gastroenteritis outbreaks in Europe and epidemic spread of new norovirus variant. *Lancet* 363: 682-8

Lorenz MG, Wackernagel W (1994) Bacterial Gene-Transfer by Natural Genetic-Transformation in the Environment. *Microbiol Rev* 58: 563-602

Loskill P, Pereira PM, Jung P, Bischoff M, Herrmann M, Pinho MG, Jacobs K (2014) Reduction of the Peptidoglycan Crosslinking Causes a Decrease in Stiffness of the *Staphylococcus aureus* Cell Envelope. *Biophys J* 107: 1082-1089

Lotka AJ (1932) The growth of mixed populations: Two species competing for a common food supply. *Journal of the Washington Academy of Sciences* 22: 461-469

Louis P, Hold GL, Flint HJ (2014) The gut microbiota, bacterial metabolites and colorectal cancer. *Nat Rev Microbiol* 12: 661-72

Lu J, Breitwieser FP, Thielen P, Salzberg SL (2017) Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science* 3: e104

Lugli GA, Milani C, Mancabelli L, van Sinderen D, Ventura M (2016) MEGAnnotator: a user-friendly pipeline for microbial genomes assembly and annotation. *FEMS Microbiol Lett* 363

Lukes J, Stensvold CR, Jirku-Pomajbikova K, Wegener Parfrey L (2015) Are Human Intestinal Eukaryotes Beneficial or Commensals? *PLoS Pathog* 11: e1005039

Lutz TA, Bueter M (2014) Physiological mechanisms behind Roux-en-Y gastric bypass surgery. *Dig Surg* 31: 13-24

Lwoff A, Horne R, Tournier P (1962) A system of viruses. *Cold Spring Harb Symp Quant Biol* 27: 51-5

Ma Y, You X, Mai G, Tokuyasu T, Liu C (2018) A human gut phage catalog correlates the gut phageome with type 2 diabetes. *Microbiome* 6: 24

Macconkey A (1905) Lactose-Fermenting Bacteria in Faeces. *J Hyg (Lond)* 5: 333-79

MacConkey AT (1908) Bile Salt Media and their advantages in some Bacteriological Examinations. *The Journal of Hygiene* 8: 322-334

Macfarlane GT, Gibson GR, Beatty E, Cummings JH (1992) Estimation of Short-Chain Fatty-Acid Production from Protein by Human Intestinal Bacteria Based on Branched-Chain Fatty-Acid Measurements. *Fems Microbiol Ecol* 101: 81-88

Macfarlane GT, Macfarlane S (1993) Factors affecting fermentation reactions in the large bowel. *Proc Nutr Soc* 52: 367-73

Macfarlane GT, Macfarlane S (2012) Bacteria, colonic fermentation, and gastrointestinal health. *J AOAC Int* 95: 50-60

Macfarlane S, Macfarlane GT (2003) Regulation of short-chain fatty acid production. *Proc Nutr Soc* 62: 67-72

Macpherson AJ, de Agüero MG, Ganai-Vonarburg SC (2017) How nutrition and the maternal microbiota shape the neonatal immune system. *Nat Rev Immunol* 17: 508-517

Maddox B (2003) The double helix and the 'wronged heroine'. *Nature* 421: 407-408

Magne F, Abely M, Boyer F, Morville P, Pochart P, Suau A (2006) Low species diversity and high interindividual variability in faeces of preterm infants as revealed by sequences of 16S rRNA genes and PCR-temporal temperature gradient gel electrophoresis profiles. *FEMS Microbiol Ecol* 57: 128-38

Mai V, Young CM, Ukhanova M, Wang X, Sun Y, Casella G (2011a) Fecal microbiota in premature infants prior to necrotizing enterocolitis. *PLoS One* 6

Mai V, Young CM, Ukhanova M, Wang X, Sun Y, Casella G, Theriaque D, Li N, Sharma R, Hudak M, Neu J (2011b) Fecal microbiota in premature infants prior to necrotizing enterocolitis. *Plos One* 6: e20647

Mallory A, Kern F, Jr., Smith J, Savage D (1973) Patterns of bile acids and microflora in the human small intestine. I. Bile acids. *Gastroenterology* 64: 26-33

Mallory A, Savage D, Kern F, Jr., Smith JG (1973) Patterns of bile acids and microflora in the human small intestine. II. Microflora. *Gastroenterology* 64: 34-42

Maltby R, Leatham-Jensen MP, Gibson T, Cohen PS, Conway T (2013) Nutritional basis for colonization resistance by human commensal *Escherichia coli* strains HS and Nissle 1917 against *E. coli* O157:H7 in the mouse intestine. *Plos One* 8: e53957

Manichanh C, Rigottier-Gois L, Bonnaud E, Gloux K, Pelletier E, Frangeul L, Nalin R, Jarrin C, Chardon P, Marteau P (2006) Reduced diversity of faecal microbiota in Crohn's disease revealed by a metagenomic approach. *Gut* 55: 205-211

Manley R, Boots M, Wilfert L (2015) Emerging viral disease risk to pollinating insects: ecological, evolutionary and anthropogenic factors. *J Appl Ecol* 52: 331-340

Manrique P, Bolduc B, Walk ST, van der Oost J, de Vos WM, Young MJ (2016) Healthy human gut phageome. *Proceedings of the National Academy of Sciences of the United States of America* 113: 10400-5

Manrique P, Dills M, Young MJ (2017a) The Human Gut Phage Community and Its Implications for Health and Disease. *Viruses* 9

Manrique P, Dills M, Young MJ (2017b) The Human Gut Phage Community and Its Implications for Health and Disease. *Viruses* 9: 141

Marcais G, Kingsford C (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* 27: 764-70

Marino-Ramirez L, Spouge JL, Kanga GC, Landsman D (2004) Statistical analysis of over-represented words in human promoter sequences. *Nucleic Acids Res* 32: 949-58

Markets M (2016) *Bioinformatics Market by Sector (Molecular Medicine, Agriculture, Forensic, Animal, Research & Gene Therapy), Product (Sequencing Platforms, Knowledge Management &*

Data Analysis) & Application (Genomics, Proteomics & Metabolomics) - Global Forecast to 2021. In Market Reports Hub:

Marks PA, Xu WS (2009) Histone Deacetylase Inhibitors: Potential in Cancer Therapy. *J Cell Biochem* 107: 600-608

Marraffini LA, Sontheimer EJ (2010) Self versus non-self discrimination during CRISPR RNA-directed immunity. *Nature* 463: 568-U194

Marti E, Variatza E, Balcazar JL (2014) Bacteriophages as a reservoir of extended-spectrum beta-lactamase and fluoroquinolone resistance genes in the environment. *Clin Microbiol Infec* 20: O456-O459

Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. 2011 17

Martinez-Alcantara A, Ballesteros E, Feng C, Rojas M, Koshinsky H, Fofanov VY, Havlak P, Fofanov Y (2009) PIQA: pipeline for Illumina G1 genome analyzer data quality assessment. *Bioinformatics* 25: 2438-9

Masoudi-Nejad A, Tonomura K, Kawashima S, Moriya Y, Suzuki M, Itoh M, Kanehisa M, Endo T, Goto S (2006) EGAssembler: online bioinformatics service for large-scale processing, clustering and assembling ESTs and genomic DNA fragments. *Nucleic Acids Research* 34: W459-W462

Masuda H, Tan Q, Awano N, Wu KP, Inouye M (2012) YeeU enhances the bundling of cytoskeletal polymers of MreB and FtsZ, antagonizing the CbtA (YeeV) toxicity in *Escherichia coli*. *Mol Microbiol* 84: 979-89

Matsushiro A, Sato K, Miyamoto H, Yamamura T, Honda T (1999) Induction of prophages of enterohemorrhagic *Escherichia coli* O157 : H7 with norfloxacin. *J Bacteriol* 181: 2257-2260

Maxwell EJ, Chen DD (2008) Twenty years of interface development for capillary electrophoresis-electrospray ionization-mass spectrometry. *Anal Chim Acta* 627: 25-33

McDonnell G, Russell AD (1999) Antiseptics and disinfectants: activity, action, and resistance. *Clin Microbiol Rev* 12: 147-79

McGrath S, Seegers JF, Fitzgerald GF, van Sinderen D (1999) Molecular characterization of a phage-encoded resistance system in *Lactococcus lactis*. *Applied and environmental microbiology* 65: 1891-9

McKinney W (2010) Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pp 51-56. Austin, TX

McMurtry VE, Gupta RW, Tran L, Blanchard E, Penn D, Taylor CM, Ferris MJ (2015) Bacterial diversity and *Clostridia* abundance decrease with increasing severity of necrotizing enterocolitis. *Microbiome* 3: 11

Medina E, Wiczorek D, Medina EM, Yang Q, Feiss M, Catalano CE (2010) Assembly and Maturation of the Bacteriophage Lambda Procapsid: gpC Is the Viral Protease. *J Mol Biol* 401: 813-830

Melsted P, Pritchard JK (2011) Efficient counting of k-mers in DNA sequences using a bloom filter. *Bmc Bioinformatics* 12

Micha R, Khatibzadeh S, Shi P, Andrews KG, Engell RE, Mozaffarian D, Global Burden of Diseases N, Chronic Diseases Expert G (2015) Global, regional and national consumption of major food groups in 1990 and 2010: a systematic analysis including 266 country-specific nutrition surveys worldwide. *BMJ Open* 5: e008705

Mikheenko A, Saveliev V, Gurevich A (2016) MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* 32: 1088-90

Miller ES, Heidelberg JF, Eisen JA, Nelson WC, Durkin AS, Ciecko A, Feldblyum TV, White O, Paulsen IT, Nierman WC, Lee J, Szczypinski B, Fraser CM (2003) Complete genome sequence of the broad-host-range vibriophage KVP40: Comparative genomics of a T4-related bacteriophage. *J Bacteriol* 185: 5220-5233

Miller S, Krijnse-Locker J (2008) Modification of intracellular membrane structures for virus replication. *Nature Reviews Microbiology* 6: 363-374

Miller TL, Wolin MJ (1979) Fermentations by saccharolytic intestinal bacteria. *Am J Clin Nutr* 32: 164-72

Mirzaei MK, Maurice CF (2017) Menage a trois in the human gut: interactions between host, bacteria and phages. *Nat Rev Microbiol* 15: 397-408

Moelling K (2013) What contemporary viruses tell us about evolution: a personal view. *Arch Virol* 158: 1833-1848

Mohawk KL, O'Brien AD (2011) Mouse models of Escherichia coli O157:H7 infection and shiga toxin injection. *J Biomed Biotechnol* 2011: 258185

Mokili JL, Rohwer F, Dutilh BE (2012) Metagenomics and future perspectives in virus discovery. *Curr Opin Virol* 2: 63-77

Molin S, Tolker-Nielsen T (2003) Gene transfer occurs with enhanced efficiency in biofilms and induces enhanced stabilisation of the biofilm structure. *Curr Opin Biotech* 14: 255-261

Molineux IJ, Panja D (2013) Popping the cork: mechanisms of phage genome ejection. *Nat Rev Microbiol* 11: 194-204

Monton MR, Soga T (2007) Metabolome analysis by capillary electrophoresis-mass spectrometry. *J Chromatogr A* 1168: 237-46; discussion 236

Mora A, Herrera A, Lopez C, Dahbi G, Mamani R, Pita JM, Alonso MP, Llovo J, Bernardez MI, Blanco JE, Blanco M, Blanco J (2011) Characteristics of the Shiga-toxin-producing enteroaggregative Escherichia coli O104:H4 German outbreak strain and of STEC strains isolated in Spain. *Int Microbiol* 14: 121-41

Moreau R, He B (2017) cAMP-dependent and independent mitigation of mTORC1-driven lipogenesis by short-chain fatty acids, R-alpha-lipoic acid and 4-phenylbutyric acid. *Faseb J* 31

Moreno F, Wandersman C (1980) Ompc and Lamb Proteins Can Serve as Substitute Receptors for Host Range Mutants of Coliphage Tuia. *J Bacteriol* 144: 1182-1185

Moriya Y, Itoh M, Okuda S, Yoshizawa AC, Kanehisa M (2007) KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Research* 35: W182-W185

Morona R, Klose M, Henning U (1984) Escherichia-Coli K-12 Outer-Membrane Protein (Ompa) as a Bacteriophage Receptor - Analysis of Mutant-Genes Expressing Altered Proteins. *J Bacteriol* 159: 570-578

Morowitz MJ, Poroyko V, Caplan M, Alverdy J, Liu DC (2010) Redefining the role of intestinal microbes in the pathogenesis of necrotizing enterocolitis. *Pediatrics* 125: 777-85

Morrison LA, Sidman RL, Fields BN (1991) Direct Spread of Reovirus from the Intestinal Lumen to the Central-Nervous-System through Vagal Autonomic Nerve-Fibers. *Proceedings of the National Academy of Sciences of the United States of America* 88: 3852-3856

Morrow AL, Lagomarcino AJ, Schibler KR, Taft DH, Yu Z, Wang B (2013a) Early microbial and metabolomic signatures predict later onset of necrotizing enterocolitis in preterm infants. *Microbiome* 1

Morrow AL, Lagomarcino AJ, Schibler KR, Taft DH, Yu Z, Wang B, Altaye M, Wagner M, Gevers D, Ward DV, Kennedy MA, Huttenhower C, Newburg DS (2013b) Early microbial and metabolomic signatures predict later onset of necrotizing enterocolitis in preterm infants. *Microbiome* 1: 13

Moss B (2013) Poxvirus DNA replication. *Cold Spring Harbor perspectives in biology* 5

Mshvildadze M, Neu J, Shuster J, Theriaque D, Li N, Mai V (2010) Intestinal Microbial Ecology in Premature Infants Assessed with Non-Culture-Based Techniques. *The Journal of pediatrics* 156: 20-25

Muhire BM, Golden M, Murrell B, Lefeuvre P, Lett JM, Gray A, Poon AYP, Ngandu NK, Semegni Y, Tanov EP, Monjane AL, Harkins GW, Varsani A, Shepherd DN, Martin DP (2014) Evidence of

Pervasive Biologically Functional Secondary Structures within the Genomes of Eukaryotic Single-Stranded DNA Viruses. *Journal of Virology* 88: 1972-1989

Muir P, Li S, Lou S, Wang D, Spakowicz DJ, Salichos L, Zhang J, Weinstock GM, Isaacs F, Rozowsky J, Gerstein M (2016) The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biol* 17: 53

Muniesa M, Hammerl JA, Hertwig S, Appel B, Brussow H (2012) Shiga Toxin-Producing *Escherichia coli* O104:H4: a New Challenge for Microbiology. *Applied and environmental microbiology* 78: 4065-4073

Munita JM, Arias CA (2016) Mechanisms of Antibiotic Resistance. *Microbiology spectrum* 4: 10.1128/microbiolspec.VMBF-0016-2015

Munz C, Lunemann JD, Getts MT, Miller SD (2009) Antiviral immune responses: triggers of or triggered by autoimmunity? *Nat Rev Immunol* 9: 246-58

Murphy EF, Cotter PD, Hogan A, O'Sullivan O, Joyce A, Fouhy F, Clarke SF, Marques TM, O'Toole PW, Stanton C, Quigley EM, Daly C, Ross PR, O'Doherty RM, Shanahan F (2013) Divergent metabolic outcomes arising from targeted manipulation of the gut microbiota in diet-induced obesity. *Gut* 62: 220-6

Musich S, MacLeod S, Bhattarai GR, Wang SS, Hawkins K, Bottone FG, Jr., Yeh CS (2016) The Impact of Obesity on Health Care Utilization and Expenditures in a Medicare Supplement Population. *Gerontol Geriatr Med* 2: 2333721415622004

Musso G, Gambino R, Cassader M (2010) Obesity, diabetes, and gut microbiota: the hygiene hypothesis expanded? *Diabetes Care* 33: 2277-84

Muyzer G, de Waal EC, Uitterlinden AG (1993) Profiling of complex microbial populations by denaturing gradient gel electrophoresis analysis of polymerase chain reaction-amplified genes coding for 16S rRNA. *Applied and environmental microbiology* 59: 695-700

Nakao H, Kiyokawa N, Fujimoto J, Yamasaki S, Takeda T (1999) Monoclonal antibody to Shiga toxin 2 which blocks receptor binding and neutralizes cytotoxicity. *Infection and Immunity* 67: 5717-5722

Needham BD, Trent MS (2013) Fortifying the barrier: the impact of lipid A remodelling on bacterial pathogenesis. *Nat Rev Microbiol* 11: 467-481

Nejman-Falencyk B, Bloch S, Licznarska K, Dydecka A, Felczykowska A, Topka G, Wegrzyn A, Wegrzyn G (2015) A small, microRNA-size, ribonucleic acid regulating gene expression and development of Shiga toxin-converting bacteriophage Phi24Beta. *Scientific reports* 5: 10080

Neu J, Rushing J (2011) Cesarean versus vaginal delivery: long-term infant outcomes and the hygiene hypothesis. *Clin Perinatol* 38: 321-31

Neu J, Walker WA (2011) Necrotizing enterocolitis. *The New England journal of medicine* 364: 255-64

Newland JW, Strockbine NA, Miller SF, O'Brien AD, Holmes RK (1985) Cloning of Shiga-like toxin structural genes from a toxin converting phage of *Escherichia coli*. *Science* 230: 179-81

Nguyen S, Baker K, Padman BS, Patwa R, Dunstan RA, Weston TA, Schlosser K, Bailey B, Lithgow T, Lazarou M, Luque A, Rohwer F, Blumberg RS, Barr JJ (2017) Bacteriophage Transcytosis Provides a Mechanism To Cross Epithelial Cell Layers. *mBio* 8

Ni B, Ghosh B, Paldy FS, Colin R, Heimerl T, Sourjik V (2017) Evolutionary Remodeling of Bacterial Motility Checkpoint Control. *Cell Rep* 18: 866-877

Nie YF, Hu J, Yan XH (2015) Cross-talk between bile acids and intestinal microbiota in host metabolism and health. *J Zhejiang Univ-Sc B* 16: 436-446

Nieuwdorp M., Gijljamse. P.W., Pai N., Kaplan L. M. (2014) Role of the Microbiome in Energy Regulation and Metabolism. *The Gut Microbiome and Disease*. 146: 1525-1533

Nikaido H (2001) Preventing drug access to targets: cell surface permeability barriers and active efflux in bacteria. *Semin Cell Dev Biol* 12: 215-223

Nikolaidis I, Favini-Stabile S, Dessen A (2014) Resistance to antibiotics targeted to the bacterial cell wall. *Protein Sci* 23: 243-59

Ning Z, Cox AJ, Mullikin JC (2001) SSAHA: a fast search method for large DNA databases. *Genome Res* 11: 1725-9

Nishimura I, Kurokawa M, Liu L, Ying BW (2017) Coordinated Changes in Mutation and Growth Rates Induced by Genome Reduction. *MBio* 8

Nonga HE, Muhairwa AP (2010) Prevalence and antibiotic susceptibility of thermophilic *Campylobacter* isolates from free range domestic duck (*Cairina moschata*) in Morogoro municipality, Tanzania. *Tropical animal health and production* 42: 165-72

Norman JM, Handley SA, Baldrige MT, Droit L, Liu CY, Keller BC, Kambal A, Monaco CL, Zhao G, Fleshner P, Stappenbeck TS, McGovern DP, Keshavarzian A, Mutlu EA, Sauk J, Gevers D, Xavier RJ, Wang D, Parkes M, Virgin HW (2015) Disease-specific alterations in the enteric virome in inflammatory bowel disease. *Cell* 160: 447-60

Nyambe S, Burgess C, Whyte P, Bolton D (2016) Survival studies of a temperate and lytic bacteriophage in bovine faeces and slurry. *J Appl Microbiol*

O'Brien AD, Chen ME, Holmes RK, Kaper J, Levine MM (1984) Environmental and human isolates of *Vibrio cholerae* and *Vibrio parahaemolyticus* produce a *Shigella dysenteriae* 1 (Shiga)-like cytotoxin. *Lancet* 1: 77-8

O'Brien A, D., and Kaper, J, B. (1998) *Escherichia coli* O157:H7 and other Shiga toxin-producing *E. coli* strains.

Ogilvie LA, Jones BV (2017) The human gut virome: form and function. *Emerging Topics in Life Sciences* 1: 351-362

Oh C, Lee K, Cheong Y, Lee SW, Park SY, Song CS, Choi IS, Lee JB (2015) Comparison of the Oral Microbiomes of Canines and Their Owners Using Next-Generation Sequencing. *Plos One* 10

Ohnishi M, Terajima J, Kurokawa K, Nakayama K, Murata T, Tamura K, Ogura Y, Watanabe H, Hayashi T (2002) Genomic diversity of enterohemorrhagic *Escherichia coli* O157 revealed by whole genome PCR scanning. *Proceedings of the National Academy of Sciences of the United States of America* 99: 17043-8

Olesky M, Zhao SQ, Rosenberg RL, Nicholas RA (2006) Porin-mediated antibiotic resistance in *Neisseria gonorrhoeae*: Ion, solute, and antibiotic permeation through PIB proteins with penB mutations. *J Bacteriol* 188: 2300-2308

Oliver KM, Degan PH, Hunter MS, Moran NA (2009) Bacteriophages encode factors required for protection in a symbiotic mutualism. *Science* 325: 992-4

Orbitrap P (2018) Q Exactive. In

Osto M, Abegg K, Bueter M, le Roux CW, Cani PD, Lutz TA (2013) Roux-en-Y gastric bypass surgery in rats alters gut microbiota profile along the intestine. *Physiol Behav* 119: 92-6

Ounit R, Wanamaker S, Close TJ, Lonardi S (2015) CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC genomics* 16

Ozdam T, Sela DA, Xiao JB, Boyacioglu D, Chen F, Capanoglu E (2016) The Reciprocal Interactions between Polyphenols and Gut Microbiota and Effects on Bioaccessibility. *Nutrients* 8

Pacton M, Wacey D, Corinaldesi C, Tangherlini M, Kilburn MR, Gorin GE, Danovaro R, Vasconcelos C (2014) Viruses as new agents of organomineralization in the geological record. *Nature communications* 5

Palm NW, de Zoete MR, Cullen TW, Barry NA, Stefanowski J, Hao LM, Degan PH, Hu JZ, Peter I, Zhang W, Ruggiero E, Cho JH, Goodman AL, Flavell RA (2014) Immunoglobulin A Coating Identifies Colitogenic Bacteria in Inflammatory Bowel Disease. *Cell* 158: 1000-1010

Palmer KL, Kos VN, Gilmore MS (2010) Horizontal gene transfer and the genomics of enterococcal antibiotic resistance. *Curr Opin Microbiol* 13: 632-639

Pammi M, Cope J, Tarr PI, Warner BB, Morrow AL, Mai V, Gregory KE, Kroll JS, McMurtry V, Ferris MJ, Engstrand L, Lilja HE, Hollister EB, Versalovic J, Neu J (2017) Intestinal dysbiosis in preterm infants preceding necrotizing enterocolitis: a systematic review and meta-analysis. *Microbiome* 5: 31

Pan ZZ, Raftery D (2007) Comparing and combining NMR spectroscopy and mass spectrometry in metabolomics. *Anal Bioanal Chem* 387: 525-527

Paramsothy S, Paramsothy R, Rubin DT, Kamm MA, Kaakoush NO, Mitchell HM, Castano-Rodriguez N (2017) Faecal Microbiota Transplantation for Inflammatory Bowel Disease: A Systematic Review and Meta-analysis. *J Crohns Colitis* 11: 1180-1199

Parent KN, Erb ML, Cardone G, Nguyen K, Gilcrease EB, Porcek NB, Pogliano J, Baker TS, Casjens SR (2014) OmpA and OmpC are critical host factors for bacteriophage Sf6 entry in *Shigella*. *Mol Microbiol* 92: 47-60

Parsley LC, Consuegra EJ, Kakirde KS, Land AM, Harper WF, Jr., Liles MR (2010) Identification of diverse antimicrobial resistance determinants carried on bacterial, plasmid, or viral metagenomes from an activated sludge microbial assemblage. *Applied and environmental microbiology* 76: 3753-7

Parsons JB, Rock C (2013) Bacterial lipids: Metabolism and membrane homeostasis. *Prog Lipid Res* 52: 249-276

Paton A, W., and Paton, J, C. (1996) *Enterobacter cloacae* producing a Shigalike toxin II-related cytotoxin associated with a case of hemolytic-uremic syndrome. *Clinical microbiology* 34: 463-465

Payne S, Gibson G, Wynne A, Hudspith B, Brostoff J, Tuohy K (2003) In vitro studies on colonization resistance of the human gut microbiota to *Candida albicans* and the effects of tetracycline and *Lactobacillus plantarum* LPK. *Curr Issues Intest Microbiol* 4: 1-8

Pearson ES (1927) The Application of the Theory of Differential Equations to the Solution of Problems Connected with the Interdependence of Species. *Biometrika* 19: 216-222

Pedulla ML, Ford ME, Houtz JM, Karthikeyan T, Wadsworth C, Lewis JA, Jacobs-Sera D, Falbo J, Gross J, Pannunzio NR, Brucker W, Kumar V, Kandasamy J, Keenan L, Bardarov S, Kriakov J, Lawrence JG, Jacobs WR, Hendrix RW, Hatfull GF (2003) Origins of highly mosaic mycobacteriophage genomes. *Cell* 113: 171-182

Peng Y, Leung HCM, Yiu SM, Chin FYL (2010) IDBA - A Practical Iterative de Bruijn Graph De Novo Assembler. *Lect N Bioinform* 6044: 426-440

Perez-Cobas AE, Gosalbes MJ, Friedrichs A, Knecht H, Artacho A, Eismann K, Otto W, Rojo D, Bargiela R, von Bergen M, Neuling SC, Daumer C, Heinsen FA, Latorre A, Barbas C, Seifert J, dos Santos VM, Ott SJ, Ferrer M, Moya A (2013) Gut microbiota disturbance during antibiotic therapy: a multi-omic approach. *Gut* 62: 1591-1601

Perucchetti R, Parris W, Becker A, Gold M (1988) Late stages in bacteriophage lambda head morphogenesis: in vitro studies on the action of the bacteriophage lambda D-gene and W-gene products. *Virology* 165: 103-114

Petatan-Sagahon I, Anducho-Reyes MA, Silva-Rojas HV, Arana-Cuenca A, Tellez-Jurado A, Cardenas-Alvarez IO, Mercado-Flores Y (2011) Isolation of Bacteria with Antifungal Activity against the Phytopathogenic Fungi *Stenocarpella maydis* and *Stenocarpella macrospora*. *Int J Mol Sci* 12: 5522-5537

Philippe, N., Legendre, M., Dautre, G., Couté, Y., Poirot, O., Lescot, M., et al. (2013). Pandoraviruses: amoeba viruses with genomes Up to 2.5 Mb reaching that of parasitic eukaryotes. *Science*. 341: 281–286.

Pietilä.M.K., Demina.T.A., Atanasova N.S., Oksanen H.M., Bamford D.H. (2014) Archaeal viruses and bacteriophages: comparisons and contrasts. *Trends in Microbiology* 22: 334-334

Poole K (2012) Bacterial stress responses as determinants of antimicrobial resistance. *J Antimicrob Chemoth* 67: 2069-2089

Porter JR, Pelczar MJ (1941) The Nutrition of *Staphylococcus aureus*: The Influence of Biotin, Biotinase and Vitamin H on the Growth of Several Strains. *J Bacteriol* 41: 173-192

Portune KJ, Beaumont M, Davila AM, Tome D, Blachier F, Sanz Y (2016) Gut microbiota role in dietary protein metabolism and health-related outcomes: The two sides of the coin. *Trends Food Sci Tech* 57: 213-232

Possemiers S, Bolca S, Verstraete W, Heyerick A (2011) The intestinal microbiome: a separate organ inside the body with the metabolic potential to influence the bioactivity of botanicals. *Fitoterapia* 82: 53-66

POURCEL C (2017) CRISPRdb. In POURCEL C (ed) University of Paris

Ptashne M, Jeffrey A, Johnson AD, Maurer R, Meyer BJ, Pabo CO, Roberts TM, Sauer RT (1980) How the Lambda-Repressor and Cro Work. *Cell* 19: 1-11

Qian T, Zhang R, Zhu L, Shi P, Yang J, Yang CY, Chen DM, Shi JY, Zhou XG, Qiu YP, Yang Y, He L, He SR, Cao YT, Wei QF, Kumar M, Chen C, Chinese Collaborative Study Group for Neonatal Necrotizing E (2017) Necrotizing enterocolitis in low birth weight infants in China: Mortality risk factors expressed by birth weight categories. *Pediatr Neonatol* 58: 509-515

Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glockner FO (2013) The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res* 41: D590-6

Raffaelli A, Saba A (2003) Atmospheric pressure photoionization mass spectrometry. *Mass Spectrom Rev* 22: 318-331

Raisanen L, Schubert K, Jaakonsaari T, Alatosava T (2004) Characterization of lipoteichoic acids as *Lactobacillus delbrueckii* phage receptor components. *J Bacteriol* 186: 5529-5532

Rajagopala SV, Casjens S, Uetz P (2011) The protein interaction map of bacteriophage lambda. *Bmc Microbiol* 11

Rajilic-Stojanovic M, de Vos WM (2014) The first 1000 cultured species of the human gastrointestinal microbiota. *FEMS Microbiol Rev* 38: 996-1047

Ramirez F, Ryan DP, Gruning B, Bhardwaj V, Kilpert F, Richter AS, Heyne S, Dundar F, Manke T (2016) deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res* 44: W160-5

Rampelli S, Soverini M, Turrone S, Quercia S, Biagi E, Brigidi P, Candela M (2016) ViromeScan: a new tool for metagenomic viral community profiling. *BMC genomics* 17: 165

Raybould HE (2010) Gut chemosensing: Interactions between gut endocrine cells and visceral afferents. *Auton Neurosci-Basic* 153: 41-46

Reeve JN, Shaw JE (1979) Lambda encodes an outer membrane protein: the lom gene. *Molecular & general genetics* : MGG 172: 243-8

Rehakova Z, Capkova J, Stepankova R, Sinkora J, Louzecka A, Ivanyi P, Weinreich S (2000) Germ-free mice do not develop ankylosing enthesopathy, a spontaneous joint disease. *Human immunology* 61: 555-8

Reichardt LF (1975) Control of bacteriophage lambda repressor synthesis: regulation of the maintenance pathway of the cro and cI products. *J Mol Biol* 93: 289-309

Reyes A, Blanton LV, Cao S, Zhao G, Manary M, Trehan I, Smith MI, Wang D, Virgin HW, Rohwer F, Gordon JI (2015) Gut DNA viromes of Malawian twins discordant for severe acute malnutrition. *Proceedings of the National Academy of Sciences of the United States of America* 112: 11941-6

Reyes A, Haynes M, Hanson N, Angly FE, Heath AC, Rohwer F, Gordon JI (2010) Viruses in the faecal microbiota of monozygotic twins and their mothers. *Nature* 466: 334-U81

Reyes A, Semenkovich NP, Whiteson K, Rohwer F, Gordon JI (2012) Going viral: next-generation sequencing applied to phage populations in the human gut. *Nat Rev Microbiol* 10: 607-17

Richardson EJ, Watson M (2013) The automatic annotation of bacterial genomes. *Brief Bioinform* 14: 1-12

Riley LW, Remis RS, Helgerson SD, McGee HB, Wells JG, Davis BR, Hebert RJ, Olcott ES, Johnson LM, Hargrett NT, Blake PA, Cohen ML (1983) Hemorrhagic Colitis Associated with a Rare *Escherichia coli* Serotype. *New England Journal of Medicine* 308: 681-685

Riva A, Borgo F, Lassandro C, Verduci E, Morace G, Borghi E, Berry D (2016) Pediatric Obesity Is Associated with an Altered Gut Microbiota and Discordant Shifts in Firmicutes Populations. *Digest Liver Dis* 48: E268-E268

Rizk G, Lavenier D, Chikhi R (2013) DSK: k-mer counting with very low memory usage. *Bioinformatics* 29: 652-653

Roa M (1979) Interaction of bacteriophage K10 with its receptor, the lamB protein of *Escherichia coli*. *J Bacteriol* 140: 680-6

Robertson DM, Paganelli R, Dinwiddie R, Levinsky RJ (1982) Milk antigen absorption in the preterm and term neonate. *Arch Dis Child* 57: 369-72

Roberts RC, Strom AR, Helinski DR (1994) The Parde Operon of the Broad-Host-Range Plasmid Rk2 Specifies Growth-Inhibition Associated with Plasmid Loss. *J Mol Biol* 237: 35-51

Robinson CM, Jesudhasan PR, Pfeiffer JK (2014) Bacterial lipopolysaccharide binding enhances virion stability and promotes environmental fitness of an enteric virus. *Cell Host Microbe* 15: 36-46

Rodriguez-Brito B, Li LL, Wegley L, Furlan M, Angly F, Breitbart M, Buchanan J, Desnues C, Dinsdale E, Edwards R, Felts B, Haynes M, Liu H, Lipson D, Mahaffy J, Martin-Cuadrado AB, Mira A, Nulton J, Pasic L, Rayhawk S et al. (2010) Viral and microbial community dynamics in four aquatic environments. *Isme J* 4: 739-751

Roediger WE (1980) Role of anaerobic bacteria in the metabolic welfare of the colonic mucosa in man. *Gut* 21: 793-8

Roesch LF, Casella G, Simell O, Krischer J, Wasserfall CH, Schatz D, Atkinson MA, Neu J, Triplett EW (2009) Influence of fecal sample storage on bacterial community diversity. *Open Microbiol J* 3: 40-6

Rogers GB, Marsh P, Stressmann AF, Allen CE, Daniels TVW, Carroll MP, Bruce KD (2010) The exclusion of dead bacterial cells is essential for accurate molecular analysis of clinical samples. *Clin Microbiol Infec* 16: 1656-1658

Rogers Y-H, Venter JC (2005) Massively parallel sequencing. *Nature* 437: 326

Rohwer F, Thurber RV (2009) Viruses manipulate the marine environment. *Nature* 459: 207-212

Rooks MG, Garrett WS (2016) Gut microbiota, metabolites and host immunity. *Nat Rev Immunol* 16: 341-352

Rosario K, Duffy S, Breitbart M (2012) A field guide to eukaryotic circular single-stranded DNA viruses: insights gained from metagenomics. *Arch Virol* 157: 1851-1871

Round JL, Mazmanian SK (2009) The gut microbiota shapes intestinal immune responses during health and disease (vol 9, pg 313, 2009). *Nat Rev Immunol* 9: 600-600

Roux S, Enault F, Bronner G, Vaultot D, Forterre P, Krupovic M (2013) Chimeric viruses blur the borders between the major groups of eukaryotic single-stranded DNA viruses. *Nature Communications* 4

Rowland I, Gibson G, Heinken A, Scott K, Swann J, Thiele I, Tuohy K (2018) Gut microbiota functions: metabolism of nutrients and other food components. *Eur J Nutr* 57: 1-24

Roy AM, Coleman J (1994) Mutations in *firA*, encoding the second acyltransferase in lipopolysaccharide biosynthesis, affect multiple steps in lipopolysaccharide biosynthesis. *J Bacteriol* 176: 1639-46

Roy RS, Bhattacharya D, Schliep A (2014) Turtle: identifying frequent k-mers with cache-efficient algorithms. *Bioinformatics* 30: 1950-7

Rubin BE, Gibbons SM, Kennedy S, Hampton-Marcell J, Owens S, Gilbert JA (2013) Investigating the impact of storage conditions on microbial community composition in soil samples. *Plos One* 8: e70460

Rubinchik S, Parris W, Gold M (1994) The in-Vitro Atpases of Bacteriophage-Lambda Terminase and Its Large Subunit, Gene-Product-a - the Relationship with Their DNA Helicase and Packaging Activities. *J Biol Chem* 269: 13586-13593

Ruby JG, Bellare P, Derisi JL (2013) PRICE: software for the targeted assembly of components of (Meta) genomic sequence data. *G3 (Bethesda)* 3: 865-80

Ruud J, Wilhelms DB, Nilsson A, Eskilsson A, Tang YJ, Strohle P, Caesar R, Schwaninger M, Wunderlich T, Backhed F, Engblom D, Blomqvist A (2013) Inflammation- and tumor-induced anorexia and weight loss require MyD88 in hematopoietic/myeloid cells but not in brain endothelial or neural cells. *Faseb J* 27: 1973-1980

Rybicki EP (2015) A Top Ten list for economically important plant viruses. *Arch Virol* 160: 17-20

Sadygov RG, Maroto FM, Huhmer AFR (2006) ChromAlign: A two-step algorithmic procedure for time alignment of three-dimensional LC-MS chromatographic surfaces. *Anal Chem* 78: 8207-8217

Saile N, Voigt A, Kessler S, Stressler T, Klumpp J, Fischer L, Schmidt H (2016) Escherichia coli O157:H7 Strain EDL933 Harbors Multiple Functional Prophage-Associated Genes Necessary for the Utilization of 5-N-Acetyl-9-O-Acetyl Neuraminic Acid as a Growth Substrate. *Applied and environmental microbiology* 82: 5940-50

Sakaguchi Y, Hayashi T, Kurokawa K, Nakayama K, Oshima K, Fujinaga Y, Ohnishi M, Ohtsubo E, Hattori M, Oguma K (2005) The genome sequence of Clostridium botulinum type C neurotoxin-

converting phage and the molecular mechanisms of unstable lysogeny. *Proceedings of the National Academy of Sciences of the United States of America* 102: 17472-7

Sako T, Sawaki S, Sakurai T, Ito S, Yoshizawa Y, Kondo I (1983) Cloning and expression of the staphylokinase gene of *Staphylococcus aureus* in *Escherichia coli*. *Molecular & general genetics* : MGG 190: 271-7

Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, Marcais G, Pop M, Yorke JA (2012) GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res* 22: 557-67

Sanders ME, Guarner F, Guerrant R, Holt PR, Quigley EMM, Sartor RB, Sherman PM, Mayer EA (2013) An update on the use and investigation of probiotics in health and disease. *Gut* 62: 787-796

Sanger F, Coulson AR (1975) A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J Mol Biol* 94: 441-8

Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* 74: 5463-7

Sao-Jose C, de Frutos M, Raspaud E, Santos MA, Tavares P (2007) Pressure built by DNA packing inside virions: Enough to drive DNA ejection in vitro, largely insufficient for delivery into the bacterial cytoplasm. *Journal of Molecular Biology* 374: 346-355

Sassone-Corsi M, Nuccio SP, Liu H, Hernandez D, Vu CT, Takahashi AA, Edwards RA, Raffatellu M (2016) Microcins mediate competition among *Enterobacteriaceae* in the inflamed gut. *Nature* 540: 280-+

Scarpellini E, Campanale M, Leone D, Purchiaroni F, Vitale G, Lauritano EC, Gasbarrini A (2010) Gut microbiota and obesity. *Intern Emerg Med* 5 Suppl 1: S53-6

Scarpellini E, Ianiro G, Attili F, Bassanelli C, De Santis A, Gasbarrini A (2015) The human gut microbiota and virome: Potential therapeutic implications. *Dig Liver Dis* 47: 1007-12

Schauber J, Svanholm C, Termen S, Iffland K, Menzel T, Scheppach W, Melcher R, Agerberth B, Luhrs H, Gudmundsson GH (2003) Expression of the cathelicidin LL-37 is modulated by short chain fatty acids in colonocytes: relevance of signalling pathways. *Gut* 52: 735-741

Schloissnig S, Arumugam M, Sunagawa S, Mitreva M, Tap J, Zhu A, Waller A, Mende DR, Kultima JR, Martin J, Kota K, Sunyaev SR, Weinstock GM, Bork P (2013) Genomic variation landscape of the human gut microbiome. *Nature* 493: 45-50

Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Van Horn DJ, Weber CF (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology* 75: 7537-41

Schmidt H, Montag, M., Bockemuhl, J., Heeseman, J. and Karch, H. (1993) Shiga-like toxin II related cytotoxins in *Citrobacter freundii* strains from human and beef samples. *Infection and immunity* 61: 534-543

Schoch CL, Seifert KA, Huhndorf S, Robert V, Spouge JL, Levesque CA, Chen W, Bolchacova E, Voigt K, Crous PW, Miller AN, Wingfield MJ, Aime MC, An KD, Bai FY, Barreto RW, Begerow D, Bergeron MJ, Blackwell M, Boekhout T et al. (2012) Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi. *P Natl Acad Sci USA* 109: 6241-6246

Schubert RA, Dodd IB, Egan JB, Shearwin KE (2007) Cro's role in the CI-Cro bistable switch is critical for lambda's transition from lysogeny to lytic development. *Gene Dev* 21: 2461-2472

Schuler GD, Epstein JA, Ohkawa H, Kans JA (1996) Entrez: molecular biology database and retrieval system. *Methods Enzymol* 266: 141-62

Schwabe RF, Jobin C (2013) The microbiome and cancer. *Nat Rev Cancer* 13: 800-12

Schwartz M (1975) Reversible Interaction between Coliphage Lambda and Its Receptor Protein. *Journal of Molecular Biology* 99: 185-201

Scigelova M, Makarov A (2006) Orbitrap mass analyzer - Overview and applications in proteomics. *Proteomics*: 16-21

Scott D, Ely B (2015) Comparison of genome sequencing technology and assembly methods for the analysis of a GC-rich bacterial genome. *Curr Microbiol* 70: 338-44

Sears CL, Garrett WS (2014) Microbes, microbiota, and colon cancer. *Cell Host Microbe* 15: 317-28

Seemann T (2014) Prokka: rapid prokaryotic genome annotation. *Bioinformatics* 30: 2068-2069

Segata N, Waldron L, Ballarini A, Narasimhan V, Jousson O, Huttenhower C (2012) Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods* 9: 811-+

Sellon RK, Tonkonogy S, Schultz M, Dieleman LA, Grenther W, Balish E, Rennick DM, Sartor RB (1998) Resident enteric bacteria are necessary for development of spontaneous colitis and immune system activation in interleukin-10-deficient mice. *Infection and immunity* 66: 5224-31

Shaikh S, Fatima J, Shakil S, Rizvi SMD, Kamal MA (2015) Antibiotic resistance and extended spectrum beta-lactamases: Types, epidemiology and treatment. *Saudi J Biol Sci* 22: 90-101

Shimizu T, Ohta Y, Noda M (2009) Shiga Toxin 2 Is Specifically Released from Bacterial Cells by Two Different Mechanisms. *Infection and Immunity* 77: 2813-2823

Shin NR, Whon TW, Bae JW (2015) Proteobacteria: microbial signature of dysbiosis in gut microbiota. *Trends Biotechnol* 33: 496-503

Shin R, Suzuki M, Morishita Y (2002) Influence of intestinal anaerobes and organic acids on the growth of enterohaemorrhagic *Escherichia coli* O157:H7. *J Med Microbiol* 51: 201-6

Short BR, Vargas MA, Thomas JC, O'Hanlon S, Enright MC (2006) In vitro activity of a novel compound, the metal ion chelating agent AQ+, against clinical isolates of *Staphylococcus aureus*. *The Journal of antimicrobial chemotherapy* 57: 104-9

Shulla A, Randall G (2016) (+) RNA virus replication compartments: a safe home for (most) viral replication. *Curr Opin Microbiol* 32: 82-88

Silva JB, Storms Z, Sauvageau D (2016) Host receptors for bacteriophage adsorption. *Fems Microbiology Letters* 363

Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I (2009) ABySS: a parallel assembler for short read sequence data. *Genome Res* 19: 1117-23

Skorupski K, Tomaszewski J, Ruger W, Simon LD (1988) A Bacteriophage-T4 Gene Which Functions to Inhibit *Escherichia-Coli* Lon Protease. *J Bacteriol* 170: 3016-3024

Smith ASG, Rawlings DE (1998) Efficiency of the pTF-FC2 pas poison-antidote stability system in *Escherichia coli* is affected by the host strain, and antidote degradation requires the Lon protease. *J Bacteriol* 180: 5458-5462

Smith B, Bodé S, Skov TH, Mirsepasi H, Greisen G, Krogfelt KA (2012a) Investigation of the early intestinal microflora in premature infants with/without necrotizing enterocolitis using two different methods. *Pediatric research* 71

Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G (2006) XCMS: Processing mass spectrometry data for metabolite profiling using Nonlinear peak alignment, matching, and identification. *Anal Chem* 78: 779-787

Smith DL, James CE, Sergeant MJ, Yaxian Y, Saunders JR, McCarthy AJ, Allison HE (2007a) Short-tailed Stx phages exploit the conserved YaeT protein to disseminate Shiga Toxin genes among enterobacteria. *J Bacteriol* 189: 7223-7233

Smith DL, Rooks DJ, Fogg PC, Darby AC, Thomson NR, McCarthy AJ, Allison HE (2012b) Comparative genomics of Shiga toxin encoding bacteriophages. *BMC genomics* 13: 311

Smith DL, Wareing BM, Fogg PC, Riley LM, Spencer M, Cox MJ, Saunders JR, McCarthy AJ, Allison HE (2007b) Multilocus characterization scheme for shiga toxin-encoding bacteriophages. *Applied and environmental microbiology* 73: 8032-40

Snell EE, Mitchell HK (1941a) Purine and Pyrimidine as Growth Substances for Lactic Acid Bacteria. *Proceedings of the National Academy of Sciences of the United States of America* 27: 1-7

Snell EE, Mitchell HK (1941b) Purine and Pyrimidine as Growth Substances for Lactic Acid Bacteria. *Proceedings of the National Academy of Sciences* 27: 1

Soga T, Baran R, Suematsu M, Ueno Y, Ikeda S, Sakurakawa T, Kakazu Y, Ishikawa T, Robert M, Nishioka T, Tomita M (2006) Differential metabolomics reveals ophthalmic acid as an oxidative stress biomarker indicating hepatic glutathione consumption. *J Biol Chem* 281: 16768-16776

Soto SM (2013) Role of efflux pumps in the antibiotic resistance of bacteria embedded in a biofilm. *Virulence* 4: 223-229

Southam AD, Payne TG, Cooper HJ, Arvanitis TN, Viant MR (2007) Dynamic range and mass accuracy of wide-scan direct infusion nanoelectrospray Fourier transform ion cyclotron resonance mass spectrometry-based metabolomics increased by the spectral stitching method. *Anal Chem* 79: 4595-4602

Stedman K (2013) Mechanisms for RNA Capture by ssDNA Viruses: Grand Theft RNA. *J Mol Evol* 76: 359-364

Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai CX, Efron MJ, Iyer R, Schatz MC, Sinha S, Robinson GE (2015) Big Data: Astronomical or Genomical? *Plos Biol* 13

Stern A, Sorek R (2011) The phage-host arms race: shaping the evolution of microbes. *BioEssays : news and reviews in molecular, cellular and developmental biology* 33: 43-51

Steven AC, Trus BL, Maizel JV, Unser M, Parry DAD, Wall JS, Hainfeld JF, Studier FW (1988) Molecular Substructure of a Viral Receptor-Recognition Protein - the Gp17 Tail-Fiber of Bacteriophage-T7. *J Mol Biol* 200: 351-365

Stewart AC, Osborne B, Read TD (2009) DIYA: a bacterial annotation pipeline for any genomics lab. *Bioinformatics* 25: 962-963

Stewart CJ, Embleton ND, Marrs EC, Smith DP, Nelson A, Abdulkadir B, Skeath T, Petrosino JF, Perry JD, Berrington JE, Cummings SP (2016) Temporal bacterial and metabolic development of the preterm gut reveals specific signatures in health and disease. *Microbiome* 4: 67

Stewart CJ, Embleton ND, Marrs ECL, Smith DP, Fofanova T, Nelson A, Skeath T, Perry JD, Petrosino JF, Berrington JE, Cummings SP (2017) Longitudinal development of the gut microbiome and metabolome in preterm neonates with late onset sepsis and healthy controls. *Microbiome* 5: 75

Stewart CJ, Nelson A, Scribbins D, Marrs EC, Lanyon C, Perry JD, Embleton ND, Cummings SP, Berrington JE (2013a) Bacterial and fungal viability in the preterm gut: NEC and sepsis. *Arch Dis Child Fetal Neonatal Ed* 98: F298-303

Stewart CJ, Nelson A, Scribbins D, Marrs ECL, Perry JD, Embleton ND (2013b) Bacterial and fungal viability in the preterm gut: NEC and sepsis. *Arch Dis Child Fetal Neonatal Ed* 98

Stewart CJ, Skeath T, Nelson A, Fernstad SJ, Marrs EC, Perry JD, Cummings SP, Berrington JE, Embleton ND (2015) Preterm gut microbiota and metabolome following discharge from intensive care. *Scientific reports* 5: 17141

Strunk T, Currie A, Richmond P, Simmer K, Burgner D (2011a) Innate immunity in human newborn infants: prematurity means more than immaturity. *J Matern-Fetal Neo M* 24: 25-31

Strunk T, Currie A, Richmond P, Simmer K, Burgner D (2011b) Innate immunity in human newborn infants: prematurity means more than immaturity. *The Journal of Maternal-Fetal & Neonatal Medicine* 24: 25-31

- Su LK, Lu CP, Wang Y, Cao DM, Sun JH, Yan YX (2010) [Lysogenic infection of a Shiga toxin 2-converting bacteriophage changes host gene expression, enhances host acid resistance and motility]. *Molekuliarnaia biologii* 44: 60-73
- Sullivan MB, Coleman ML, Weigele P, Rohwer F, Chisholm SW (2005) Three *Prochlorococcus* cyanophage genomes: signature features and ecological interpretations. *PLoS Biol* 3: e144
- Sumi Y, Miyakawa M, Kanzaki M, Kotake Y (1977) Vitamin B-6 deficiency in germfree rats. *J Nutr* 107: 1707-14
- Sundin OH, Mendoza-Ladd A, Zeng MT, Diaz-Arevalo D, Morales E, Fagan BM, Ordonez J, Velez P, Antony N, McCallum RW (2017) The human jejunum has an endogenous microbiota that differs from those in the oral cavity and colon. *Bmc Microbiol* 17
- Suttle CA (2007a) Marine viruses--major players in the global ecosystem. *Nat Rev Microbiol* 5: 801-12
- Suttle CA (2007b) Marine viruses - major players in the global ecosystem. *Nature Reviews Microbiology* 5: 801-812
- Suzuki K, Komagata K (1983) Taxonomic Significance of Cellular Fatty-Acid Composition in Some *Coryneform* Bacteria. *Int J Syst Bacteriol* 33: 188-200
- Szathmary E, Demeter L (1987) Group Selection of Early Replicators and the Origin of Life. *J Theor Biol* 128: 463-486
- Szczuka E, Szumala-Kakol A, Siuda A, Kaznowski A (2010) Clonal analysis of *Staphylococcus aureus* strains isolated in obstetric-gynaecological hospital. *Pol J Microbiol* 59: 161-5
- Tagliabue A, Elli M (2013) The role of gut microbiota in human obesity: recent findings and future perspectives. *Nutr Metab Cardiovasc Dis* 23: 160-8
- Takeda Y, Matsubara K, Ogata K (1975) Regulation of early gene expression in bacteriophage lambda: effect of *toF* mutation on strand-specific transcriptions. *Virology* 65: 374-84

Takeuchi N, Hogeweg P (2012) Evolutionary dynamics of RNA-like replicator systems: A bioinformatic approach to the origin of life. *Physics of Life Reviews* 9: 219-263

Tamang MD, Sunwoo H, Jeon B (2017) Phage-mediated dissemination of virulence factors in pathogenic bacteria facilitated by antibiotic growth promoters in animals: a perspective. *Anim Health Res Rev*: 1-7

Tariq MA, Everest FL, Cowley LA, De Soyza A, Holt GS, Bridge SH, Perry A, Perry JD, Bourke SJ, Cummings SP, Lanyon CV, Barr JJ, Smith DL (2015) A metagenomic approach to characterize temperate bacteriophage populations from Cystic Fibrosis and non-Cystic Fibrosis bronchiectasis patients. *Frontiers in microbiology* 6: 97

Tejedor C, Foulds J, Zasloff M (1982) Bacteriophages in sputum of patients with bronchopulmonary *Pseudomonas* infections. *Infection and immunity* 36: 440-1

Tenaillon O, Denamur E, Matic I (2004) Evolutionary significance of stress-induced mutagenesis in bacteria. *Trends Microbiol* 12: 264-70

Tetart F, Desplats C, Krisch HM (1998) Genome plasticity in the distal tail fiber locus of the T-even bacteriophage: Recombination between conserved motifs swaps adhesin specificity. *J Mol Biol* 282: 543-556

Thomason LC, Costantino N, Court DL (2007) *E. coli* genome manipulation by P1 transduction. *Curr Protoc Mol Biol* Chapter 1: Unit 1 17

Tjalsma H, Boleij A, Marchesi JR, Dutilh BE (2012) A bacterial driver-passenger model for colorectal cancer: beyond the usual suspects. *Nat Rev Microbiol* 10: 575-82

Tree JJ, Granneman S, McAteer SP, Tollervey D, Gally DL (2014) Identification of Bacteriophage-Encoded Anti-sRNAs in Pathogenic *Escherichia coli*. *Mol Cell* 55: 199-213

Tremaroli V, Karlsson F, Werling M, Stahlman M, Kovatcheva-Datchary P, Olbers T, Fandriks L, le Roux CW, Nielsen J, Backhed F (2015) Roux-en-Y Gastric Bypass and Vertical Banded Gastroplasty

Induce Long-Term Changes on the Human Gut Microbiome Contributing to Fat Mass Regulation. *Cell Metab* 22: 228-38

Truong DT, Franzosa EA, Tickle TL, Scholz M, Weingart G, Pasolli E, Tett A, Huttenhower C, Segata N (2015) MetaPhlan2 for enhanced metagenomic taxonomic profiling. *Nat Methods* 12: 902-3

Tsai IJ, Otto TD, Berriman M (2010) Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. *Genome Biol* 11: R41

Tsai YT, Cheng PC, Pan TM (2014) Anti-obesity effects of gut microbiota are associated with lactic acid bacteria. *Appl Microbiol Biotechnol* 98: 1-10

Tschape F, Prager, H, R., Streckel, W., Fruth, A., Tietre, H., and Bohme, A. (1995) Verotoxigenic *Citrobacter freundii* associated with severe gastroenteritis and cases of haemolytic uraemic syndrome in a nursery school: green butter as the infection source. *Epidemiology and infection* 114: 441-450

Twort FW. (1915). An investigation on the nature of ultramicroscopic viruses. *Lancet*. 2: 1241–1243.

Typas A, Banzhaf M, Gross CA, Vollmer W (2012) From the regulation of peptidoglycan synthesis to bacterial growth and morphology. *Nat Rev Microbiol* 10: 123-136

Uauy RD, Fanaroff AA, Korones SB, Phillips EA, Phillips JB, Wright LL (1991) Necrotizing enterocolitis in very low birth weight infants: biodemographic and clinical correlates. National Institute of Child Health and Human Development Neonatal Research Network. *The Journal of pediatrics* 119: 630-8

Ubeda C, Djukovic A, Isaac S (2017) Roles of the intestinal microbiota in pathogen protection. *Clin Transl Immunology* 6: e128

Underkofler LA, Bantz AC, Peterson WH (1943a) Growth Factors for Bacteria: XIV. Growth Requirements of *Acetobacter suboxydans*. *J Bacteriol* 45: 183-190

Underkofler LA, Bantz AC, Peterson WH (1943b) Growth Factors for Bacteria: XIV. Growth Requirements of *Acetobacter suboxydans*. *J Bacteriol* 45: 183-90

Unkmeir A, Schmidt H (2000) Structural analysis of phage-borne stx genes and their flanking sequences in Shiga toxin-producing *Escherichia coli* and *Shigella dysenteriae* type 1 strains. *Infection and immunity* 68: 4856-4864

Unterholzner SJ, Poppenberger B, Rozhon W (2013) Toxin-antitoxin systems: Biology, identification, and application. *Mob Genet Elements* 3: e26219

Valen LV (1977) The Red Queen. *The American Naturalist* 111: 809-810

Van Boeckel TP, Gandra S, Ashok A, Caudron Q, Grenfell BT, Levin SA, Laxminarayan R (2014) Global antibiotic consumption 2000 to 2010: an analysis of national pharmaceutical sales data. *Lancet Infect Dis* 14: 742-750

van Nimwegen FA, Penders J, Stobberingh EE, Postma DS, Koppelman GH, Kerkhof M, Reijmerink NE, Dompeling E, van den Brandt PA, Ferreira I, Mommers M, Thijs C (2011) Mode and place of delivery, gastrointestinal microbiota, and their influence on asthma and atopy. *J Allergy Clin Immunol* 128: 948-U371

Vanalphen L, Lugtenberg B, Rietschel ET, Mommers C (1979) Architecture of the Outer-Membrane of *Escherichia-Coli-K12* - Phase-Transitions of the Bacteriophage-K3 Receptor Complex. *European Journal of Biochemistry* 101: 571-579

VanMelderen L, Thi MHD, Lecchi P, Gottesman S, Couturier M, Maurizi MR (1996) ATP-dependent degradation of CcdA by Lon protease - Effects of secondary structure and heterologous subunit interactions. *J Biol Chem* 271: 27730-27738

Vasquez A, Ahrne S, Pettersson B, Molin G (2001) Temporal temperature gradient gel electrophoresis (TTGE) as a tool for identification of *Lactobacillus casei*, *Lactobacillus paracasei*, *Lactobacillus zeae* and *Lactobacillus rhamnosus*. *Lett Appl Microbiol* 32: 215-219

Ventola CL (2015) The Antibiotic Resistance Crisis: Part 1: Causes and Threats. *Pharmacy and Therapeutics* 40: 277-283

Veses-Garcia M, Liu X, Rigden DJ, Kenny JG, McCarthy AJ, Allison HE (2015) Transcriptomic analysis of Shiga-toxigenic bacteriophage carriage reveals a profound regulatory effect on acid resistance in *Escherichia coli*. *Applied and environmental microbiology* 81: 8118-25

Viazis S, Diez-Gonzalez F (2011) Enterohemorrhagic *Escherichia Coli*: The Twentieth Century's Emerging Foodborne Pathogen: A Review. *Adv Agron* 111: 1-50

Virgin HW (2014) The virome in mammalian physiology and disease. *Cell* 157: 142-50

Vostrov AA, Vostrukhina OA, Svarchevsky AN, Rybchin VN (1996) Proteins responsible for lysogenic conversion caused by coliphages N15 and phi80 are highly homologous. *J Bacteriol* 178: 1484-6

Vrieze A, Van Nood E, Holleman F, Salojarvi J, Kootte RS, Bartelsman JF, Dallinga-Thie GM, Ackermans MT, Serlie MJ, Oozeer R, Derrien M, Druesne A, Van Hylckama Vlieg JE, Bloks VW, Groen AK, Heilig HG, Zoetendal EG, Stroes ES, de Vos WM, Hoekstra JB et al. (2012) Transfer of intestinal microbiota from lean donors increases insulin sensitivity in individuals with metabolic syndrome. *Gastroenterology* 143: 913-6 e7

Vuorio R, Vaara M (1992) Mutants Carrying Conditionally Lethal Mutations in Outer-Membrane Genes *OmsA* and *FirA* (*Ssc*) Are Phenotypically Similar, and *OmsA* Is Allelic to *FirA*. *J Bacteriol* 174: 7090-7097

Wagner EGH, Unoson C (2012) The toxin-antitoxin system *tisB-istR1* Expression, regulation and biological role in persister phenotypes. *Rna Biol* 9: 1513-1519

Wagner PL, Waldor MK (2002) Bacteriophage control of bacterial virulence. *Infection and immunity* 70: 3985-3993

Waldecker M, Kautenburger T, Daumann H, Busch C, Schrenk D (2008) Inhibition of histone-deacetylase activity by short-chain fatty acids and some polyphenol metabolites formed in the colon. *J Nutr Biochem* 19: 587-593

Wallis A, Ball M, Butt H, Lewis DP, McKechnie S, Paull P, Jaa-Kwee A, Bruck D (2018) Open-label pilot for treatment targeting gut dysbiosis in myalgic encephalomyelitis/chronic fatigue syndrome: neuropsychological symptoms and sex comparisons. *J Transl Med* 16

Wandersman C, Schwartz M (1978) Protein Ia and the lamB protein can replace each other in the constitution of an active receptor for the same coliphage. *Proceedings of the National Academy of Sciences of the United States of America* 75: 5636-9

Wang J, Hofnung M, Charbit A (2000) The C-terminal portion of the tail fiber protein of bacteriophage lambda is responsible for binding to LamB, its receptor at the surface of *Escherichia coli* K-12. *J Bacteriol* 182: 508-512

Wang P, Tang H, Fitzgibbon MP, McIntosh M, Coram M, Zhang H, Yi E, Aebersold R (2007) A statistical method for chromatographic alignment of LC-MS data. *Biostatistics* 8: 357-367

Wang X, Kim Y, Ma Q, Hong SH, Pokusaeva K, Sturino JM, Wood TK (2010) Cryptic prophages help bacteria cope with adverse environments. *Nature communications* 1: 147

Wang XX, Lord DM, Cheng HY, Osbourne DO, Hong SH, Sanchez-Torres V, Quiroga C, Zheng K, Herrmann T, Peti W, Benedik MJ, Page R, Wood TK (2012) A new type V toxin-antitoxin system where mRNA for toxin GhoT is cleaved by antitoxin GhoS. *Nat Chem Biol* 8: 855-861

Wang Y, Qian PY (2009) Conservative Fragments in Bacterial 16S rRNA Genes and Primer Design for 16S Ribosomal DNA Amplicons in Metagenomic Studies. *Plos One* 4

Warner BB, Deych E, Zhou Y, Hall-Moore C, Weinstock GM, Sodergren E (2016a) Gut bacteria dysbiosis and necrotising enterocolitis in very low birthweight infants: a prospective case-control study. *Lancet* 387

Warner BB, Deych E, Zhou Y, Hall-Moore C, Weinstock GM, Sodergren E, Shaikh N, Hoffmann JA, Linneman LA, Hamvas A, Khanna G, Rouggy-Nickless LC, Ndao IM, Shands BA, Escobedo M,

Sullivan JE, Radmacher PG, Shannon WD, Tarr PI (2016b) Gut bacteria dysbiosis and necrotising enterocolitis in very low birthweight infants: a prospective case-control study. *Lancet* 387: 1928-1936

Watson JD, Crick FH (1953) The structure of DNA. *Cold Spring Harb Symp Quant Biol* 18: 123-31

Webber MA, Piddock LJV (2003) The importance of efflux pumps in bacterial antibiotic resistance. *J Antimicrob Chemoth* 51: 9-11

Weber F, Wagner V, Rasmussen SB, Hartmann R, Paludan SR (2006) Double-Stranded RNA Is Produced by Positive-Strand RNA Viruses and DNA Viruses but Not in Detectable Amounts by Negative-Strand RNA Viruses. *Journal of Virology* 80: 5059-5064

Wen L, Ley RE, Volchkov PY, Stranges PB, Avanesyan L, Stonebraker AC, Hu C, Wong FS, Szot GL, Bluestone JA, Gordon JI, Chervonsky AV (2008) Innate immunity and intestinal microbiota in the development of Type 1 diabetes. *Nature* 455: 1109-13

West KH, Bystrom JM, Wojnarowicz C, Shantz N, Jacobson M, Allan GM, Haines DM, Clark EG, Krakowka S, McNeilly F, Konoby C, Martin K, Ellis JA (1999) Myocarditis and abortion associated with intrauterine infection of sows with porcine circovirus 2. *J Vet Diagn Invest* 11: 530-2

White, Bruns T, Lee S, Taylor J (1990) White, T. J., T. D. Bruns, S. B. Lee, and J. W. Taylor. Amplification and direct sequencing of fungal ribosomal RNA Genes for phylogenetics.

Whitman WB, Coleman DC, Wiebe WJ (1998) Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences of the United States of America* 95: 6578-83

WHO (2014) Water-related diseases. In *Water Sanitation Health*, Organization WH (ed) Geneva: World Health Organization

Wieczorek DJ, Feiss M (2003) Genetics of cosQ, the DNA-packaging termination site of phage lambda: local suppressors and methylation effects. *Genetics* 165: 11-21

Williams EJ, Kadambari S, Berrington JE, Luck S, Atkinson C, Walter S, Embleton ND, James P, Griffiths P, Davis A, Sharland M, Clark JE (2014) Feasibility and acceptability of targeted screening for congenital CMV-related hearing loss. *Arch Dis Child Fetal Neonatal Ed* 99: F230-6

Williams RJ, Eakin RE, Snell EE (1940) The Relationship of Inositol, Thiamin, Biotin, Pantothenic Acid and Vitamin B6 to the Growth of Yeasts. *Journal of the American Chemical Society* 62: 1204-1207

Willshaw GA, Smith HR, Scotland SM, Rowe B (1985) Cloning of genes determining the production of vero cytotoxin by *Escherichia coli*. *Journal of general microbiology* 131: 3047-53

Winfield MD, Groisman EA (2003) Role of nonhost environments in the lifestyles of *Salmonella* and *Escherichia coli*. *Applied and environmental microbiology* 69: 3687-3694

Winstanley C, Langille MG, Fothergill JL, Kukavica-Ibrulj I, Paradis-Bleau C, Sanschagrín F, Thomson NR, Winsor GL, Quail MA, Lennard N, Bignell A, Clarke L, Seeger K, Saunders D, Harris D, Parkhill J, Hancock RE, Brinkman FS, Levesque RC (2009) Newly introduced genomic prophage islands are critical determinants of *in vivo* competitiveness in the Liverpool Epidemic Strain of *Pseudomonas aeruginosa*. *Genome Res* 19: 12-23

Wiseman M (2008) The second World Cancer Research Fund/American Institute for Cancer Research expert report. Food, nutrition, physical activity, and the prevention of cancer: a global perspective. *Proc Nutr Soc* 67: 253-6

Wisse BE, Ogimoto K, Tang J, Harris MK, Raines EW, Schwartz MW (2007) Evidence that lipopolysaccharide-induced anorexia depends upon central, rather than peripheral, inflammatory signals. *Endocrinology* 148: 5230-5237

Wood DE, Salzberg SL (2014) Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol* 15

Woodard GA, Encarnacion B, Downey JR, Peraza J, Chong K, Hernandez-Boussard T, Morton JM (2009) Probiotics improve outcomes after Roux-en-Y gastric bypass surgery: a prospective randomized trial. *J Gastrointest Surg* 13: 1198-204

Worley B, Powers R (2013) Multivariate Analysis in Metabolomics. *Current Metabolomics* 1: 92-107

Wostmann BS (1981) The germfree animal in nutritional studies. *Annu Rev Nutr* 1: 257-79

Wostmann BS, Knight PL (1965) Antagonism between vitamins A and K in the germfree rat. *J Nutr* 87: 155-60

Wu GD, Chen J, Hoffmann C, Bittinger K, Chen YY, Keilbaugh SA, Bewtra M, Knights D, Walters WA, Knight R, Sinha R, Gilroy E, Gupta K, Baldassano R, Nessel L, Li H, Bushman FD, Lewis JD (2011) Linking long-term dietary patterns with gut microbial enterotypes. *Science* 334: 105-8

Wu HJ, Ivanov, II, Darce J, Hattori K, Shima T, Umesaki Y, Littman DR, Benoist C, Mathis D (2010) Gut-residing segmented filamentous bacteria drive autoimmune arthritis via T helper 17 cells. *Immunity* 32: 815-27

Xia JG, Sinelnikov IV, Han B, Wishart DS (2015) MetaboAnalyst 3.0-making metabolomics more meaningful. *Nucleic Acids Research* 43: W251-W257

Xia LC, Cram JA, Chen T, Fuhrman JA, Sun F (2011a) Accurate genome relative abundance estimation based on shotgun metagenomic reads. *Plos One* 6: e27992

Xia LC, Cram JA, Chen T, Fuhrman JA, Sun FZ (2011b) Accurate Genome Relative Abundance Estimation Based on Shotgun Metagenomic Reads. *Plos One* 6

Xu J, Hendrix RW, Duda RL (2013) A Balanced Ratio of Proteins from Gene G and Frameshift-Extended Gene GTIs Required for Phage Lambda Tail Assembly. *J Mol Biol* 425: 3476-3487

Yadav SK, Boppana S, Ito N, Mindur JE, Mathay MT, Patel A, Dhib-Jalbut S, Ito K (2017) Gut dysbiosis breaks immunological tolerance toward the central nervous system during young adulthood. *Proceedings of the National Academy of Sciences of the United States of America* 114: E9318-E9327

Yang B, Wang Y, Qian PY (2016a) Sensitivity and correlation of hypervariable regions in 16S rRNA genes in phylogenetic analysis. *Bmc Bioinformatics* 17

Yang JY, Kim MS, Kim E, Cheon JH, Lee YS, Kim Y, Lee SH, Seo SU, Shin SH, Choi SS, Kim B, Chang SY, Ko HJ, Bae JW, Kweon MN (2016b) Enteric Viruses Ameliorate Gut Inflammation via Toll-like Receptor 3 and Toll-like Receptor 7-Mediated Interferon-beta Production. *Immunity* 44: 889-900

Yang Q, Catalano CE (2003) Biochemical characterization of bacteriophage lambda genome packaging in vitro. *Virology* 305: 276-87

Yang X, Liu D, Liu F, Wu J, Zou J, Xiao X, Zhao F, Zhu B (2013) HTQC: a fast quality control toolkit for Illumina sequencing data. *BMC Bioinformatics* 14: 33

Yao Z, Davis RM, Kishony R, Kahne D, Ruiz N (2012) Regulation of cell size in response to nutrient availability by fatty acid biosynthesis in *Escherichia coli*. *Proceedings of the National Academy of Sciences of the United States of America* 109: E2561-8

Ye J, McGinnis S, Madden TL (2006) BLAST: improvements for better sequence analysis. *Nucleic Acids Res* 34: W6-9

Yeo A, Feiss M (1995) Specific interaction of terminase, the DNA packaging enzyme of bacteriophage lambda, with the portal protein of the prohead. *J Mol Biol* 245: 141-50

Yutin N, Koonin EV (2012) Hidden evolutionary complexity of Nucleo-Cytoplasmic Large DNA viruses of eukaryotes. *Virol J* 9

Zerbino DR, Birney E (2008) Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* 18: 821-829

Zhang H, DiBaise JK, Zuccolo A, Kudrna D, Braidotti M, Yu Y, Parameswaran P, Crowell MD, Wing R, Rittmann BE, Krajmalnik-Brown R (2009) Human gut microbiota in obesity and after gastric bypass. *Proc Natl Acad Sci U S A* 106: 2365-70

Zhang Q, Pell J, Canino-Koning R, Howe AC, Brown CT (2014) These Are Not the K-mers You Are Looking For: Efficient Online K-mer Counting Using a Probabilistic Data Structure. *Plos One* 9: e101271

Zhang Q, Raoof M, Chen Y, Sumi Y, Sursal T, Junger W, Brohi K, Itagaki K, Hauser CJ (2010) Circulating mitochondrial DAMPs cause inflammatory responses to injury. *Nature* 464: 104-7

Zhang YM, Rock CO (2008a) Membrane lipid homeostasis in bacteria. *Nat Rev Microbiol* 6: 222-233

Zhang YM, Rock CO (2008b) Membrane lipid homeostasis in bacteria. *Nat Rev Microbiol* 6: 222-33

Zhao XN, Cui YJ, Yan YF, Du ZM, Tan YF, Yang HY, Bi YJ, Zhang PP, Zhou L, Zhou DS, Han YP, Song YJ, Wang XY, Yang RF (2013) Outer Membrane Proteins Ail and OmpF of *Yersinia pestis* Are Involved in the Adsorption of T7-Related Bacteriophage Yep-phi. *Journal of virology* 87: 12260-12269

Zheng P, Zeng B, Zhou C, Liu M, Fang Z, Xu X, Zeng L, Chen J, Fan S, Du X, Zhang X, Yang D, Yang Y, Meng H, Li W, Melgiri ND, Licinio J, Wei H, Xie P (2016) Gut microbiome remodeling induces depressive-like behaviors through a pathway mediated by the host's metabolism. *Mol Psychiatr* 21: 786-796

Zhou B, Xiao JF, Tuli L, Ransom HW (2012) LC-MS-based metabolomics. *Mol Biosyst* 8: 470-81

Zhou SL, Song Q, Tang Y, Weng ND (2005) Critical review of development, validation, and transfer for high throughput bioanalytical LC-MS/MS methods. *Curr Pharm Anal* 1: 3-14

Zhou Y, Shan G, Sodergren E, Weinstock G, Walker WA, Gregory KE (2015a) Longitudinal analysis of the premature infant intestinal microbiome prior to necrotizing enterocolitis: a case-control study. *Plos One* 10: e0118632

Zhou Y, Shan G, Sodergren E, Weinstock G, Walker WA, Gregory KE (2015b) Longitudinal analysis of the premature infant intestinal microbiome prior to necrotizing enterocolitis: a case-control study. *PLoS One* 10

Zitvogel L, Ayyoub M, Routy B, Kroemer G (2016) Microbiome and Anticancer Immunosurveillance. *Cell* 165: 276-87

Zoetendal EG, Raes J, van den Bogert B, Arumugam M, Booijink CCGM, Troost FJ, Bork P, Wels M, de Vos WM, Kleerebezem M (2012) The human small intestinal microbiota is driven by rapid uptake and conversion of simple carbohydrates. *Isme Journal* 6: 1415-1426

Zumbrun SD, Hanson L, Sinclair JF, Freedy J, Melton-Celsa AR, Rodriguez-Canales J, Hanson JC, O'Brien AD (2010) Human Intestinal Tissue and Cultured Colonic Cells Contain Globotriaosylceramide Synthase mRNA and the Alternate Shiga Toxin Receptor Globotetraosylceramide. *Infection and Immunity* 78: 4488-4499

Chapter 10. Appendices

10.1 Appendix 1

10.1.1 Phage receptors localised in capsular and slime polysaccharides, pili and flagella

Adsorption of the phage to the bacterial host flagella can allow transitioning to the cell wall via the corkscrew motion of the flagella, once reaching the baseplate of the flagella the binding becomes irreversible. Advantages of this mechanism would likely be improved chance of collision and therefore infection. In some cases both the head and tail of the phage bind to the flagella, leaving the majority of the tail to adsorb to the cell wall, as well as leaving potential of binding to a neighbouring bacterial cell.

The capsular method of phage adsorption involves the binding to a targeted antigen alongside enzymes localised to the phage tail used for either deacetylation (cleavage of acetyl groups) or depolymerisation, depending on the phage and enzyme. Deacetylation prevents further phage binding, whilst still being reversible.

There are two types of pili adsorption phage; RNA-containing viruses with isometric capsids and DNA-containing viruses in the form of filaments. The RNA-containing viruses second capsid component is protein A, protein A is responsible for recognition and adsorption of virion to pili. Interestingly these pili phages use only the sex (conjugative) pili of bacteria, able to adsorb several hundred phage virions.

There are two groups of DNA-containing pili adsorption viruses, these are; 'Ff' phages that adsorb to terminal parts of F pili and 'If' phages that adsorb to terminal parts of 'I' pili. In this case only a few of these virion can adsorb to a single pilus.

Table 10.1 Regulatory elements of bacteriophage Lambda

Regulatory element	Function
Proteins	
CI	At low concentrations a repressor of P_R and P_L and an activator of P_{RM} ; at high concentrations also represses P_{RM}
CII	An activator of P_{RE} , pI , pE , P_{int} , paQ
CIII	Stabalises CII
Cro	At low concentrations a repressor of P_{RM} ; at high concentrations also represses P_L and P_R
N	An antiterminator at t_{L1} , t_{R1} , t_{R2} , and other terminators
Q	An antiterminator for late gene transcription (tR')
sieB	Superinfection exclusion
Promoters	
P_R	Major rightward transcription
(P_L) P_{L1}	Major leftward transcription
(P_L) P_{L2}	Major leftward transcription
O_R	Transcription for right operator (O_R) site
O_L	Transcription for left operator (O_L) site
P_{RM}	Transcription for repressor maintenance
P_{RE}	Transcription for repressor establishment
$pOOP$ or P_O	Transcription for OOP
$psieB$	Transcription for <i>sieB</i>
pI	transcription for <i>int</i>
paQ	Reduces the Q anti-termination function and thus late gene expression
pR'	Promoter for late gene transcription
P_{int}	Transcription of genes for integration and excision
Terminators	
tR'	Terminates late gene transcription
$tR1$	least Rho sensitive dependent termination of P_R transcription
$tR2$	3rd most Rho sensitive dependent termination of P_R transcription
$tR3$	2nd most Rho sensitive dependent termination of P_R transcription
$tR4$	Most Rho sensitive dependent termination of P_R transcription
tO	Terminates transcription after OOP
$tL1$	Termination pf P_L transcription
$tL2$	Termination pf P_L transcription
$tL3$	Termination pf P_L transcription
Anti-sense RNA	
OOP	indirectly inhibits <i>cII</i> translation
Operator regions (function with CI binding)	
O_{R1}	P_R repressor
O_{R2}	P_R repressor, P_{RM} activator (repressor when binding is co-operative with O_{R3})
O_{R3}	P_{RM} repressor

O_{L1}	co-operative binding with O _{L2} represses P _L . O _{L1} can help form an intervening DNA loop, O _{L1} -O _{R1} = dimer = decreasing CI requirement for binding. O _{L1} -O _{R1} +O _{L2} -O _{R2} = octamer = Represses P _R and activates P _{RM}
O_{L2}	co-operative binding with O _{L1} represses PL. O _{L2} can help form an intervening DNA loop, O _{L2} -O _{R2} = dimer =decreasing CI requirement for binding. O _{L1} -O _{R1} +O _{L2} -O _{R2} = octamer = Represses PR and activates PRM
O_{L3}	Repress P _L .Helps form an intervening DNA loop (O _{L3} -O _{R3}) decreasing CI requirement for binding, enhance repression of R _{PM}
Host factors	
UP	Enhances activation of P _{RM} by interaction with α-CTD of RNAP bound to P _{RM} (when looped). Allows P _{L2} expression.
Integration host factor (IHF)	Stimulates PL1, represses PL2. Modulates CII and CIII levels and N gene translation
hostencoded HflB(FtsH)-HflC-HflK protease complex	Destabalises CII stability by degrading it
cAMP	May control HflBCK levels
ppGpp	May control HflBCK levels
RNase III	OOP RNA from lambda Po promoter is antisense to the 3' portion of cII gene mRNA and acts to destabilize that message with the help of RNase III
HflD	May directly inhibit DNA binding by CII protein,
DnaA	May affect PL transcription
SeqA	May affect PL transcription
LexA	Represses the OOP promoter

10.1.2 Cancer

Metabolites produced by the microbiome have been shown to directly or indirectly influence the development of cancer, either directly or indirectly via the immune system. One area of investigation is the role of the gut microbiota in colorectal cancer, which is the third most common cause of cancer mortality in the world (Jemal, Bray et al., 2011). Both diet and lifestyle have been implicated in colorectal cancer (Gill & Rowland, 2002, Jemal et al., 2011, Wiseman, 2008). The gut microbiota is also known to influence inflammatory bowel disease (IBD), and IBD has shown to increase incidence of colorectal cancer (Danese, Malesci et al., 2011, Jess, Gamborg et al., 2005). The gut microbiota and their metabolic products have collectively been shown to have a role in the protection against and the predisposition of colorectal cancer (see Figure 10.2) (Elinav, Nowarski et

al., 2013, Gill & Rowland, 2002, Kostic, Chun et al., 2013, Schwabe & Jobin, 2013, Sears & Garrett, 2014, Tjalsma, Boleij et al., 2012). As well as the collective microbiota, species of bacteria have also been implemented in anti-tumour properties (see Table 10.2)

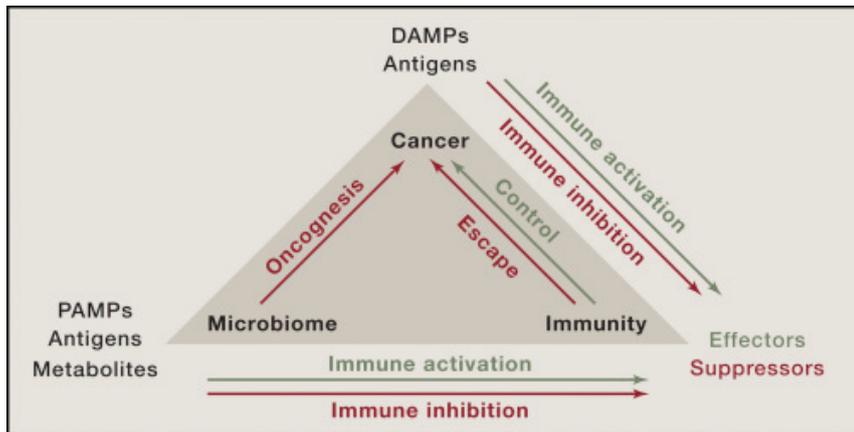


Figure 10.1 Triangulation between the Microbiome, the Immune System, and Cancer (Zitvogel, Ayyoub et al., 2016)

Dietary and environmental compounds	Microbial products	Known effect on host
Non-digestible carbohydrates	SCFAs	<ul style="list-style-type: none"> • Microbiota modulation • Cellular differentiation; apoptosis • Inflammation
Phytochemicals	Phenolic acids; isothiocyanates	<ul style="list-style-type: none"> • Xenobiotic detoxification • Microbiota modulation • Cellular differentiation; apoptosis • Inflammation
Protein	NOCs; ammonia	• ROS production; genotoxicity
	Polyamines	<ul style="list-style-type: none"> • Inflammation • ROS production; genotoxicity
	Hydrogen sulphide	<ul style="list-style-type: none"> • Inflammation • ROS production; genotoxicity
Fat → Bile acids	Taurine	• Microbiota modulation
	Secondary bile acids	<ul style="list-style-type: none"> • Microbiota modulation • Cellular differentiation; apoptosis • ROS production; genotoxicity
Xenobiotics	Carcinogens	• ROS production; genotoxicity
Ethanol	Acetaldehyde	• ROS production; genotoxicity

Figure 10.2 Major microbial metabolites formed from dietary and environmental compounds that are involved in the initiation and/or progression of colorectal cancer. Metabolites that are thought to exert mainly anti-carcinogenic properties are shown in blue, and metabolites that have mainly pro-carcinogenic properties are shown in red. Only clearly established modes of action in the host are indicated. NOCs, N-nitroso compounds; ROS, reactive oxygen species; SCFAs, short-chain fatty acids (Louis, Hold et al., 2014).

Table 10.2 The immunological effects of gut microbiota(Botticelli, Zizzari et al., 2017)

Bacteria	Model	Effects on immune system	References
Lactobacillus johnsonii	mouse	Stimulates the differentiation of TH17 cells and Th1 cells	Viaud 2013
Enterococcus hirae	mouse	Stimulates the differentiation of TH17 cells and Th1 cells	Viaud 2013
Ruminococcus	mouse	TNF production, promotes response to immunotherapy	lida 2013
Alistipes shahii	mouse	TNF production, promotes response to immunotherapy	lida 2013
Lactobacillus fermentum	mouse	TNF production , impairs response to immunotherapy	lida 2013
Bacteroides fragilis	mouse	Induces TH 1 in tumor draining lymph nodes. Promotes the maturation of intratumoral dendritic cells Increases the activity of anti-CTLA4 in vivo Reduces the inflammatory response Reduced histopathology signs of colitis induced by CTLA4 blockade	Vetizou 2015
Bacteroides thetaiotamicron	mouse	Increaseas the activity of anti-CTLA4 in vivo Reduced the inflammatory response	Vetizou 2015
Bacteroidales	mouse	Decreased after CTLA4 blockade	Vetizou 2015
Burkholderiales	mouse	Decreased after CTLA4 blockade	Vetizou 2015
Clostridiales	mouse	Increased after CTLA4 blockade	Vetizou 2015
Bifidobacterium breve, Bifidobacterium longum, Bifidobacterium adolescentis	mouse	Enhanced dendritic cells activation, Increased CD8 +T cell accumulation	Sivan 2015
Bifidobacterium breve Bifidobacterium longu,	mouse	Improved the response to PDL-1 Improved IFN γ levels	Sivan 2015
Bacteroidetes	human	Enriched in colitis-resistant patients treated with ipilimumab	Dubin 2015
Clostridium species	mouse	Stimulates the induction of suppressive FOXP3+ Treg	Geuking 2011
Bacteroides fragilis	mouse	Stimulates the induction of suppressive FOXP3+ Treg	Geuking 2011
Staphylococcus aureus	mouse	Converts CD4+ T cells into Foxp3+ Treg cell	Hardis rabe 2013
Bacteroidaceae	mouse	Decreases in mice PD-1-/-	Kawamoto 2012
Bifidobacterium	mouse	Decreases in mice PD-1-/-	Kawamoto 2012
Enterobacteriaceae	mouse	Increases in mice PD1-/-	Kawamoto 2012
Erysipelotrichaceae Prevotellaceae Alcaligenaceae			
TM7 incerte saedis	mouse	Increase in mice PD1-/-	Kawamoto 2012

10.1.3 Eating disorders and obesity

Obesity is a serious problem in the developed world, though underlying mechanisms are yet to be completely identified, research has implicated the gut microbiota (Alang & Kelly, 2015, Boroni Moreira, Fiche Salles Teixeira et al., 2012, Bruce-Keller, Salbaum et al., 2015, Cani, Bibiloni et al., 2008, Cani & Delzenne, 2010, Cox, West et al., 2015, Delzenne, Neyrinck et al., 2011, Flint, 2011, Murphy, Cotter et al., 2013, Musso, Gambino et al., 2010, Riva, Borgo et al., 2016, Scarpellini, Campanale et al., 2010, Tagliabue & Elli, 2013, Tsai, Cheng et al., 2014, Zhang, DiBaise et al., 2009). It has been identified that obese mice and humans have comparatively different microbiota to that of their lean counterparts. The difference in microbiota seems to have a role in improved metabolic function, and has been shown in many studies (Aron-Wisnewsky, Dore et al., 2012, Bueter, Abegg et al., 2012, Cani, Osto et al., 2012, Cox & Blaser, 2013, Graessler, Qin et al., 2013, Kallus & Brandt, 2012, Kong, Tap et al., 2013, Kugelberg, 2013, Lutz & Bueter, 2014, Osto, Abegg et al., 2013, Tremaroli, Karlsson et al., 2015, Woodard, Encarnacion et al., 2009, Zhang et al., 2009). Of particular note, Liou et al (2013) demonstrated weight loss and reduced fat mass after faecal transplantation from Roux-en-Y gastric bypass treated mice into germ-free mice (Liou, Paziuk et al., 2013). SCFA produced by the microbiota have been shown to aid in insulin sensitivity (Gao, Yin et al., 2009), glucose homeostasis, and protection against diet-induced obesity (Lin, Frassetto et al., 2012). SCFA have also been associated to cholesterol, lipid, and glucose metabolism (Conterno, Fava et al., 2011, Heimann, Nyman et al., 2016).

Gut–brain communication changes in eating disorders due to an impaired regulation of appetite control and satiety. There is an evolutionary drive for the microbiota to alter host feeding behaviour as varied bacterial communities and species have different nutritional needs, examples include; the strong association of *Bacteroides*' preference for fat and protein and *Prevotella*'s preference for carbohydrates (Wu, Chen et al., 2011). It has been hypothesised that a diverse gut microbiota is important for host diet regulation, as dominance by certain groups of microbes could create constant biased nutritional drives on the host, potentially causing dietary patterns and/or preferences (Alcock et al., 2014). Gut bacteria can affect the activity and production of appetite-

regulating hormones. An example of modification in hormone secretion can be seen in the interaction of bacterial products like flagellin, lipopolysaccharides, and peptides with Toll-like receptors and immune responses (Raybould, 2010). Though not all research agrees (Ruud, Wilhelms et al., 2013), investigation has suggested the gut microbiota may stimulate inflammation-induced anorexia via lipopolysaccharide production (Wisse, Ogimoto et al., 2007). Bacterial metabolites that are analogues of mammalian appetite regulating hormones may have an important role in pathogenesis and progression of eating disorders, as autoimmune responses associated to appetite hormones have been correlated with psychological traits of eating disorders (Fetissov, Hallman et al., 2002, Fetissov, Harro et al., 2005). Interestingly, the effects of the microbiota on behavioural abnormalities have been shown to be transmissible to a germ-free host via a cecal content transplant (Bercik, Denou et al., 2011, Lam, Maguire et al., 2017, Zheng, Zeng et al., 2016).

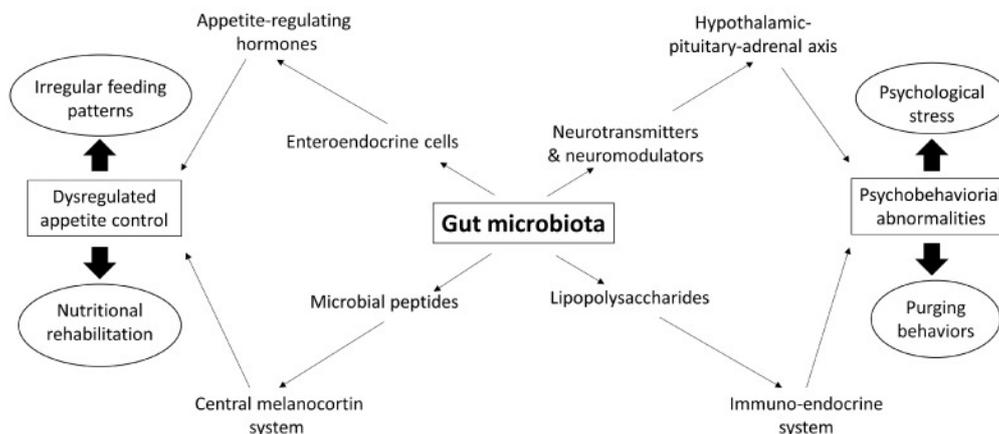


Figure 10.3 Potential mechanistic role of the gut microbiota in the aetiology and progression of eating disorders. The gut microbiota has been shown to modulate the hypothalamic–pituitary–adrenal axis and the enteroendocrine, central melanocortin, and immune-endocrine systems that may collectively contribute to dysregulated appetite control and psychobehavioral abnormalities typically seen in eating disorders. Emerging evidence also suggests that the outcomes of the illnesses and the related treatments (in ovals) may feed back to the gut ecosystem that further negatively impact on the progression of the diseases (Lam et al., 2017).

10.1.4 Prebiotics and Probiotics

Prebiotics are dietary products such as complex carbohydrates and protein which influence the gut microbiota to benefit host health (see Table 10.3 and Figure 10.4). Examples of prebiotics include complex carbohydrates like fructans, arabinoxylan, and inulin. These complex carbohydrates, which are fermented by the gut microbiota into short-chain fatty acids, can have numerous positive influences on host health (as previously discussed). Diet can also alter the gut microbiota by promoting the growth of different microbes, in the case of prebiotics it can be used to promote a healthy gut microbiota.

Prebiotic and Probiotic techniques are now used in an attempt to achieve improved results, this technique is termed synbiotics, examples of synbiotics can be seen in Table 10.5.

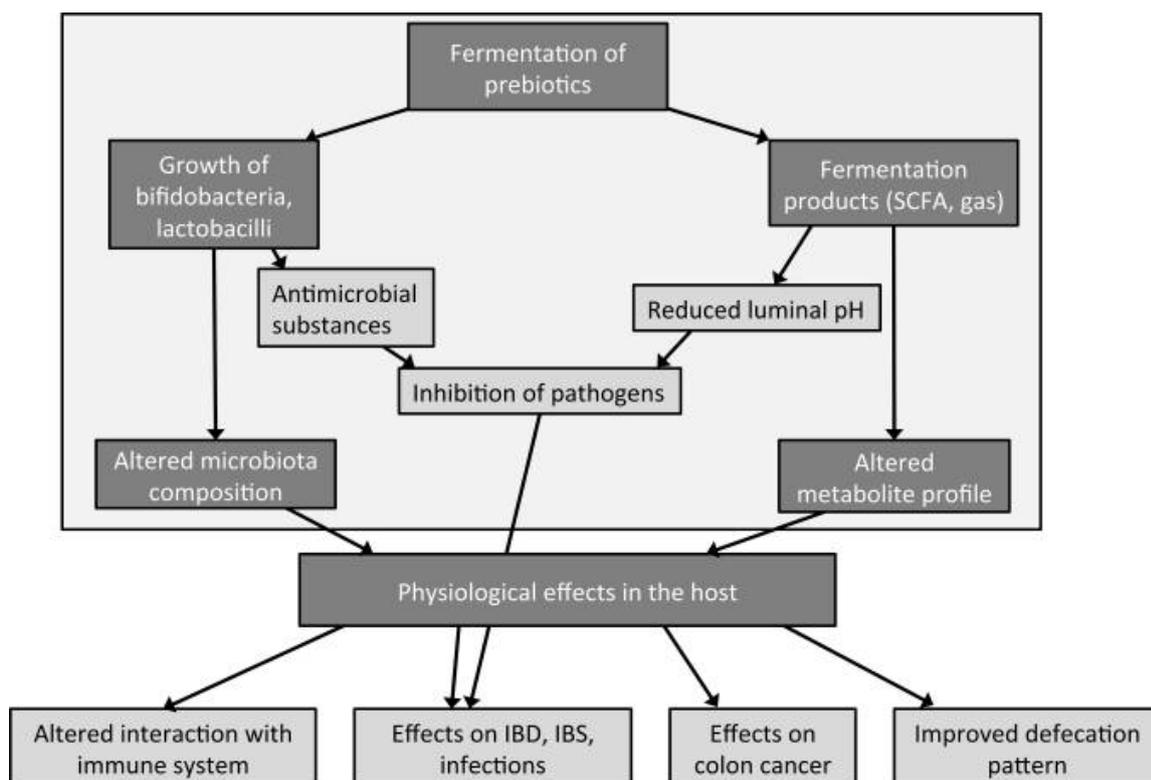


Figure 10.4 Proposed mechanism of action of prebiotics. IBD, inflammatory bowel disease; IBS, irritable bowel syndrome (Hamer, De Preter et al., 2012).

Table 10.3 Prebiotics: Randomised controlled trials

Study	Population, Sample size (n)	Duration (wks), Design	Treatment: daily dose	Control: daily dose	Results and study details, Comparisons are between intervention vs. control after intervention (<.05) unless specified ^b
Parnell & Reimer, 2009 (19)	OW/OB M/F n = 37	12 DB	OF 21 g n = 20	MD 21 g n = 17	Decreased: BW, FBG, fasting insulin, calorie intake, ghrelin Increased: Protein-YY NS: TC, LDL, HDL, TG, GLP-1
Genta et al, 2009 (20)	OB, F, n = 35	16 DB	Yacon syrup, containing 0.14 g FOS/kg (~10 g/70 kg BW) n = 20	Placebo syrup n = 15	Decreased: BW, BMI, WaistC, FBG (-11.9 mg/dL), HOMA-IR Decreased: Fasting insulin (-9.9 mU/mL), LDL (-35.2 mg/dL) NS: TC, HDL
Tovar et al, 2012 (21)	OW/OB, F, n = 59	12 OL	LCDiet + Inulin 10 g, n = 30	LCDiet, n = 29	NS: BW, WaistC, FBG, TC, LDL, HDL, TG NS: BW, BMI, FBG, fasting insulin, A1c, TC, LDL, HDL, LPS Decreased: Bacteroides, Propionibacterium Increased: Bifidobacterium, Faecalibacterium prausnitzii Decreased: fecal total SCFA, acetate, & propionate Increased: B. longum, B. pseudocatenulatum, B. adolescentis Negatively correlated: B. longum with LPS and endotoxin (P<.01). Positively correlated: Fecal total SCFA, acetate, & propionate with BMI, fasting insulin and HOMA-IR
Dewulf et al, 2013 (22), Salazar et al, 2015 (23)	OB, F, n = 30	12 DB	Inulin + OF, 50/50% mix, 16 g, n = 15	MD 16 g, n = 15	Decreased: fasting insulin (-1.7 mU/mL), TC (-11.6 mg/dL), TG (-8.9 mg/dL)
Vulevic et al, 2013 (24)	OW/OB & MetS, M/F, n = 45	12 CO	GOS, 5.5 g, n = 45	MD, 5.5 g, n = 45	Decreased: CRP, fecal calprotectin NS: BW, BP, FBG, LDL, HDL Decreased: FBG (-8.5%), A1c (-10.4%) Decreased: marker of oxidative stress malondialdehyde (-37.2%) Increased: antioxidant defense (total antioxidant capacity 18.8% & SOD 4.36%)
Gargari et al, 2013 (25)	OB & DM2, F, n = 49	8 DB	Inulin 10 g, n = 24	MD 10 g, n = 25	NS: fasting insulin, HOMA-IR, antioxidant catalase and GSH Decreased: TC (-12.9%), TG (-23.6%), LDL (-35.3%)
Dehghan et al, 2013 (26)	OB & DM2, F, n = 49	8 DB	Inulin 10 g, n = 24	MD 10 g, n = 25	Increased: HDL (19.9%) Decreased: hsCRP (-35.6%), TNF-α (-23.1%), and LPS (-27.9%)
Dehghan et al, 2014 (27)	OB & DM2, F, n = 52	8 DB	OF-enriched Inulin 10 g, n = 27	MD 10 g, n = 25	Decreased: FBG (19.2 mg/dL; 9.50%), A1c (1.0%; 8.40%), IL-6 (1.3 pg/mL; 8.15%), TNF-α (3.0 pg/mL; 19.80%) & LPS (6.0 EU/mL; 21.95%) NS: hsCRP, IF-, IL-10

Table 10.4 Probiotics: Randomised controlled trials

Study	Population, Sample size (n)	Duration (wks), Design	Treatment: daily dose	Control: daily dose	Results and study details, Comparisons are between intervention vs. control after intervention (<.05) unless specified ^b
Agerholm-Larsen et al, 2000 (36)	OW/OB, M/F, n = 70	8 DB	Yogurt (Y) 450 mL, 3 groups: StLa, StLr, G, 10 ⁷ -10 ¹⁰ CFU	PL 2 groups: PY, PP	Comparison G vs. PY, PP: NS: BW, WHR, BP, FatM, TC, HDL, TG Decreased: LDL (-8.4%); Increased: fibrinogen after adjusting for BW Groups: Gr1 (StLa), n = 16: Y fermented with S.thermophiles (2 strains) + L. acidophilus, Gr2 (PY), n = 14: PL Y fermented with delta-acid-lactone, Gr3 (StLr), n = 14: Yogurt fermented with S.thermophiles (2 strains) + L. rhamnosus, Gr4 (G), n = 16: Y fermented with S. thermophiles (2 strains) + Enterococcus faecium, Gr5 (PP), n = 10: 2 PL pills daily
Kadooka et al, 2010 (37)	OW/OB, M/F, n = 87	12 DB	Yogurt 200 g with L. gasseri (LG2055), 5 x 10 ¹⁰ CFU, n = 43	PL Yogurt 200 g (PL), n = 44	Decreased: VisFat -4.6%, SubFat -3.3%, BW -1.4%, BMI -1.5%, WaistC, 1.8%, HipC -1.5%, FatM -0.8 kg Increased: adiponectin NS: SubFat, LeanM Gr1 vs. PL Decreased: BMI -1.1%, WaistC -1.4%, HipC -1.5%, VisFat -8.5%, FatM -2.4 kg NS: SubFat, LeanM
Kadooka et al, 2013 (38)	OW/OB, M/F, n = 210	12 DB	Yogurt 200 g with L. gasseri (LG2055)	2 groups, PL Yogurt, 200 g (PL), n = 70	Gr2 vs. PL Decreased: BMI -1.6%, WaistC -1.2%, VisFat -8.2%, FatM -2.2 kg NS: SubFat, LeanM Groups: Gr1, n = 69: LG2055 10 ⁷ CFU; Gr2, n = 71: LG2055 10 ⁶ CFU
Zarrati et al, 2013 (39)	OW/OB, M/F, n = 50	8 DB	LCDiet + L. spp.-yogurt with: L. acidophilus La5+, B. Bb12+, L. casei DN001, 10 ⁸ CFU, n = 25	Regular yogurt, n = 25	NS: BW, BMI, WaistC, HipC, WHR, SBP, DBP, hs-CRP, IL-17, TNF-a Yogurt vs. Milk: increased HOMA-IR (+12.5%), ProCap vs. PL: Increased FBG (+2.8%)
Ivey et al, 2014 (40), 2015 (41)	OW/OB, M/F, n = 156	6 DB	Yogurt + ProCap, Both containing: L. acidophilus La5 + B. animalis subsp. lactis Bb12, 3.0 x 10 ⁹ CFU, 4 groups	Milk + PL, n = 40	NS: A1c, BP, TC, LDL, HDL, TG Groups: Gr1, n = 40: Yogurt + ProCap; Gr 2, n = 37: Yogurt + PL; Gr3, n = 39: Milk + ProCap; Gr4, n = 40: Milk + PL
Chang et al, 2011 (42)	OW/OB & MetS, M/F, n = 101	8 DB	Functional yogurt 150 mg BID with S. thermophiles L. acidophilus B. infantis 109-1010 CFU n = 53	PL Yogurt, 150 mg BID, n = 48	Decreased: BW, BMI, LDL NS: WaistC, BP, FBG, A1c, TC, HDL, TG NS: BW, BMI, FBG, AUC-Glucose, Fasting insulin, HOMA-IR, HOMA-b, ISI
Tripolt et al, 2014 (43)	OW/OB & MetS, M/F, n = 28	12 OL	Yakult 195 ml: L. casei Shirota 3 x 6.5 x 10 ⁹ CFU, n = 13	None, n = 15	NS: LDL, TNF, hsCRP, IL-6 Decreased: sVCAM Decreased: FBG & homocysteine
Barreto et al, 2014 (44)	OW/OB & MetS, F, n = 24	12 DB	Yogurt 80 mL with L. plantarum 1.25 x 10 ⁷ CFU, n = 12	Milk, 80 mL, n = 12	NS: BW, BMI, WaistC, SBP, DBP, fasting insulin, HOMA-IR, TC, LDL, HDL, TG NS: CRP, IL-6, TNF-a
Jung et al, 2013 (45)	PreDM, M/F, n = 48	12 DB	Pill: L. gasseri BNR17, 1010 CFU, n = 22	PL, n = 26	NS: BW, BMI, BP, Body Fat (5), WaistC, HipC, VisFat, SubFat, NS: FBG, fasting insulin, A1c, TC, LDL, HDL, TG, BMR, oxygen consumption
Hariri et al, 2015 (46)	DM2, M/F, n = 40	8 DB	Probiotic soy milk, 200 mL with L. plantarum A7 2 x 10 ⁷ , n = 20	Soy milk 200 mL n = 20 PL N = 22 at 12 wks	Decreased: SBP, DBP NS: BW, BMI, WHR
Woodard et al, 2009 (47)	OB & RYGB, M/F, n = 44	24 DB	ProCap: L. spp. 2.4 x 10 ⁹ CFU n = 17 at 12 wks n = 15 at 24 wks	N = 20 at 24 wks	Decreased: BW at 12 wks: ProCap -48% vs. PL -39% Increased: B12 at 12 wks & 24 wks NS: BW at 24 wks: ProCap -67% vs. PL -60

Table 10.5 Synbiotics: Randomised controlled trials

Study	Population, Sample size (n)	Duration (wks), Design	Treatment: daily dose	Control: daily dose	Results and study details, Comparisons are between intervention vs. control after intervention (<.05) unless specified ^b
Lee SJ et al, 2014 (48)	OW/OB, M/F, n = 50	8 DB	BTS + DUOLAC, 7:5 × 10 ⁹ CFU, n = 17	BTS + PL, n = 19	NS: BW, BMI, WaistC, FatM (by BIA), LeanM, FBG, LPS, TC, LDL, TG, NS: Gut permeability Increased: HDL Correlations of BW with L. plantarum (r = 0.425) and LPS with B. breve (r = -0.350). GMB change: within DUOLAC7 group: Increased: B. breve, B. lactis, B. rhamnosus, B. plantarum DUOLAC7: L. acidophilus, L. plantarum, L. rhamnosus, B. lactis, B. longum, B. breve, S. thermophilus Comparison between groups: NS for all markers Comparison of Females: at week 24 Decreased: BW (-5.2 kg), FatM (-4.8 kg), SBP -1.5 mmHg, leptin (-11.3 ng/mL) NS: mean daily energy, BMR, respiratory quotient, FBG, fasting insulin
Sanchez et al, 2014 (49)	OB, M/F, n = 93	24 DB	OF 200 mg + Inulin 100 mg + L. rhamnosus (LPR) 1.6 × 10 ⁸ CFU, 2 pills per day, n = 45 (F = 26)	MD 250 mg + PL, n = 48, (F = 28)	NS: TC, LDL, HDL, TG, adiponectin, NEFA, hydroxybutyrate, LBP, CRP Increased abundance of Lachnospiraceae family Phase 1: 12 wks of weight loss (500 kcal energy restriction) Phase 2: 12 wks of weight maintenance Decreased: FBG, TC, TG Increased: HDL
Eslamparast et al, 2014 (50)	OB & MetS, M/F, n = 38	28 DB	FOS 250 mg + 7 strains, 2 × 10 ⁸ -10 ¹⁰ CFU, n = 19	MD, n = 19	NS: BW, BMI, WaistC, fasting insulin, HOMA-IR, LDL, MET 7 strains: L. acidophilus, L. casei, L. rhamnosus, L. bulgaricus, B. longum, B. breve, S. thermophilus Smaller increase in: FBG, HOMA-IR, hs-CRP, GSH
Asemi et al, 2013 (51)	OW/OB, & DM2, M/F, n = 54	8 DB	FOS 100 mg + 7 strains, 2 × 10 ⁸ -10 ¹⁰ CFU	PL	NS: BW, BMI, A1c, Fasting insulin, TC, LDL, HDL, TG, uric acid 7 strains: L. acidophilus, L. casei, L. rhamnosus, L. bulgaricus, B. longum, B. breve, S. thermophilus Decreased: hsCRP (-51%)
Asemi et al, 2014 (52)	OB & DM2, M/F, n = 62	6 DB, CO	Inulin 1.08 g + L. sporogenes, 2.7 × 10 ⁸ CFU, n = 62	PL, n = 62	Increased: GSH (46%), uric acid (12%) NS: FBG, fasting insulin, TC, LDL, HDL, TG Decreased: LDL, CRP, TNF-α, LPS
Malaguarnera et al, 2012 (53)	OW/OB & NASH, M/F, n = 66	24 DB	FOS 2.5 g + B. longum W11, 5 × 10 ⁹ CFU, n = 34	FOS 2.5 g + PL, n = 32	NS: BMI, FBG, Fasting insulin, C-peptide, HOMA-IR, TC, HDL, TG Decreased: steatosis (by liver biopsy)

10.2 Appendix 2

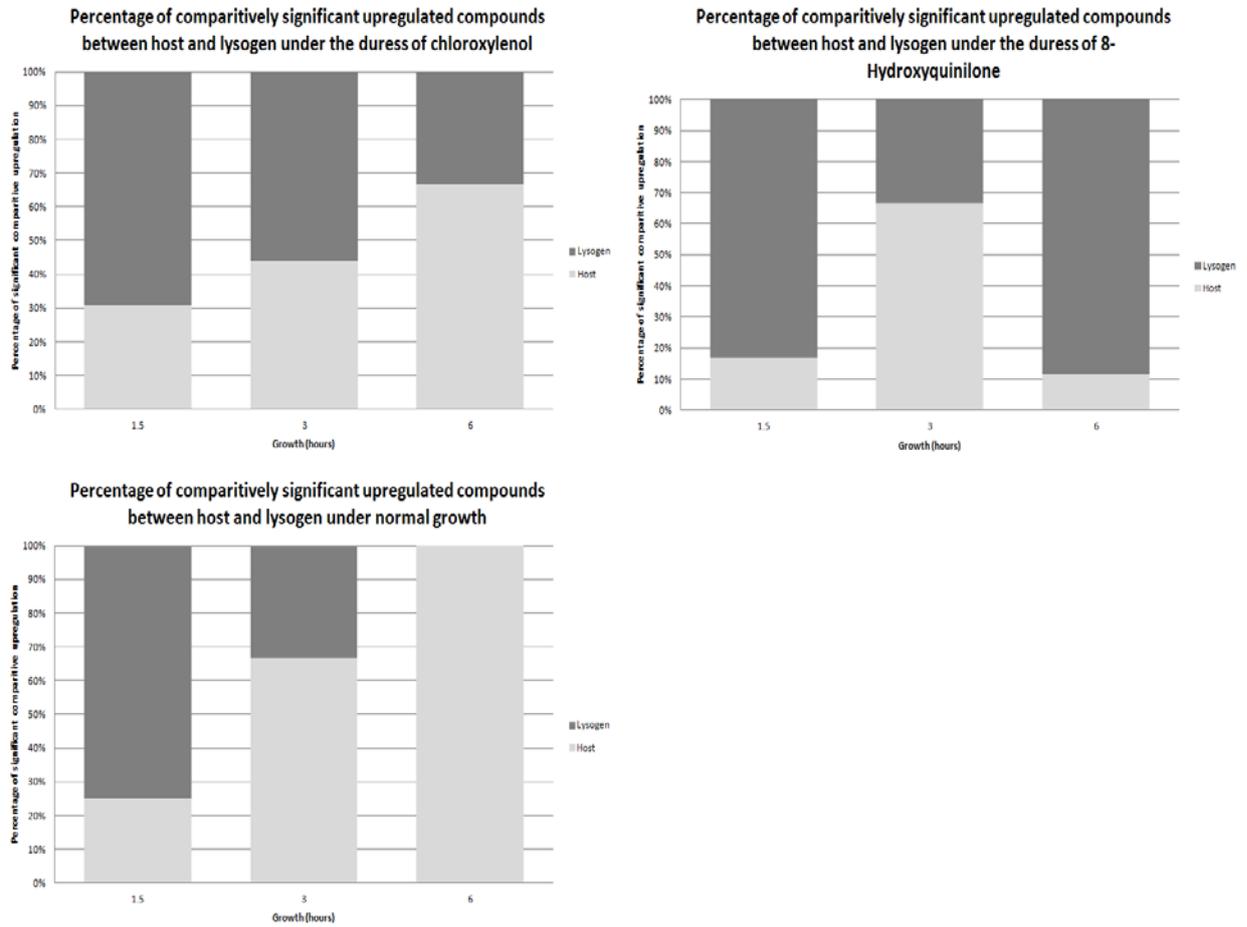


Figure 10.5 Percentage differences of metabolites present in MC1061 and ϕ 24B, where incidence is significantly higher (P value <0.05), sampled during growth, under test 8-hydroxyquinoline and chloroxylenol.

Table 10.6 Putative metabolite identities and statistics.

Compound/Retention time/Mz	Putative ID	mass error ppm	Fragment peak matches	No. of fragments matching top reference fragments	Adduct of compound	Formula of compound	isotope similarity	Anova (P)	Max Fold Change	Up-regulated
1 1.71_217.86 47m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00163439	1.191	Host
2 9.37_173.08 06m/z	2-PROpylglutanic acid	-7.5	7	0	M-H	C8H14O4	95.03	0.00063323	1.452	Lysogen
3 11.62_742.1 533n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.03458018	1.364	Lysogen
4 11.80_293.1 760m/z	Myrsinone	0.59	6	2	M-H	C17H26O4	95.9	0.00002419	1.201	Lysogen
5 12.83_384.1 933n	ARMI	-1.1	10	4	M+NA or M+K or M+H	C23H28O5	93.42	0.00000001	1.892	Host
6 10.12_273.1 951n	HEPTAN	3.82	5	0	M+H or M+Na	C14H27NO4	96.95	0.00056996	1.182	Lysogen
7 10.51_287.2 924n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00008937	1.23	Lysogen
8 11.13_772.3 460n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.000054	1.253	Lysogen
9 6.62_192.13 64n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00000048	1.341	Lysogen
10 9.71_176.18 08n	CANAV	510	0	0	M+Na or M+K	C5H12N4O3	95.5	0.00010806	1.639	Host
11 1.85_161.08 10m/z	Ethyl	1.28	4	0	M+H	C7H12O4	91.78	0.00120174	1.161	Lysogen
12 11.96_420.3 836m/z	EUROCOYL	-0.1	3	2	M+ACN+H	C25H46O2	95.49	0.00000001	1.428	Lysogen
13 2.14_220.15 46m/z	MIGLUSTAT	1.1	12	3	M+H	C10H21NO4	31.1	0.00007761	1.136	Host
14 8.93_277.12 91m/z	Pentoxifylline	-5.4	1	0	M-H	C13H18N4O3	88.77	0.03615228	1.074	Host
15 6.00_174.03 96m/z	FAPy-Adenine	-1	2	0	M+Na-2H	C5H7N5O	80.57	0.00190075	1.426	Host
16 12.17_381.0 798n	4-(((2,6-dichlorophenyl)carbonyl)amino)-N-piperidin-4-yl-1H-pyrazole-3-carboxamide	10.3	3	1	M+H or M+Na or M+K	C16H17Cl2N5O2	90.83	0.0244585	1.06	Lysogen
5 12.83_384.1 933n	ARM	-1	10	4	M+NA or M+K or M+H	C23H28O5	93.52	0.00672773	1.28	Host
17 11.93_252.1 728n	Epioxylubimin	1.05	25	2	M+H OR M+Na	C15H24O3	93.23	0.00453567	1.142	Host
18 11.79_295.1 910m/z	9-DECENO	0.62	5	0	M+K	C15H30NO2	96.92	0.0142806	1.091	Lysogen
18 11.79_295.1 910m/z	Gingerol	1.99	10	3	M+H	C17H26O4	96.92			
19 11.95_233.1 540m/z	Turmeronol B	1.53	9	2	M+H	C15H20O2	75.12	0.00010829	1.779	Host
19 11.95_233.1 540m/z	1,3,11(13)-EUD	1.53	9	1	M+H	C15H20O2	75.12			
20 6.11_192.09 95m/z	Epsilon-heptenoic acid	0.17	3	0	M+ACN+Na	C7H12O2	92.2	0.04028174	1.084	Lysogen
12 11.97_420.3 837m/z	Erucoylacetone	0.2	3	2	M+ACN+H	C25H46O2	95.73	0.00991001	1.259	Lysogen
21 11.12_370.0 693m/z	Citrasine	1.6	8	3	M+K	C17H17NO6	95.32	0.00070557	1.109	Lysogen

22	10.25_272.2 594m/z	hexadecanoic acid	3.53	16	4	M+H	C16H33NO2	94.86	0.01076165	1.08	Lysogen
23	9.63_286.17 62m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00031671	2.413	Host
24	7.61_311.18 32m/z	n5-Pan	-38	0	0	M+Na	C14H28N2O4	84.31	0.00069227	2.311	Host
25	11.12_368.0 724m/z	Alpha-Methylene Adenosine Monophosphate	-1.9	7	2	M+Na	C11H16N5O6P	95.36	0.00056106	1.09	Lysogen
26	10.78_286.2 846m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.01460937	1.164	Lysogen
27	10.68_288.2 895m/z	Sphinganine	-0.6	15	5	M+H	C17H37NO2	97.63	0.00441829	1.135	Lysogen
2A	8.37_174.08 79n	Ethyladipic acid or 2-Propylglutaric acid	-7.7	8	1	M-H, or M+Na-2H	C8H14O4	95.79	0.00478386	1.847	Host
28	10.79_278.9 872m/z	2-Keto-3-deoxy-6-phosphogluconic acid	-6.1	8	1	M+Na-2H	C6H11O9P	92.2	0.02553743	1.086	Host
28 (2 nd ID)	10.79_278.9 872m/z	ETHOXZ	-3	2	0	M+Na-2H	C9H10N2O3S2	81.06			
29	10.93_256.9 850m/z	6-phosphono	-85	5	0	M-H	C6H11O9P	94.62	0.0335785	1.084	Host
30	11.17_346.9 947m/z	2-(ALPHA	-126	5	0	M-H	C9H17O12P	88.02	0.0138194	1.101	Host
31	9.88_378.17 72m/z	3,3-DIMETHYL	93.2	2	0	M-H	C16H21N5O6	88.13	0	1.621	Lysogen
32	2.82_236.07 82m/z	8-[(Amino	-1	3	0	M-H	C9H19NO2S2	85.57	0.00000008	1.591	Host
32	2.82_236.07 82m/z	N-(1-Deoxy-1-fructosyl)glycine	2.69	5	0	M-H	C8H15NO7	94.85			
33	5.28_309.11 89m/z	lmazam	-11	1	0	M+Na-2H	C16H20N2O3	86.45	0.03600555	1.39	Lysogen
33	5.28_309.11 89m/z	Desloratad	8.12	0	0	M-H	C19H19ClN2	63.92			
34	1.42_257.07 76m/z	Imidazoleacetic acid riboside	-1.2	7	1	M-H	C10H14N2O6	94.13	0.03095166	1.094	Host
35	1.93_212.97 22n	2 AMINO or L ASPARTYL	-149	0	0	M+H or M+ACN+H or M+Na	C4H8NO7P	94.5	0.00477583	1.119	Host
36	10.32_268.1 505n	Isoleucyl-Histidine or Histidinyl-isoleucine or Histidinyl-Leucine or Leucyl-Histidine	-12	7	0	M+H or M+Na	C12H20N4O3	92.25	0.00000002	1.607	Lysogen
37	1.83_259.09 26m/z	5-Methyluridine	0.45	7	2	M+H	C10H14N2O6	96.5	0.04657029	1.058	Host
38	9.29_251.16 47m/z	1HYDROXY	2.19	7	0	M+H	C15H22O3	95.8	0.00000007	1.618	Lysogen
39	13.36_1020. 0887n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0002051	1.252	Lysogen
40	9.13_514.26 25n	GLYCINO	11.3	2	1	M-H or M+Na-2H	C29H38O8	84.79	0.00259487	1.122	Lysogen
41	13.52_297.1 529m/z	GRAVEL	11	4	1	M-H	C19H22O3	94.99	0.00674144	1.263	Host

43	12.68_733.0 280m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00924811	1.111	Lysogen
44	2.50_175.14 43m/z	N-Dimethyl-lysine	1.08	6	0	M+H	C8H18N2O2	97.67	0.03036812	1.572	Lysogen
45	1.98_186.01 28n	3-PHOSPHO	107	4	0	M-H or M+Na-2H	C3H7O7P	97.91	0.00021209	1.152	Host
4	11.82_293.1 760m/z	Myrsinone	0.71	6	2	M-H	C17H26O4	95.65	0.03289389	1.068	Host
46	10.85_179.1 065m/z	3 POSS ID'S	147	0	0	M-H	C6H14NO5	92.2	0.03020868	1.118	Host
47	5.93_252.15 75n	2-(2-{2-[2-(2-Methoxy-Ethoxy)-Ethoxy]-Ethoxy}-Ethoxy)-Ethanol	0.86	5	0	M+H or M+Na	C11H24O6	92.56	0.03127143	1.085	Host
48	8.16_286.16 52m/z	POLYETHYLENE	1.3	8	1	M+ACN+H	C12H20O5	88.26	0.00000068	1.304	Lysogen
49	7.75_203.12 80m/z	SEBACIC ACID	1.31	5	0	M+H	C10H18O4	88.75	0.000017	1.226	Lysogen
50	4.72_1100.3 683n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00000026	1.56	Lysogen
51	9.47_454.24 08n	Ethyl Cellulose	-1.3	4	0	M-H or M+Na-2H	C20H38O11	92.02	0.00000026	1.56	Lysogen
39	13.36_1020. 0885n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.01735946	1.051	Lysogen
52	1.42_365.04 28n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.02345941	1.118	Host
53	7.71_209.04 45m/z	3-(3,4-Dihydroxy-5-methoxy)-2-propenoic acid	-4.9	2	0	M-H	C10H10O5	95.49	0.00007118	1.223	Lysogen
54	10.64_264.9 911m/z	RU78299	11.4	5	1	M+Na-2H	C9H9O6P	87.27	0.03671033	1.144	Lysogen
55	11.05_253.1 440m/z	GALACTO	201	2	0	M-H	C9H18O8	97.51	0.02875719	1.052	Host
56	6.45_259.11 84m/z	Glycerol tripropanoate	-1	0	0	M-H	C12H20O6	90.27	0.00794597	1.089	Host
15	5.96_174.03 95m/z	Diureido-Acetate	0.43	0	0	M-H	C4H7N4O4	81	0.00000763	2.459	Lysogen
57	9.34_328.17 65m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00000854	1.634	Lysogen
58	1.42_459.94 96m/z	POLYTHIAZIDE	9.18	3	0	M+Na-2H	C11H13ClF3 N3O4S3	58.45	0.00007997	2.68	Lysogen
59	9.67_297.15 86n	3-[(4-AMINO-1-TERT-BUTYL-1H-PYRAZOLO[3,4-D]PYRIMIDIN-3-YL)METHYL]PHENOL	-1.4	5	0	M+H or M+Na	C16H19N5O	92.81	0.00000037	1.913	Lysogen
60	9.89_242.15 05n	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00000754	1.899	Lysogen
61	1.43_289.12 77n	Ophthalmic acid	1.31	7	0	M+H or M+Na or M+K	C11H19N3O 6	95.35	0.0000012	1.327	Lysogen
38	9.77_251.16 49m/z	N1-ACTEYL-103.12		0	0	M+ACN+Na	C9H21N3O	97.4	0.00003191	2.204	Lysogen
62	9.29_402.28 60m/z	Sphingofungin F	2.62	5	0	M+H	C21H39NO6	88.38	0.00004939	2.35	Lysogen

63	4.45_843.29 12m/z	3-Sialyl Lewis	6.57	14	2	M+Na	C31H52N2O 23	94.34	0.04643126	1.1	Lysogen
64	9.71_269.17 55m/z	Capryloylcholine	1.4	5	2	M+K	C13H28NO2	96.32	0.00001729	2.086	Lysogen
S1	10.26_411.1 987m/z	icariside B8	-0.5	11	1	M+Na	C19H32O8	84.89	0.03032644	1.157	Host
S2	10.26_277.1 407m/z	1-[(2-Amino-6,9-Dihydro-1h-Purin-6-Yl)Oxy]-3-Methyl-2-Butanol	0	2	0	M+ACN+H	C10H13N5O 2	91.1	0.02763331	1.171	Lysogen
S3	9.65_389.18 27m/z	(-)-11-hydroxy-9,10-dihydrojasmonic acid 11-beta-D-glucoside	2.58	4	0	M-H	C18H30O9	83.64	0.04213067	1.095	Lysogen
S4	10.68_207.1 020m/z	tuberonic acid or 12-hydroxyjasmonic acid or epi-4'-hydroxyjasmonic acid	-2.9	8	0	M-H2O-H	C12H18O4	94.4	0.02510448	1.105	Lysogen
S5	10.47_240.1 956m/z	RISHITIN	-1	27	1	M+NH4	C14H22O2	93.93	0.0392768	1.804	Lysogen
S6	11.29_223.1 005m/z	1-METHYL-6-PHENYL-1h-IMIDAZOL[4,5-B]PYRIDIN-2-AMINE	7.26	0	0	M-H	C13H12N4	86.21	0.02797408	1.113	Host
S7	1.29_348.89 67m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.04022452	1.273	Host
S8	1.37_178.10 75m/z	Pimelic acid	0.56	8	2	M+NH4	C7H12O4	67.66	0.02768047	1.291	Host
S9	1.35_164.09 19m/z	3-DEOXY-d-GLUCOSAMINE	0.82	14	0	M+H	C6H13NO4	88.26	0.03943001	1.385	Host
S10	11.63_467.0 141m/z	deoxyuridine triphosphate	102	0	0	M-H	C9H15N2O1 4P3	88.23	0.01819624	1.072	Host
S11	11.65_473.0 048m/z	ALLURA RED AC	-10	6	1	M+Na-2H	C18H16N2O 8S2	75.97	0.00211876	1.078	Host
S12	12.24_555.0 084m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00358126	1.082	Host
S13	11.45_455.0 141m/z	HALOSULFURON-METHYL	-3.9	2	0	M+Na-2H	C13H15CIN6 O7S	63.46	0.01500173	1.064	Host
S14	10.44_252.9 904m/z	RU78300	8.8	8	0	M+Na-2H	C8H9O6P	90.34	0.02427095	1.077	Host
S15	10.58_341.0 050m/z	1,6-Di-O-Phosphono-D-Allitol (or mannitol)	1.73	2	0	M-H	C6H16O12P 2	94.13	0.00830116	1.079	Host
S16	11.43_469.0 292m/z	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00229216	1.091	Host

Table details the 80 metabolites that have been deemed significant using CV percentage, chromatogram peaks, statistical analysis and confirmed using MS-MS and fragmentation analysis under antimicrobial challenge of MC1061 and Lysogen. The 16 metabolites (S1-S16) determined to be different metabolites that discriminate between MC1061 and Lysogen under

standard laboratory growth conditions. Table lists retention times and Mz from the column. It also includes putative compound ID based from comparison searches against pre-determined databases. It also determines fragment scores of MS-MS and the level of change in abundance whether associated with MC1061 (host) or the Lysogen (phage).

Table 10.7 T-test

Cell Growth (single and double lysogen compared to uninfected host)				
Time (h)	Single or double lysogen	P value	Test type	
1	Φ24 _B Kan	<0.001	Paired T-test (SPSS)	
	Φ24 _B KanCat	<0.001		
1.5	Φ24 _B Kan	<0.001		
	Φ24 _B KanCat	0.016		
2.5	Φ24 _B Kan	<0.001		
	Φ24 _B KanCat	<0.001		
3	Φ24 _B Kan	0.014		
	Φ24 _B KanCat	<0.001		
4	Φ24 _B Kan	<0.001		
	Φ24 _B KanCat	<0.001		
5	Φ24 _B Kan	<0.001		
	Φ24 _B KanCat	<0.001		
6	Φ24 _B Kan	<0.001		
	Φ24 _B KanCat	<0.001		
7	Φ24 _B Kan	0.02		
	Φ24 _B KanCat	<0.001		
SIC				
Condition	Concentration (μmolar)	P value	Test type	
8-Hydroxyquinoline	27.6	0.019	Independent samples T-test (SPSS)	
	29.3	<0.001		
	31	<0.001		
	32.72	<0.001		
	34.4	<0.001		
	36.2	<0.001		
Chloroxyleneol (4-chlor-3,5-dimethylphenol)	28.7	<0.001		
	35.1	<0.001		
	41.5	0.023		
Oxolinic acid	0.038	<0.001		
	0.057	<0.001		
	0.077	<0.001		
	0.096	<0.001		
	0.115	<0.001		
	0.134	<0.001		
FAPy-adenine				
Condition	Growth phase	P value		Test type
8-Hydroxyquinoline	Early	0.45		Paired T-test (SPSS)
	Mid	0.93		
	Stationary	<0.001		
Chloroxyleneol (4-chlor-3,5-dimethylphenol)	Early	<0.001		
	Mid	0.002		
	Stationary	0.009		
Pimelic acid				
Condition	Growth phase	P value	Test type	
8-Hydroxyquinoline	Early	0.02	Paired T-test (SPSS)	
	Mid	N/A		
	Stationary	0.012		
Chloroxyleneol (4-chlor-3,5-dimethylphenol)	Early	<0.001		
	Mid	0.07		
	Stationary	N/A		

Table 10.8 Statistically significant differences using area under the curve

Test	Altered use of nutrient source or targeting antimicrobial where a difference was seen between lysogen and naïve MC1061	Statistical Difference AUC	Statistical difference at mid-exponential growth phase (18h)
Uridine 2'-Monophosphate	P-Source, nucleotide, pyrimidine, uracil, Phosphate	P= 0.0094	P=0.0173
8-Hydroxyquinoline	Chelator lipophilic, RNA synthesis	p=<0.0001	p=<0.0001
Chloroxylonol	Fungicide	p=0.0032	p=0.0037
Cefoxitin	wall, cephalosporin second generation	p=0.015	p=0.0361
Puromycin	protein synthesis, 30S ribosomal subunit, premature chanin termination	p=0.0168	p=0.154
Niaproof	membrane, detergent, anionic	P=0.0192	P=0.0132
Geneticin (G418)	protein synthesis, aminoglycoside	p=0.02	p=0.0146
Cefamandole	wall, cephalosporin second generation	p=0.0239	p=0.1031
Amoxicillin	wall, lactam	p=0.057	p=0.0342
Cefmetazole	wall, cephalosporin second generation	p=0.08	p=0.0026
Methyltriocylammonium chloride	membrane, detergent, cationic	p=0.08	p=0.777
Chlorhexidine	membrane, electron transport	p=0.12	p=0.14
Ceftriaxone	wall, cephalosporin third generation	p=0.14	p=0.0791
Phenylarsine oxide	tyrosine phosphatase	p=0.17	p=0.2293

	inhibitor		
Penicillin G	wall, lactam	p=0.25	p=0.33
Cefuroxime	wall, cephalosporin second generation	P=0.27	P=0.0174
Moxalactam	wall, lactam	p=0.59	p=0.077
Cefazolin	wall, cephalosporin first generation	p=0.61	P=0.55
β-D-Allose	C-Source, carbohydrate, pentose	P=0.0097	P=0.0001
Ofloxacin	DNA unwinding, gyrase (GN), topoisomerase (GP), fluoroquinolone	p=0.0048	p=0.0008
Oxolinic acid	DNA unwinding, gyrase (GN), topoisomerase (GP), quinolone	P=0.0066	P=0.0274
6% Potassium Chloride	osmotic sensitivity, KCl	P=0.0631	P=0.073
Lomefloxacin	DNA unwinding, gyrase (GN), topoisomerase (GP), fluoroquinolone	p=0.09	p=0.09
5% Potassium Chloride	osmotic sensitivity, KCl	P=0.0939	P=0.565
4% NaCl	osmotic sensitivity, NaCl	p=0.1042	p=0.134
3% NaCl	osmotic sensitivity, NaCl	p=0.1188	p=0.0699
Sodium salicylate	anti-capsule, biofilm inhibition, mar inducer	p=0.1384	P=0.055
Ciprofloxacin	DNA unwinding, gyrase (GN), topoisomerase (GP), fluoroquinolone	p=0.255	P=0.37
2% Sodium Lactate	osmotic sensitivity, sodium lactate	p=0.2675	p=0.0656
5% NaCl	osmotic sensitivity, NaCl	p=0.44	p=0.2456
Sodium metasilicate	toxic anion	P=0.7	p=0.11

Table 10.9 PLS-DA statistics

PLS-DA Standard conditions				
Component	Eigenvalue	R2Y(cum)	Q2	Q2(cum)
1	10.6	0.262	-0.556	-0.1
2	4.06	0.847	0.755	0.73
PLS-DA chloroxylenol conditions				
Component	Eigenvalue	R2Y(cum)	Q2	Q2(cum)
1	4	0.923	0.802	0.802
2	2.71	0.981	0.405	0.882
PLS-DA 8-hydroxyquinoline conditions				
Component	Eigenvalue	R2Y(cum)	Q2	Q2(cum)
1	3.84	0.89	0.74	0.74
2	3.75	0.967	0.533	0.879

10.3 Appendix 3

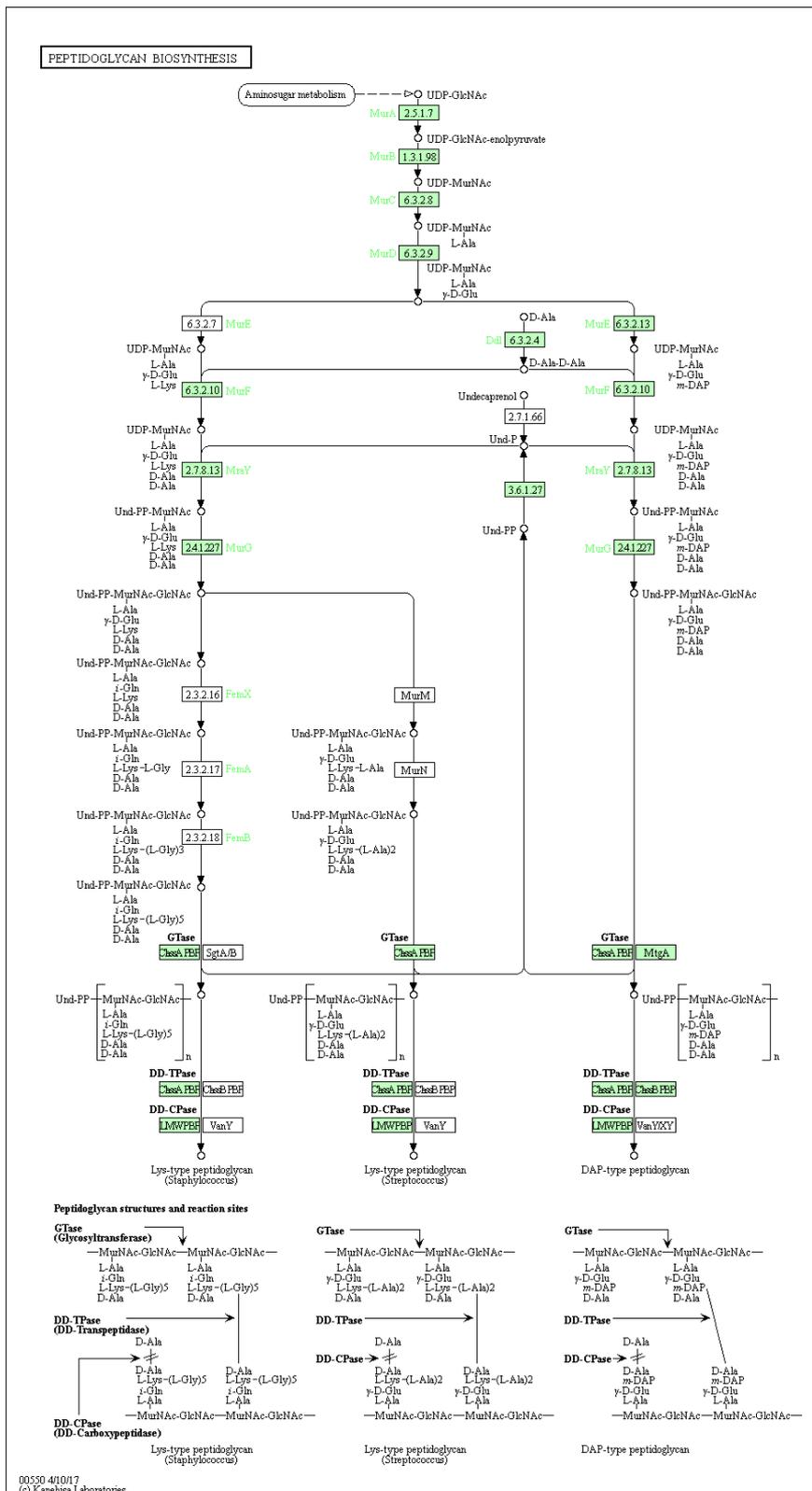


Figure 10.6 Map of the peptidoglycan pathway. Gene rectangles coloured in green are peptidoglycan pathway genes associated to k12 *E. coli*.

10.3.1 Log10 normalisation

It is noteworthy that prior to normalisation, the dominant fatty acid features in the cell wall can be clearly identified. In figure 16 (which contains all fatty acid methyl ester data) the dominant 5 fatty acids in MC1061 cell wall (regardless of any variable) are: hexadecanoic acid, Cyclopropaneoctanoic acid, 2-octyl, Cyclopropaneoctanoic acid, 2-hexyl, 9-octadecanoic acid, and tetradecanoic acid. Although fatty acids can be associated to drug type, a clearer lysogen and naïve host separation under 8-hydroxyquinoline has been observed, which is why normalisation and analyses of the drug groups is also carried out separately, allowing greater focus on the differences between lysogen and naïve host.

In figure 17A (8-hydroxyquinoline stressed) the dominant 5 fatty acids are in MC1061 cell wall (regardless of phage infection) are: Cyclopropaneoctanoic acid, 2-octyl, Cyclopropaneoctanoic acid, 2-hexyl, hexadecanoic, tetradecanoic, and dodecanoic. In figure 17B (chloroxylenol stressed) the dominant 5 fatty acids are in MC1061 cell wall (regardless of phage infection) are: Cyclopropaneoctanoic acid, 2-octyl, Cyclopropaneoctanoic acid, 2-hexyl, 9-hexadecanoic, tetradecanoic, and 9-octadecanoic acid.

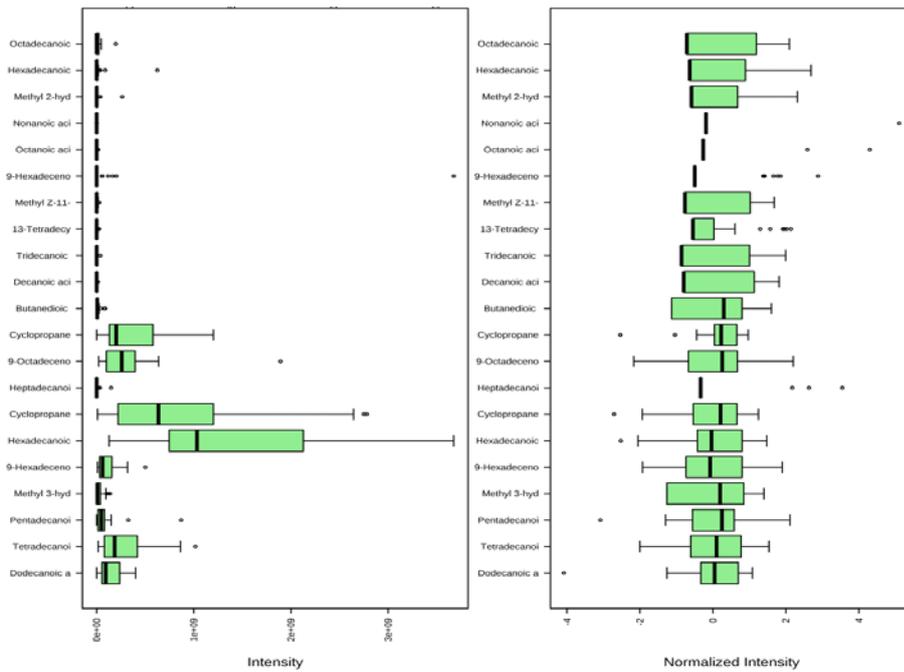


Figure 10.7 Data normalisation of fatty acid methyl esters identified over 18 hours under increasing concentrations of either antimicrobial. Plot representing the intensity and normalised intensity of fatty acids in the cell wall of both the lysogen and naïve host MC1061, grown in presence of increasing concentrations of chloroxylenol and 8-hydroxyquinoline. Samples were taken every 3 hours over an 18 hour period, each sample consisted of experimental and technical replicates (n=9).

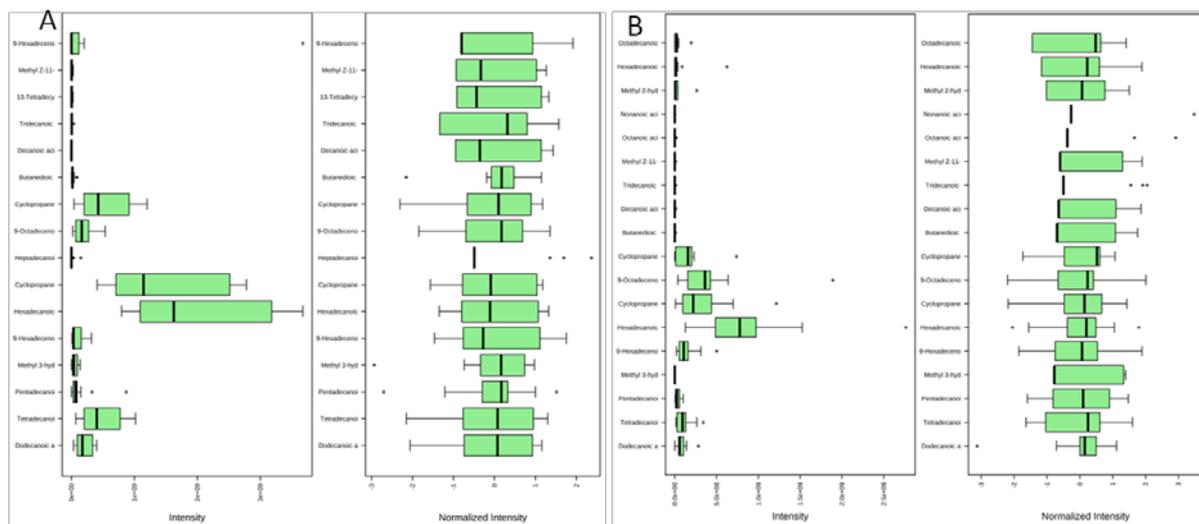


Figure 10.8 Separate normalisation of 8-hydroxyquinoline and chloroxylenol fatty acid methyl ester data. Plot representing the intensity and normalised intensity of fatty acids in the cell wall of both the lysogen and naïve host MC1061, grown in presence of increasing concentrations of 8-hydroxyquinoline (A) and chloroxylenol (B). Samples were taken every 3 hours over an 18 hour period, each sample consisted of experimental and technical replicates (n=9).

Table 10.10

Identified cell wall fatty acid methyl esters and their known functions

Fatty acid	Known functions
Dodecanoic acid, methyl ester (lauric acid), Formula: C₁₃H₂₆O₂	Changes in lauric acid effect membrane fluidity. Known to decline in low temperatures. Seems to be involved in managing temperature changes
Tetradecanoic acid, methyl ester (Myristic acid), Formula: C₁₅H₃₀O₂	Long chain fatty acid, shown to effectively kill gram negative bacteria
Pentadecanoic acid methyl ester, Formula: C₁₆H₃₂O₂	Unknown
Methyl 3-hydroxytetradecanoate, Formula: C₁₅H₃₀O₃	Unknown
9-Hexadecenoic (Palmitoleic) acid, methyl ester, (Z)- (CAS), Formula:	Essential in the membrane bilayer phospholipid.
Hexadecanoic (Palmitic) acid, methyl ester, Formula:	Essential in the membrane bilayer phospholipid. Reduction in cell wall has been previously associated to decreases in temperature
Cyclopropaneoctanoic acid, 2-hexyl-, methyl ester (CAS), Formula:	Typically produced at the onset of the stationary phase. prominent theme among the various hypotheses put forward is that CFA formation changes the fluidity or other physical properties of bacterial membranes in a biologically relevant way Associated to acid resistance in e. coli
Heptadecanoic (Margaric) acid, methyl ester (CAS), Formula:	Used in the production of lipids. Incorporated into the phospholipid structure. Can feed directly into biotin pathway.
9-Octadecenoic (oleic) acid (Z)-, methyl ester, Formula:	Free oleic acid increases the pyrimidine salvage pathway (Cdd and Udp) and specific binding-protein-dependent transport system (MalE and RbsB). The salvage pathway of E. coli functions to reutilize free bases and nucleosides produced intracellularly from nucleotide turnover. Also, the pyrimidine salvage pathway has been reported to recycle the pentose moieties of exogenous nucleosides to use them as carbon and energy sources and the amino groups of cytosine compounds as a nitrogen source. Increases membrane fluidity and pressure resistance
Cyclopropaneoctanoic acid, 2-octyl-, methyl ester (CAS), Formula:	Typically produced at the onset of the stationary phase. prominent theme among the various hypotheses put forward is that CFA formation changes the fluidity or other physical properties of bacterial membranes in a biologically relevant way
Butanedioic (succinic) acid, dimethyl ester (CAS), Formula:	Succinic acid is a dicarboxylic acid. The anion, succinate, is a component of the citric acid cycle capable of donating electrons to the electron transfer chain. SDH with a covalently attached FAD prosthetic group, binds enzyme substrates (succinate and fumarate) and physiological regulators (oxaloacetate and ATP). Oxidizing succinate links SDH to the fast-cycling Krebs cycle portion where it participates in the breakdown of acetyl-CoA throughout the whole Krebs cycle

10.3.2 Reasoning for PLS-DA modelling

PCA algorithms are unsupervised, providing less biased dimension reduction. PCA is only capable of identifying differences between groups when intra group variation is sufficiently less than variation between groups. As the complexity of our data is high, we opted to use a supervised model for analysis (PLS-DA). PLS-DA is a supervised algorithm as it does not allow for other response variables than the one for defining the groups of individuals. PLS-DA builds typologies with an intrinsic prediction power, which sharpens the separation between groups. However a supervised model such as PLS-DA can also introduce bias and false group separation, to confirm/validate differentiation observed, other methods should be used. Hierarchical clustering is one method of validation, the heatmaps presented here use the hierarchical clustering algorithm for determining separation. The PLS-DA's are further validated using R² and Q² scores (the LOOCV method of validation). Nonetheless PLS-DA can still struggle with complex data that has an over abundance of variables, this is why data has also been divided and plotted separately.

10.4 Appendix 4

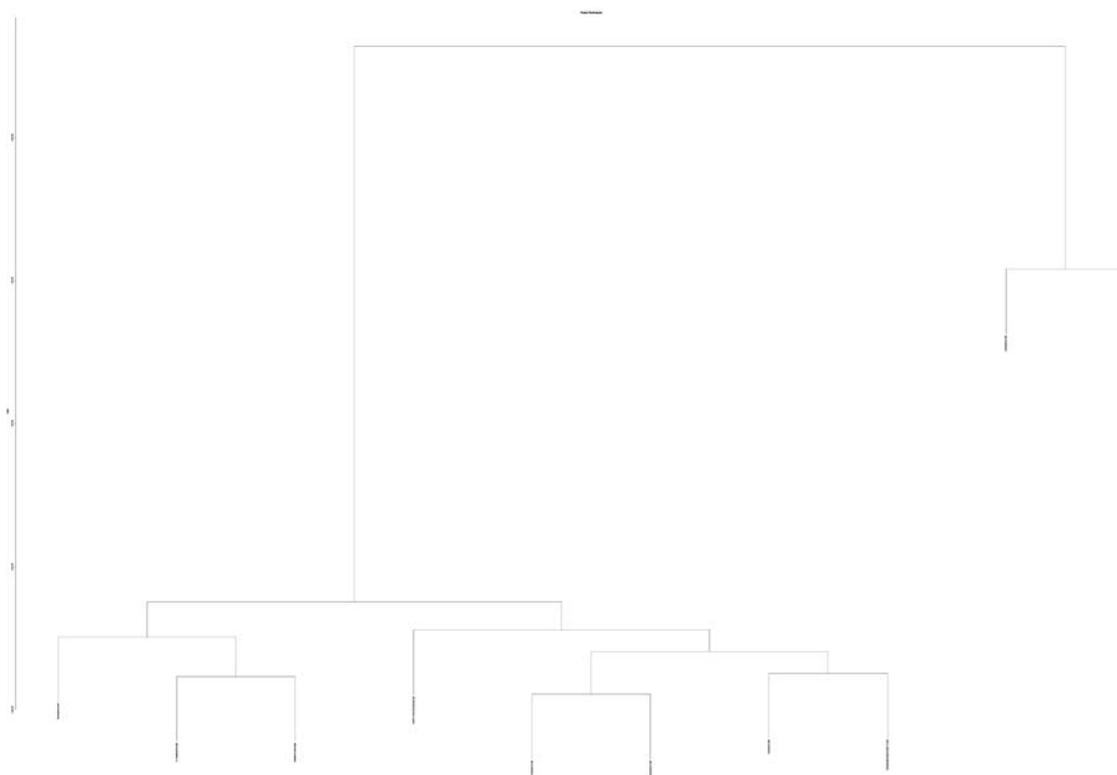


Figure 10.9 Dendrogram of CRACCD's example compound dataset. The plot demonstrates the dendrogram plot that the program CRACCD can produce after completion of step 3. Each compound is arranged along the bottom of the dendrogram, where similarities between these nodes is viewed as clusters, with each clusters similarity to another represented by the distance/branch separation between them.

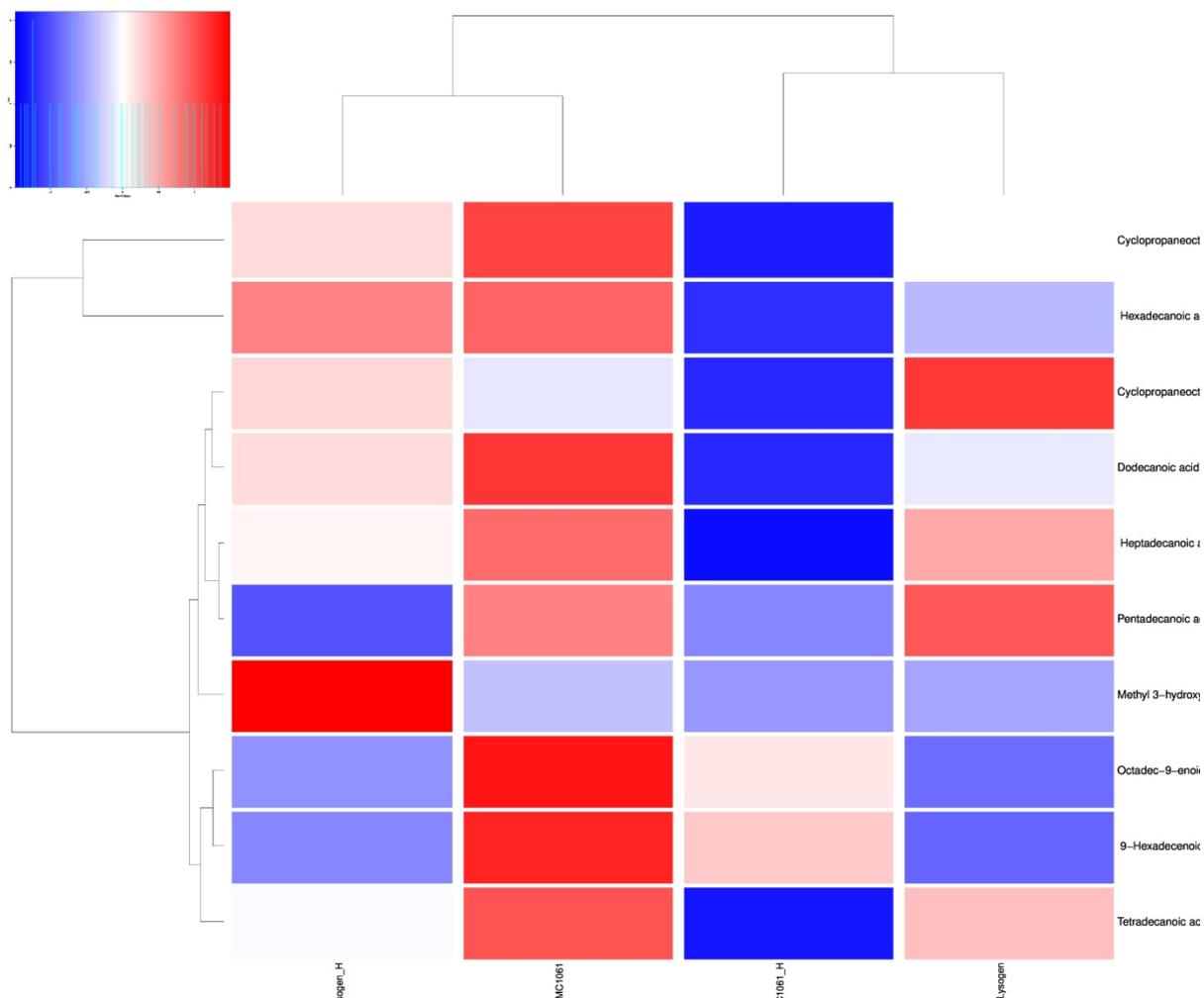


Figure 10.10 Heatmap of CRACCD's example compound dataset. The plot demonstrates the Heatmap that the program CRACCD can produce after completion of step 3. The sample type is labelled along the bottom of the heatmap, the compounds are labelled along the right of the heatmap, with corresponding dendrograms opposite the given variable type. The intensity of a given compound is represented with its associated gradient along a color scale, where blue, white and red represent low, middle and high intensity respectively.

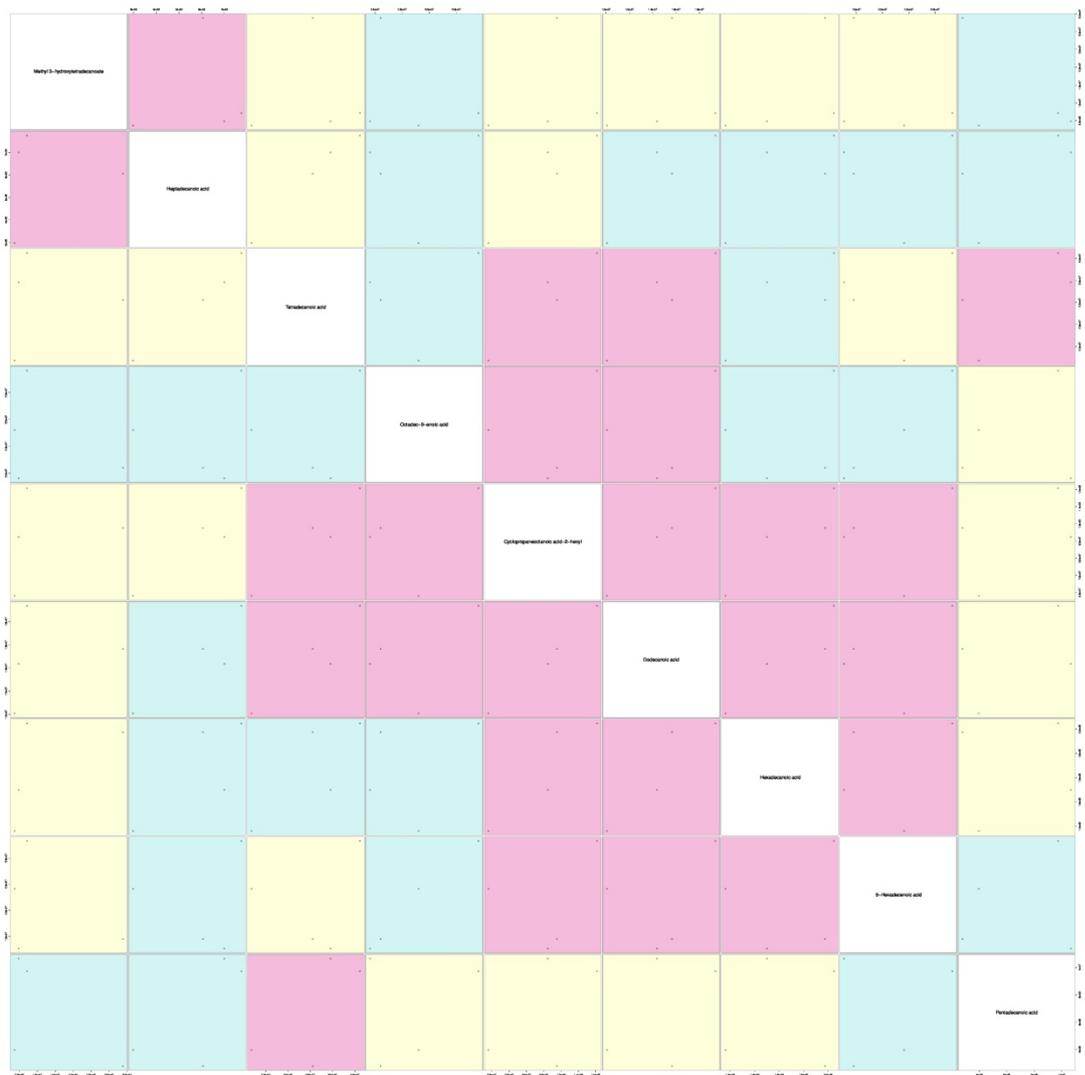


Figure 10.11 Correlation plot of CRACCD's example compound dataset. The plot demonstrates the correlation plot that the program CRACCD can produce after completion of step 3. Variables with higher correlations are closer to the principal diagonal, with color dictating the size of the correlation.

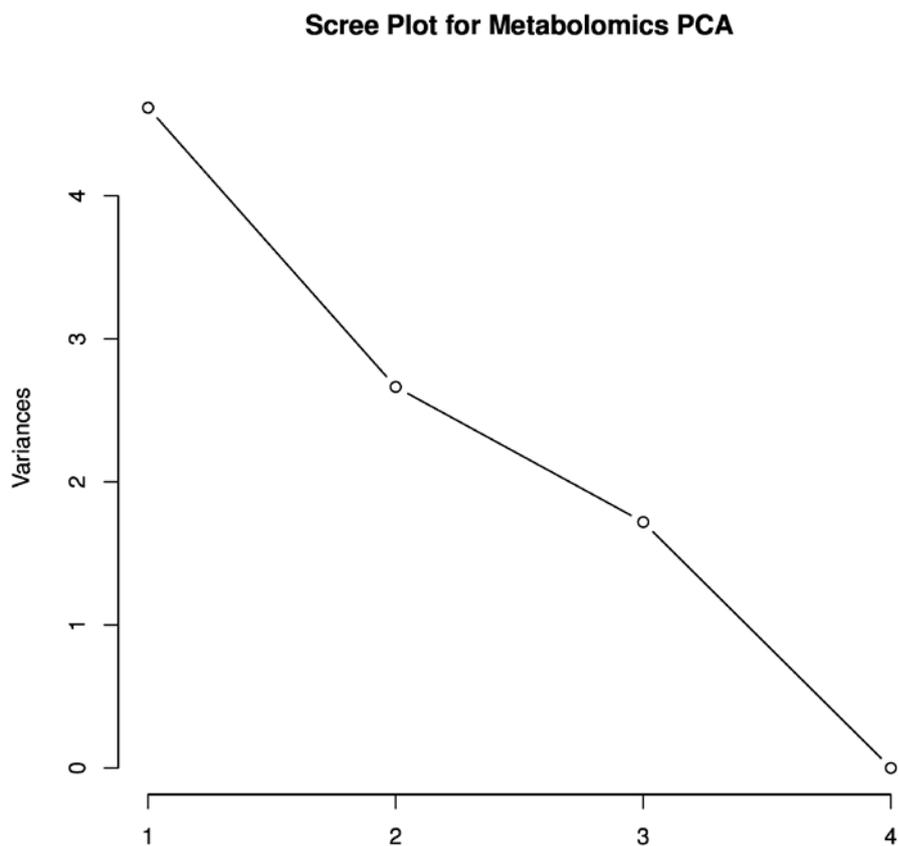


Figure 10.12 Scree plot of CRACCD's example compound dataset. The plot demonstrates the Scree plot that the program CRACCD can produce after completion of step 3. The y axis shows the eigenvalues and the x axis shows the number of factors.

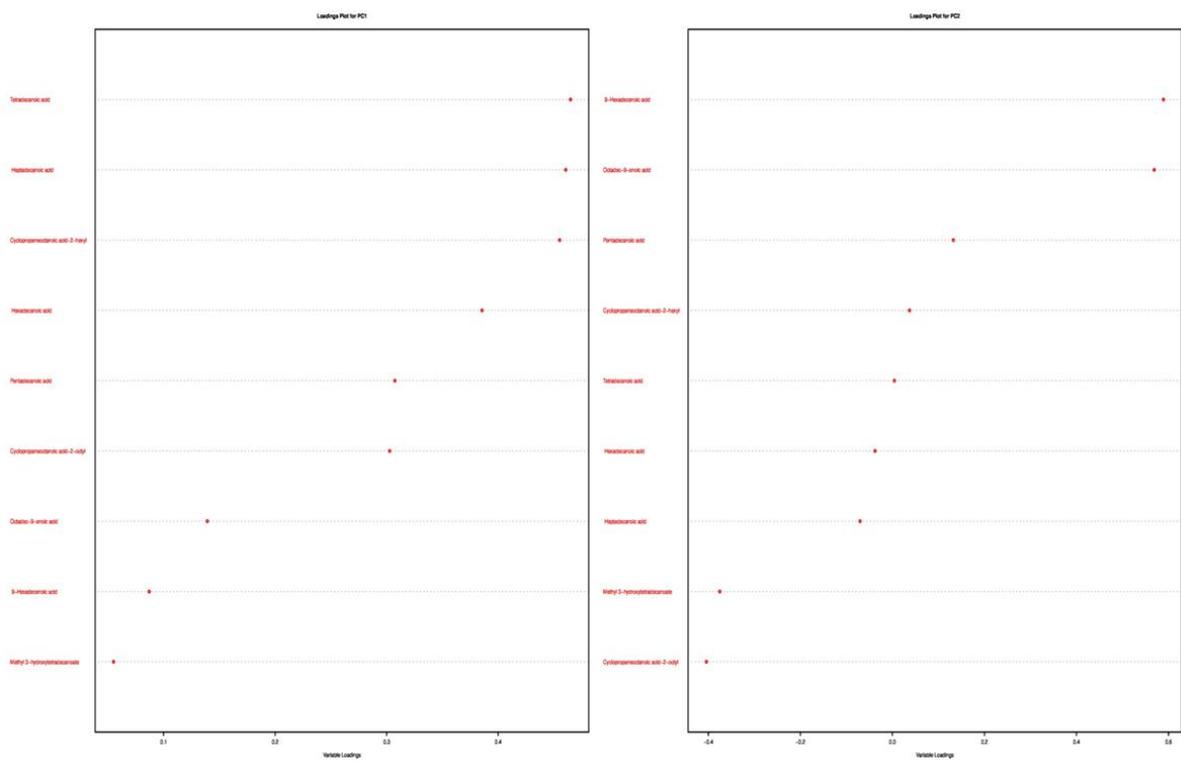


Figure 10.13 PCA Dot plot of CRACCD's example compound dataset. The plot demonstrates the dot plot that the program CRACCD can produce after completion of step 3. The x axis represents the loading values for a given variable, the variables are shown along the y axis.

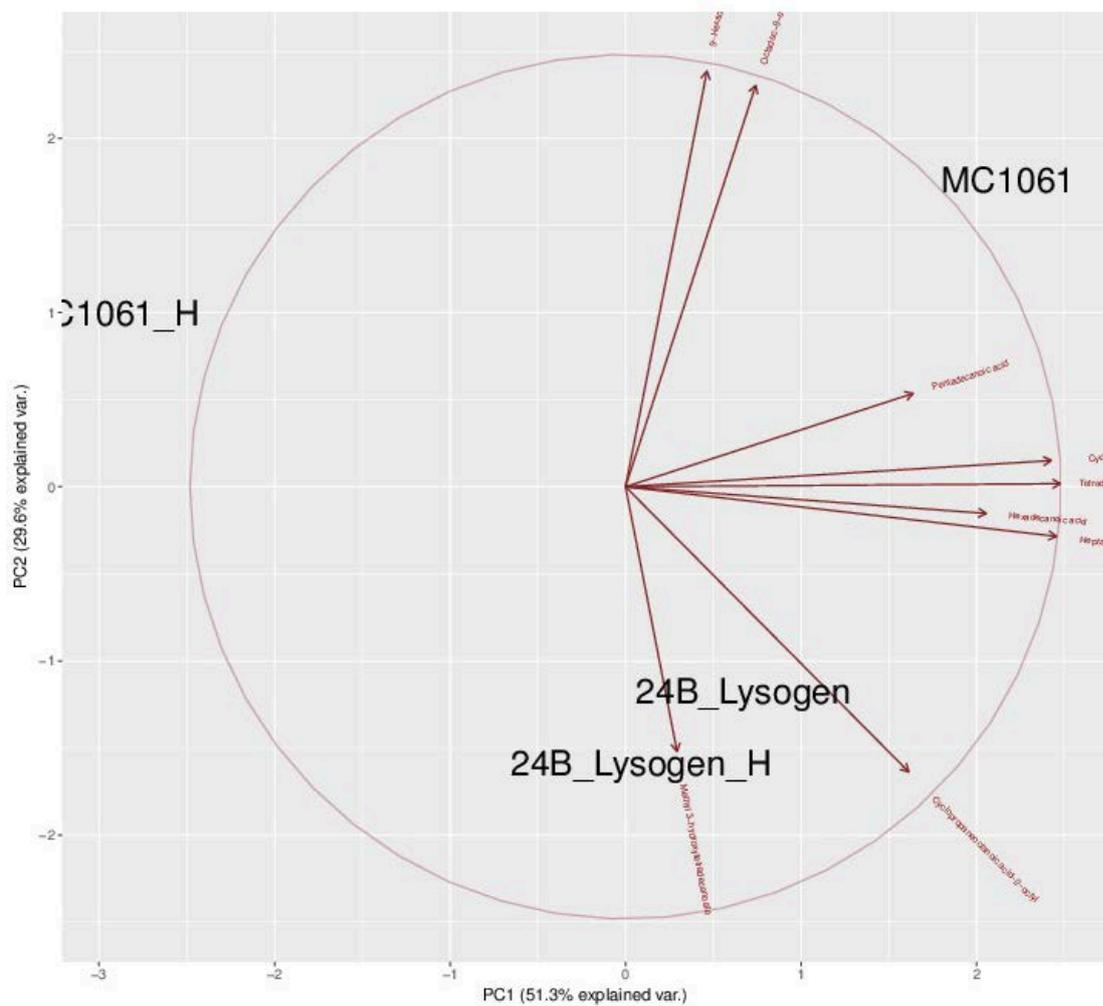


Figure 10.14 PCA biplot of CRACCD's example compound dataset. The plot demonstrates the Biplot that the program CRACCD can produce after completion of step 3. The x and y axis represent the 1st and 2nd principle components respectively, the circle is the unit circle and the arrows represent the influence of a given variable upon a sample/group.

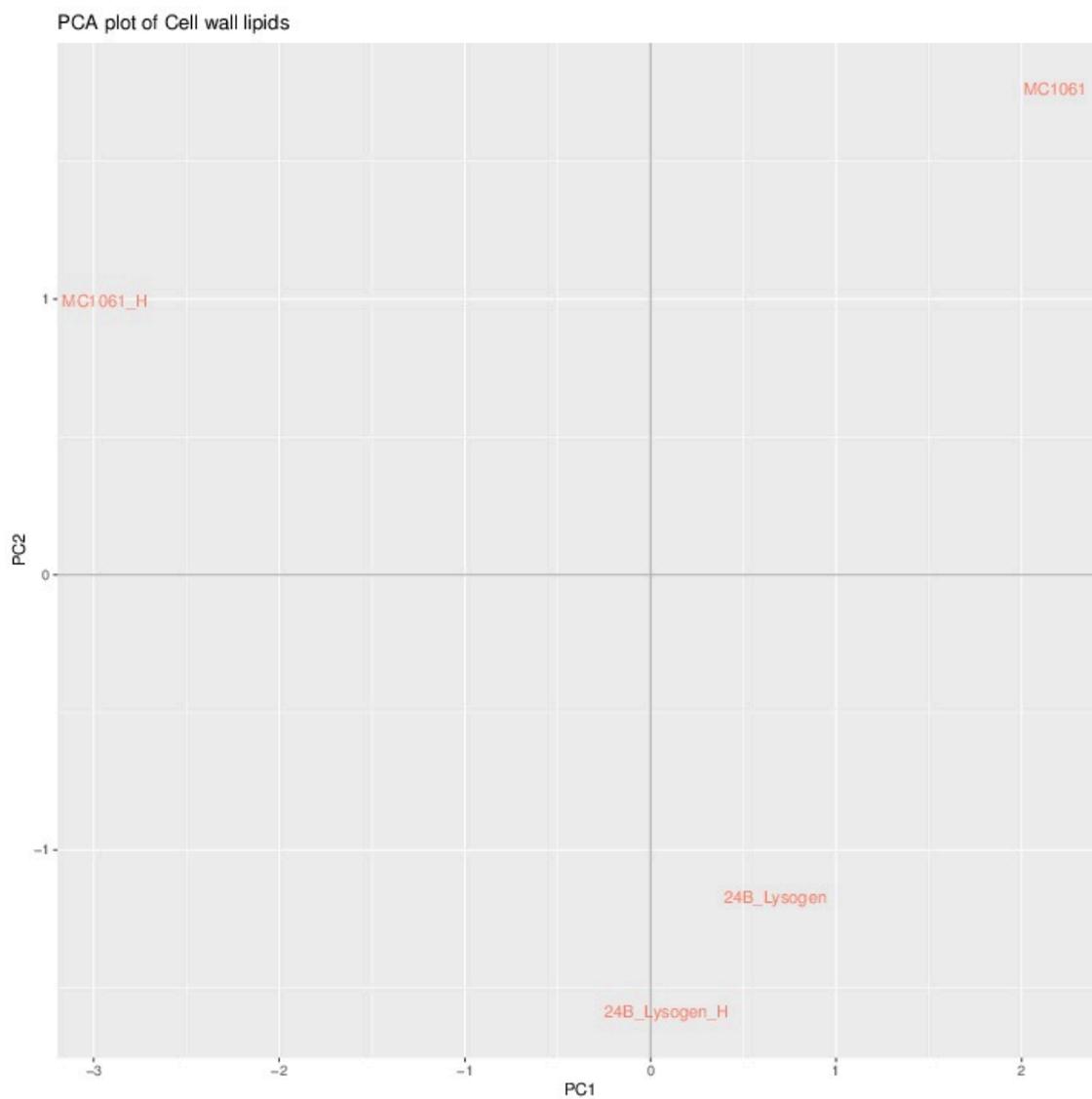


Figure 10.15 PCA of CRACCD's example compound dataset. The plot demonstrates the PCA plot that the program CRACCD can produce after completion of step 3.

10.5 Appendix 5

Table 10.11 The genomic related databases within Entrez (September 2015). Edited from (Coordinators, 2016)

Database	Records	Section within this article	Data source
SNP*	705 483 355	Genomes	D (dbSNP), N
Nucleotide*	199 827 994	Genomes	D (GenBank), C, N
GSS*	39 394 513	Genomes	D (GenBank)
Clone*	37 336 118	Genomes	D, N
Probe	32 379 570	Genomes	D
dbVar*	4 481 341	Genomes	D
BioSample	3 648 667	Genomes	D
SRA*	1 697 236	Genomes	D
Taxonomy*	1 426 896	Genomes	C, N
BioProject*	152 290	Genomes	D
Assembly*	59 566	Genomes	C, N
Genome*	13 532	Genomes	C, N
Epigenomics*	7789	Genomes	D
GEO Profiles*	108 708 851	Genes	D
EST*	75 992 479	Genes	D (GenBank)
Gene*	21 399 200	Genes	C, N
UniGene*	6 473 284	Genes	N
GEO Datasets*	1 645 202	Genes	D
PopSet*	231 877	Genes	D (GenBank)
Homologene*	141 268	Genes	N
Protein*	223 456 488	Proteins	C, N
Protein Clusters*	820 546	Proteins	N
Structure*	111 186	Proteins	C, N
CDD*	50 648	Proteins	C, N

10.6 Appendix 6

10.6.1 dsDNA viruses, replication (Baltimore classification: Type I)

There are at least 3 categories in which dsDNA viruses replicate their genomes, these are; dsDNA strand displacement (Liu, Naismith et al., 2003), dsDNA rolling circle, and bi-directional replication (Kadaja, Silla et al., 2009). Rolling circle replication is the most common method of replication in dsDNA viruses (as well as other viral types), where nearly all known dsDNA phage employ this method. Rolling circle replication starts at genomic 'origin' sites, where viral endonuclease mediated nicking occurs. On the 3' end of the DNA strand the replication machinery assembles with the DNA polymerase which begins to synthesise the DNA while DNA gyrase separates the two strands ahead of the polymerase (strand displacement synthesis). The strand displacement synthesis creates a concatemer linear ssDNA where each genomic copy is one turn of replication. Sequential RNA primer synthesis elongates Okazaki fragments on the concatemer, mediated by primase, forming dsDNA concatemer. The concatemer RNA primers are then removed and the okazaki fragments are ligated. This method can be used for both circular and linear genomes.

Table 10.12 dsDNA viral characteristics. Data collated from ICTV reports (Lefkowitz et al., 2018).

Family	Virion morphology	Symmetry type	Host	Genome	Virion diameter or WxL	Genome size
Adenoviridae	Icosahedral, non-env	Icosahedral T=pseudo25	E	L	70-90 nm	26.2-48.4 kb
Ampullaviridae	Bottle-shaped, env	-	A	L	75x230 nm	23.8 kb
Ascoviridae	bacilliform, ovoidal or allantoid, env	-	E	C	130x200-400 nm	150-190 kb
Asfarviridae	Spherical to pleomorphic, env	Icosahedral T=189-217	E	L	170-190 nm	170-190 kb
Baculoviridae	Occlusion or budded, env	-	E	C	21x260 nm	80-180 kb
Bicaudaviridae	Lemon-shaped, presumed non env	-	A	C	120x80 nm	62.7 kb
Myoviridae	Head-tail structure, contractile tail, non-env	-	B, A	L	tails: 80-455x16-20 nm	33-244 kb
Podoviridae	Head-tail structure, non-contractile tail, non-env	Icosahedral head	B	L	tails: 20x8 nm	16.0-77.6 kb
Siphoviridae	Head-tail structure, non-contractile tail, non-env	capsid Icosahedral T=7	B, A	L	tails: 65-570x7-10 nm	22.1-134.4 kb
Corticoviridae	Head-spike structure, non-contractile tail, non-env	Icosahedral T=21	B	C	57 nm	10.1 kb
Fuselloviridae	Lemon-shaped, env	-	A	C	55-60x80-100 nm	14.8-17.8 kb
Globuloviridae	Spherical, probable env	-	A	L	70-100 nm	28.3 kb
Guttaviridae	Somewhat pleiomorphic, env	-	A	C	75-90x110-185 nm	~20 kb
Alloherpesviridae	Spherical to pleomorphic, env	Icosahedral T=16	E	L	150-200 nm	134-248 kb
Herpesviridae	Spherical to pleomorphic, env	Icosahedral T=16	E	L	150-200 nm	120-240 kb
Malacoherpesviridae	Spherical to pleomorphic, env	Icosahedral T=16	E	L	150-200 nm	134 kb
Iridoviridae	-	Icosahedral T=189-217	E	L	120-350 nm	140-303 kb
Lipothrixviridae	flexible filaments (rod shaped), env	-	A	L	24-38x410-2200 nm	15.9-56 kb
Rudiviridae	Stiff rod shape, non-env	-	A	L	23x600-900 nm	24.7-35.5 kb
Mimiviridae	Roughly spherical	Icosahedral	E	L	750 nm	1181.5 kb
Nimaviridae	Ovoid or ellipsoid to bacilliform, env	-	E	C	120-150x270-290 nm	300 kb
Papillomaviridae	Non-env	capsid Icosahedral T=7	E	C	55 nm	6.8-8.4 kb
Phycodnaviridae	Env	capsid Icosahedral T=169	E	L	120-220 nm	100-560 kb
Plasmaviridae	quasi-spherical, slightly pleomorphic, env	-	B	C	50-125 nm	12 kb
Polydnaviridae	Prolate ellipsoid form or cylindrical, env	-	E	C	34-40x8-150 or 85x330 nm	190-550 kb
Polyomaviridae	Non-env	capsid Icosahedral T=7d	E	C	40-45 nm	4.7-5.3 kb
Poxviridae	somewhat pleomorphic, brick-shaped or ovoid, env	-	E	L	220-450x140-260 nm	130-375 kb
Rhizidiovirus*	round/isometric, non-env	icosahedral	E	L	60 nm	12.8 kb
Salterprovirus*	spindle-shaped, env	-	A	L	44x77 nm	14.5 kb
Tectiviridae	Non-env	Icosahedral T=25	B	L	60 nm	15 kb

10.6.2 ssDNA viruses, replication (Baltimore classification: type II)

The first major step in ssDNA viral replication (after infection) is the host DNA polymerase mediated conversion of ssDNA into dsDNA. ssDNA viral genome replication occurs via a rolling-circle mechanism, this involves the virus-encoded rolling-circle replication initiation endonuclease (RC-Rep) which carries out nicking of the viral genome (Chandler, de la Cruz et al., 2013, Krupovic, 2013, Krupovic & Forterre, 2015, Rosario, Duffy et al., 2012). The resulting transcripts (mRNA) are used for viral protein translation. Then the replicated ssDNA is converted back into dsDNA, mediated by DNA polymerase, prior to capsid packaging and host cell exit.

Table 10.13 ssDNA viral characteristics. Viraldata collated from ICTV reports (Lefkowitz et al., 2018).

Family	Virion morphology	Symmetry type	Host	Genome	Virion diameter or WxL	Genome size
-Anelloviridae	Non-env	Icosahedral T=1	E	C	30 nm	~2-3.9 kb
Circoviridae	Round, non-env	Icosahedral T=1	E	C	12-26.7 nm	~1.7-2.3 kb
+Geminiviridae	Non-env	Icosahedral T=1	E	C	22x38 nm	2.5-3.0 kb
+Inoviridae	rod of filaments, non-env	-	B	C	7x700-2000 nm	4.5-12.4 kb
+Microviridae	Round, non-env	Icosahedral T=1	B	C	~30 nm	4.4-6.1 kb
+Nanoviridae	Round, non-env	Icosahedral T=1	E	C	17-20 nm	9.2-11.1 kb
Parvoviridae	Round, isometric, non-env	Icosahedral T=1	E	L	18-26 nm	4-6.3 kb

10.6.3 dsRNA viruses (Baltimore classification: Type III)

Nearly all currently identified dsRNA viruses are eukaryotic, with the only known family of dsRNA bacterial viruses being Cystoviridae, though RNA phage diversity is suggested to be far greater (Krishnamurthy et al., 2016). Due to relationships to polymerases and proteins, it's been suggested that dsRNA eukaryotic viruses descend from this Cystoviridae dsRNA phage family (Koonin & Ilyina, 1992), as well as positive sense RNA viruses (Koonin, Gorbalenya et al., 1989). Of all the dsRNA eukaryotic viruses the Reoviridae family is most closely related to, and potentially originated from, the prokaryote dsRNA Cystoviridae family (El Omari, Sutton et al., 2013). All currently identified dsRNA viruses have an icosahedral symmetry and linear genome (see Table 10.9). Most dsRNA viruses use their icosahedral capsid to hide dsRNA replication and transcription, because dsRNA is not naturally produced by host cells. This obvious foreign molecule makes for a strong inducer of host antiviral defense mechanisms, many of these mechanisms have been identified in eukaryotes.

Replication of dsRNA viruses starts with the transcription of dsRNA by viral RNA-dependent RNA polymerase (RdRp), resulting in the synthesis of +ssRNA strands. The +ssRNA serve as mRNA templates for the translation of proteins as well as templates for -ssRNA synthesis. The -ssRNA synthesis on the template +ssRNA forms the copies of dsRNA.

Table 10.14 dsRNA viral characteristics.Viraldata collated from ICTV reports(Lefkowitz et al., 2018).

Family	Virion morphology	Symmetry type	Host	Genome	Virion diameter or WxL	Genome size
Birnaviridae	Single-shelled, non-env	Icosahedral T=13	E	L	~65 nm	~6 kb
Chrysoviridae	Isometric, non-env	Icosahedral T=1	E	L	35-40 nm	12.5 kb
Cystoviridae	Spherical, env	Icosahedral T=13	B	L	~85 nm	12.7–15.0 kb
Endornaviridae	No virions	-	E	L	-	14-17.6 kb
Partitiviridae	Isometric, non-env	Icosahedral T=2*	E	L	30-43 nm	~4 kb
Picobirnaviridae	Isometric, non-env	Icosahedral T=2*	E	L	33-37 nm	~4 kb
Reoviridae	Icosahedral/appear spherical, non-env	Icosahedral T=13	E	L	60–80 nm	18.2-30.5 kb
Totiviridae	Isometric, icosahedral, non-env	Icosahedral T=2*	E	L	40 nm	4.6-7.0 kb

10.6.4 ssRNA viruses (Baltimore classification: Type IV and V)

ssRNA viruses can be either positive (+) (type IV) or negative (-) (type V) sense, positive or negative depending on the polarity or sense of the RNA, i.e. 3'-5' (-) and 5'-3' (+). ssRNA viral hosts are almost entirely eukaryotic, the only known ssRNA viral family that infects and replicates in a bacterial host cell is the Leviviridae family, which infect primarily enterobacteria and some other proteobacteria (Bollback & Huelsenbeck, 2001). ssRNA viral characteristics can be seen in Table 10.11.

Unlike -ssRNA viruses, the +ssRNA viral genome can also function as mRNA, as it can be directly translated into viral proteins in the host cell, thereby functioning as a template for both translation and replication. An example of this is seen in the +ssRNA viral family Coronaviridae. Genomes within the Coronaviridae family act as mRNA, synthesising proteins within their host without the need of a complementary RNA intermediate, as a direct result of this their virions don't need to be packaged with RNA polymerase. The -ssRNA viral genome is complementary to its mRNA, because of this the -ssRNA is converted with RdRp to make +ssRNA, the +ssRNA form of the genome can function as mRNA, which is then translated.

+ssRNA viral genomes are transcribed by viral RNA RdRp from a dsRNA template. After infection of the host cell by +ssRNA viruses the first proteins expressed chaperone the genome to, and help form, the viral replication complex. The viral replication complex is made up of both viral and host proteins, the host proteins include chaperone proteins, RNA-binding proteins, and lipid synthesis

and membrane remodeling proteins. The lipid synthesis and membrane remodeling proteins play a role in forming the membranous vesicles in which the viruses transcription and translations occurs (Shulla & Randall, 2016). The major contributors to these replication compartments are derived from host Golgi, endoplasmic reticulum, endosomes, plasma membrane, peroxisomes, and mitochondria (Miller & Krijnse-Locker, 2008). +ssRNA replication is carried out through dsRNA intermediates, where the resulting copies of ssRNA can be further replicated or translated. The necessity of membranous vesicle is likely due to dsRNA acting as a strong inducer of anti-viral defense mechanisms (Shulla & Randall, 2016).

In -ssRNA viruses the -ssRNA is not formed into a dsRNA prior to replication (Weber, Wagner et al., 2006), this is unique to -ssRNA, it is instead used as a direct template. As described above the RdRp complex directly converts -ssRNA into +ssRNA, which functions as both mRNA and replication template. The +ssRNA versions of the genome can be used create replicate copies of the -ssRNA genome via the same process, using the RdRp complex. This method of replication requires unique mechanisms, which include “Poly A stuttering” and “Cap snatching” (Hulo, Masson et al., 2017).

Table 10.15 ssRNA viral characteristics. Viraldata collated from ICTV reports (Lefkowitz et al., 2018).

Family	Virion morphology	Symmetry type	Host Genome		Virion diameter or WxL	Genome size
+Astroviridae	Spherical, non-env	Icosahedral T=3	E	L	28-41 nm	6.4-7.7 kb
+Barnaviridae	Bacilliform, non-env	Icosahedral T=1	E	L	18-20x48-53 nm	4 kb
+Benyavirus*	Rod shaped, non-env	-	E	L	65-390x20 nm	1.3-6.7 kb
+Bromoviridae	Spherical/quasi-spherical, non-env	icosahedral or bacilliform	E	L	26-35 or 18-26x30-85 nm	~8 kb
+Caliciviridae	Non-env	Icosahedral T=3	E	L	~27-40 nm	7.4-8.3 kb
+Cileivirus*	Bacilliform, non-env	-	E	L	120-130x50-55 nm	4.7-5 kb
+Closteroviridae	Helical filaments, non-env	-	E	L	12 nm x 650-2000 nm	14.5-19.3
+Flaviviridae	Spherical, env	Icosahedral-like	E	L	40-60 nm	9.6-12.3 kb
+Hepeviridae	Spherical, non-env	Icosahedral T=1	E	L	27-34 nm	6.6-7.2 kb
+Hypoviridae	Vesicle, no virion	-	E	L	50-80 nm	9-13 kb
+Idaeovirus*	Slightly flattened, Isometric, non-env	Icosahedral	E	L	~33 nm	1-5.5 kb
+Leviviridae	Spherical, non-env	Icosahedral T=3	B	L	26 nm	3.5-4.3 kb
+Luteoviridae	Hexagonal, non-env	Icosahedral T=3	E	L	25-30 nm	5.6-6 kb
+Narnaviridae	No virion, non-encapsulated	-	E	L	-	2.3-2.9 kb
+Arteriviridae	spherical, env, isometric core	-	E	L	39-54 nm	12.7- 15.7 kb
+Coronaviridae	spherical env	-	E	L	120-160 nm	26.4-31.7 kb
+Mesoniviridae	spherical env	-	E	L	60-80 nm	20 kb
+Roniviridae	Bacilliform env	Nucleocapsid helical	E	L	45x150-200 nm	26 kb
+Nodaviridae	non-env	Icosahedral T=3	E	L	30 nm	1.4-3.1 kb
+Ourmiavirus*	unenv bacilliform	hemi-icosahedra	E	L	32-62 nm	0.9-2.8 kb
+Dicistroviridae	Spherical, non-env	Icosahedral T=pseudo3	E	L	~30 nm	~8.5-10 kb
+Iflaviridae	Spherical, non-env	Icosahedral T=3	E	L	~30 nm	8.8-9.7 kb
+Marnaviridae	Non-env	Polyhedral	E	L	25 nm	8.6 kb
+Picornaviridae	Spherical, non-env	Icosahedral T=pseudo3	E	L	22-30 nm	7-8.8 kb
+Secoviridae	Non-env	Icosahedral T=1, pseudo T=3	E	L	25-30 nm	4-8 kb
+Polemovirus*	Hexagonal, Non-env	Icosahedral T=3	E	L	34 nm	~4.6 kb
+Potyviridae	flexuous filaments, non-env	Helical	E	L	11-15 nm 200-900 nm	9.3-10.8 kb
+Sobemoviridae*	non-env	Icosahedral T=3	E	L	25-30 nm	~4-4.5 kb
+Alphatetraviridae	non-env	Icosahedral T=4	E	L	~40 nm	~6.5 kb
+Carmotetraviridae	non-env	Icosahedral T=4	E	L	~40 nm	~6.1 kb
+Permutotetraviridae	non-env	Icosahedral T=4	E	L	~40 nm	5.6 kb
+Togaviridae	Spherical, env	Icosahedral T=4	E	L	~70 nm	9.7-11.8 kb
+Tombusviridae	Spherical, non-env	Icosahedral T=3	E	L	30-35 nm	1.4-4.8 kb
+Alphaflexiviridae	flexuous filaments, non-env	Helical	E	L	10-15x470-800 nm	5.9-9 kb
+Betaflexiviridae	flexuous filaments, non-env	Helical	E	L	10-15x600-1000 nm	5.9-9 kb
+Gammaflexiviridae	flexuous filaments, non-env	-	E	L	13x720 nm	6.8 kb
+Tymoviridae	Isometric, non-env	Icosahedral T=3	E	L	~30 nm	6-7.5 kb
+Umbravirus*	Non-env	Icosahedral T=3	E	L	52 nm	4-4.2 kb
+Virgaviridae	Rod shaped, helical, non-env	Helical	E	L	2.3-2.5x20 nm	-
-Arenaviridae	spherical to pleomorphic, env	-	E	L	50-300 nm	~3.5-7.5 kb
-Bunyaviridae	spherical to pleomorphic, env	-	E	L	80-120 nm	11.9 kb
-Deltavirus*	Spherical, env	-	E	C	~36-43 nm	1.7 kb
-Emaravirus*	Spherical, env	-	E	L	80-100 nm	12.2 kb
-Bornaviridae	Spherical, env	-	E	L	90±10 nm	~8.9 kb
-Filoviridae	Bacilliform	-	E	L	80x790-970 nm	18.9-19.1 kb
-Paramyxoviridae	Most are spherical, env	-	E	L	~150 nm	~15 kb
-Rhabdoviridae	Most are bullet shaped, env	-	E	L	45-100x100-430 nm	11-15 kb
-Ophioviridae	Naked filamentous nucleocapsids	-	E	L	~3x760 nm	11.3-12.5 kb
-Orthomyxoviridae	spherical or pleomorphic, env	-	E	L	80-120 nm	10.0-14.6 kb
-Tenuivirus*	thin filamentous, non-env	-	E	L	3-10 nm	~16 kb
-Varicosavirus**	Fragile rods, non-env	-	E	-	320-360x~18 nm	12.9 kb

Table 10.16 **rtDNA and rtRNA viral characteristics.** Viraldata collated from ICTV reports(Lefkowitz et al., 2018).

Family	Virion morphology	Symmetry type	Host	Genome	Virion diameter or WxL	Genome size
dsD_Caulimoviridae	isometric or bacilliform, non-env	-	E	C	30×60-900(iso) 45-50(bac) nm	7.2-9.2 kb
dsD_Hepadnaviridae	Spherical or pleomorphic, env	Icosahedral T=4	E	C	42-50 nm	3.0-3.3 kb
R_+Metaviridae	Ovoid, env	-	E	-	-	4-~10 kb
ssR_+Pseudoviridae	Round/ovoid	some Icosahedral	E	L	-	~5-9 kb
ssR_+Retroviridae	Spherical, env	-	E	L	80-100 nm	7-11 kb

10.7 Appendix 7

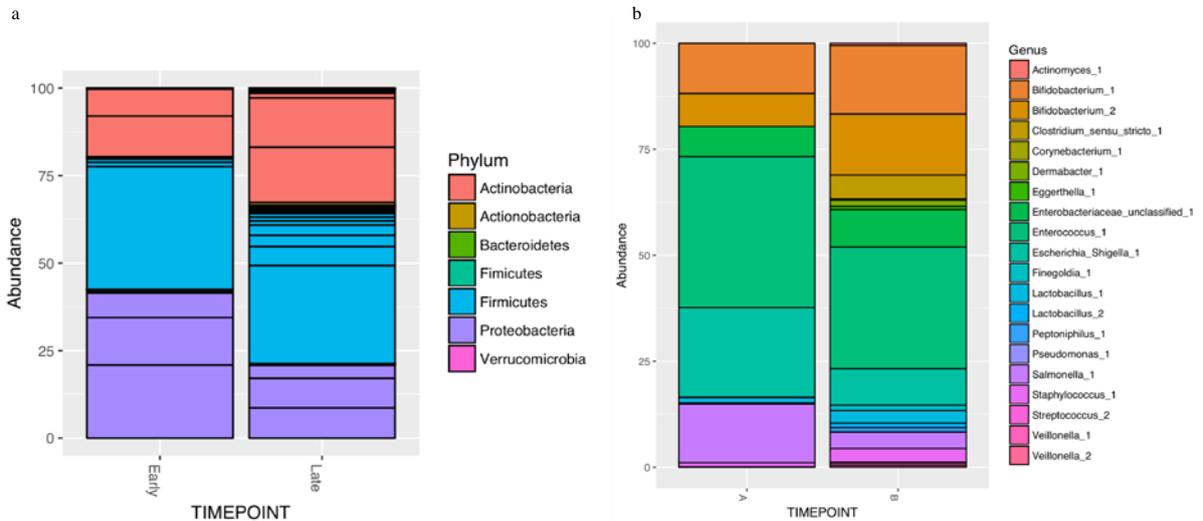


Fig. 4

Figure 10.16 shows relative abundance of phyla (a) and top twenty most abundant genera (b) observed between early and late time points. Counts for each taxonomic unit for all rarefied samples at each time point were summed prior to calculating the relative abundance within the merged sample and plotting as the above bar charts.

10.8 Appendix 8

10.8.1 CRACCD installation GUI and script

```
#!/bin/bash

#if yad isn't installed, install it
YadPresent=$(which yad)
if [ $YadPresent != "/usr/bin/yad" ];then
sudo apt-get install yad
fi

Install=2

ICON=~/.CCRACD_InstallFile/MetabolomicsMenuNew.jpg

yad --title="Metabolomics Program - CCRACD Installation" --center --image=$ICON --image-on-top --size=fit --center --buttons-layout=spread --button="Install":0
--button="gtk-quit:10" --buttons-layout=center

mode="$?"
case $mode in
    0)Install=1 ;;
esac

if [ $Install = "1" ];then
(
echo 1
echo "#Building program directories... 1% Complete"

#Directory Creation
if [ ! -d ~/.CCRACD ];then
    mkdir ~/.CCRACD
fi
sleep 0.4
echo 2
echo "#Building program directories... 2% Complete"
if [ ! -d ~/.CCRACD/R_Scripts ];then
    mkdir ~/.CCRACD/R_Scripts
fi
sleep 0.4
echo 3
echo "#Building program directories... 3% Complete"
if [ ! -d ~/.CCRACD/Scripts ];then
    mkdir ~/.CCRACD/Scripts
fi
sleep 0.4
echo 4
echo "#Building program directories... 4% Complete"
if [ ! -d ~/.CCRACD/tmp ];then
    mkdir ~/.CCRACD/tmp
fi
```

```

sleep 0.4
echo 5
echo "#Building program directories...    5% Complete"
  if [ ! -d ~/CCRACD/MetabolomicProgramOutput ];then
    mkdir ~/CCRACD/MetabolomicProgramOutput
  fi
sleep 0.4
echo 6
echo "#Building program directories...    6% Complete"
  if [ ! -d ~/CCRACD/MetabolomicProgram_Input ];then
    mkdir ~/CCRACD/MetabolomicProgram_Input
  fi
sleep 0.4
echo 7
echo "#Building program directories...    7% Complete"
  if [ ! -d ~/CCRACD/Pictures ];then
    mkdir ~/CCRACD/Pictures
  fi

#Move picture into pic directory
cp ~/CCRACD_InstallFile/MetabolomicsMenuNew.jpg
~/CCRACD/Pictures/MetabolomicsMenuNew.jpg

sleep 0.4
echo 8
echo "#CCRACD Directories Complete    8% Complete"

#Individual Script creation
sleep 0.4
echo 9
echo "#Building program scripts...    9% Complete"
#ProgramMenus
  #Create Program menus script #Remove markers, #Give permissions to the script
  awk '9854_1_START/,9854_1_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/MetabolomicsProgramMenu.sh
  sed 's/9854_1_START//g' ~/CCRACD/MetabolomicsProgramMenu.sh | sed 's/9854_1_END//g' >
~/CCRACD/tmp.sh
  mv ~/CCRACD/tmp.sh ~/CCRACD/MetabolomicsProgramMenu.sh
  chmod 755 ~/CCRACD/MetabolomicsProgramMenu.sh
sleep 0.4
echo 10
echo "#Building program scripts...    10% Complete"

#InputConRepeatStats
  awk '9854_2_START/,9854_2_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/Scripts/InputConditionRepeatStatsPostRunWindow.sh
  sed 's/9854_2_START//g' ~/CCRACD/Scripts/InputConditionRepeatStatsPostRunWindow.sh | sed
's/9854_2_END//g' > ~/CCRACD/tmp.sh
  mv ~/CCRACD/tmp.sh ~/CCRACD/Scripts/InputConditionRepeatStatsPostRunWindow.sh
  chmod 755 ~/CCRACD/Scripts/InputConditionRepeatStatsPostRunWindow.sh

sleep 0.4
echo 11
echo "#Building program scripts...    11% Complete"

```

```

#Metab_InputFileSelect.sh
awk '/9854_3_START/,/9854_3_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/Scripts/Metab_InputFileSelect.sh
sed 's/9854_3_START//g' ~/CCRACD/Scripts/Metab_InputFileSelect.sh | sed 's/9854_3_END//g' >
~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/Scripts/Metab_InputFileSelect.sh
chmod 755 ~/CCRACD/Scripts/Metab_InputFileSelect.sh

sleep 0.4
echo 12
echo "#Building program scripts...    12% Complete"
#MetabolomicsScript.sh
awk '/9854_4_START/,/9854_4_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/Scripts/MetabolomicsScript.sh
sed 's/9854_4_START//g' ~/CCRACD/Scripts/MetabolomicsScript.sh | sed 's/9854_4_END//g' >
~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/Scripts/MetabolomicsScript.sh
chmod 755 ~/CCRACD/Scripts/MetabolomicsScript.sh

sleep 0.4
echo 13
echo "#Building program scripts...    13% Complete"
#MetabSig_In_Raw_Enhanced.sh
awk '/9854_5_START/,/9854_5_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/Scripts/MetabSig_In_Raw_Enhanced.sh
sed 's/9854_5_START//g' ~/CCRACD/Scripts/MetabSig_In_Raw_Enhanced.sh | sed
's/9854_5_END//g' > ~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/Scripts/MetabSig_In_Raw_Enhanced.sh
chmod 755 ~/CCRACD/Scripts/MetabSig_In_Raw_Enhanced.sh

sleep 0.4
echo 14
echo "#Building program scripts...    14% Complete"
#MetabTableMakingScript.sh
awk '/9854_6_START/,/9854_6_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/Scripts/MetabTableMakingScript.sh
sed 's/9854_6_START//g' ~/CCRACD/Scripts/MetabTableMakingScript.sh | sed 's/9854_6_END//g'
> ~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/Scripts/MetabTableMakingScript.sh
chmod 755 ~/CCRACD/Scripts/MetabTableMakingScript.sh

sleep 0.4
echo 15
echo "#Building program scripts...    15% Complete"
#HEATMAP_GILES_METAB.R
awk '/9854_7_START/,/9854_7_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R
sed 's/9854_7_START//g' ~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R | sed
's/9854_7_END//g' > ~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R

sleep 0.4
echo 16
echo "#Building program scripts...    16% Complete"
#PCA_GILES_metab.R

```

```

awk '/9854_8_START/,/9854_8_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/R_Scripts/PCA_GILES_metab.R
sed 's/9854_8_START//g' ~/CCRACD/R_Scripts/PCA_GILES_metab.R | sed 's/9854_8_END//g' >
~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/R_Scripts/PCA_GILES_metab.R

sleep 0.4
echo 17
echo "#Building program scripts...    17% Complete"
#PLSDA_GILES_metab.R
awk '/9854_9_START/,/9854_9_END/' ~/CCRACD_InstallFile/MassScript.sh >
~/CCRACD/R_Scripts/PLSDA_GILES_metab.R
sed 's/9854_9_START//g' ~/CCRACD/R_Scripts/PLSDA_GILES_metab.R | sed
's/9854_9_END//g' > ~/CCRACD/tmp.sh
mv ~/CCRACD/tmp.sh ~/CCRACD/R_Scripts/PLSDA_GILES_metab.R

sleep 0.4
echo 20
echo "#CCRACD scripts Complete    20% Complete"

echo 35
echo "#Moving to terminal. Input your PC login password when prompted and hit enter.. 35%
Complete"
sleep 10

echo 100
echo "#CCRACD Installation Complete    100% Complete"
)|
yad --progress --auto-close --auto-kill --center --width=700 --image=$ICON --image-on-top --
title="Installing CCRACD: |C|ross |C| |R|un |A|analysis for |C|ompound |D|ata profiling" \
--percentage=0

#set up Yad window - read and search for y/n, if found echo y : | tee
~/GGOSS/LogFiles/SPAdesAssembly_LOGFILE.txt | while read -r CheckForMarker; do
# echo 40
echo "#Installing R... Updating system  40% Complete"
sudo apt-get update
# echo 60
echo "#Installing base R...  60% Complete"
sudo apt-get install r-base | tee ~/CCRACD/tmp/Monitor.txt | while read -r CheckForMarker; do
MarkerPresent1=$(echo "$CheckForMarker" | grep -c "[y/n]")
MarkerPresent2=$(echo "$CheckForMarker" | grep -c "[Y/n]")
MarkerPresent3=$(echo "$CheckForMarker" | grep -c "[Y/N]")
if [ "$MarkerPresent1" = "1" ];then
echo "y"
fi
if [ "$MarkerPresent2" = "1" ];then
echo "Y"
fi
if [ "$MarkerPresent3" = "1" ];then
echo "Y"
fi

done
# echo 70

```

```

echo "#Installing required R packages... vegan... 70% Complete"
  sudo apt-get build-dep r-cran-vegan
# echo 75
echo "#Installing required R packages... gplots... 75% Complete"
  sudo apt-get build-dep r-cran-gplots
# echo 80
echo "#Installing required R packages... ggbiplot... 80% Complete"
  sudo apt-get build-dep r-cran-ggbiplot
# echo 85
echo "#Installing required R packages... lattice... 85% Complete"
  sudo apt-get build-dep r-cran-lattice
# echo 90
echo "#Installing required R packages... gclus... 90% Complete"
  sudo apt-get build-dep r-cran-gclus
# echo 95
echo "#Installing required R packages... ggplot2... 95% Complete"
  sudo apt-get build-dep r-cran-ggplot2 & echo "Y"
  sleep 3
  echo "y"

# echo 99
echo "#Required R packages installed          99% Complete"
  sleep 0.8

```

```

yad --title="Metabolomics Program - CCRACD Installation" --width=500 --wrap --size=fit --button="Yes":3 --button="No" --buttons-layout=center \
text="CCRACD Installed:  Open CCRACD upon closing" --text-align=center --center --height=100
created by Giles Holt" --

```

```

mode="$?"
  case $mode in
    3)~/CCRACD/MetabolomicsProgramMenu.sh ;;
  esac

```

```
fi
```

10.8.2 GUI for metabolomics program CRACCD

```
#!/bin/bash

ICON=~/.CCRACD/Pictures/MetabolomicsMenuNew.jpg
FileImport=2
OpenMainMenu=2
RunMenu=2
DataAndPlotMenu=2
OpenFile=2
OpenFile1=2
MetabRunFileSelect_Stage3=2
ByPassToStage3=2
Stage3_MassMetabTrackFromRaw=2
ByPassToStage2=2
Stage2_SelectMetabFix_Menu=2
ByPassToStage1=2

if [ -f ~/.CCRACD/tmp/DataAndPlotMenu.txt ];then
DataAndPlotMenu=1
rm ~/.CCRACD/tmp/DataAndPlotMenu.txt
fi

if [ -f ~/.CCRACD/tmp/RunStage1and2Menu.txt ];then
RunMenu=1
ByPassToStage3=2
ByPassToStage2=2
ByPassToStage2=2
rm ~/.CCRACD/tmp/RunStage1and2Menu.txt
fi

if [ -f ~/.CCRACD/tmp/RunStage3Menu.txt ];then
ByPassToStage3=1
RunMenu=1
rm ~/.CCRACD/tmp/RunStage3Menu.txt
fi

if [ -f ~/.CCRACD/tmp/RunStage2Menu.txt ];then
ByPassToStage2=1
RunMenu=1
rm ~/.CCRACD/tmp/RunStage2Menu.txt
fi

if [[ "$DataAndPlotMenu" = 2 ]] && [[ "$RunMenu" = 2 ]];then
yad --title="Metabolomics PROGRAM -- Main Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --buttons-
layout=spread --button=" Import Data Files ":0 --button="CRACD Run Steps":1 --
button="Plot Data Table":2 --button="Output Files":3 --button="gtk-quit:10" --buttons-layout=center

mode="$?"
case $mode in
0)FileImport=1 ;;
1)RunMenu=1 ;;
2)DataAndPlotMenu=1 ;;
3)OpenFile=1 ;;
```

```

    esac
fi

if [[ "$FileImport" = 1 ]];then

if [ -f ~/CCRACD/tmp/dnd.txt ];then
rm ~/CCRACD/tmp/dnd.txt
rm "/tmp/MetabolomicProgramdnd.log"
rm "/tmp/MetabolomicProgramdnd2.log"
fi

#create fifo file to displaying text in --text-info pane
mkfifo "/tmp/MetabolomicProgramdnd.log"
exec 3<> "/tmp/MetabolomicProgramdnd.log"

#creating the key id for box and plugs

id=$(echo $[(($RANDOM % ($[10000 - 32000] + 1)) + 10000) ])

#the first pane is dnd box
#the second is --text-info from fifo file
yad --plug="$id" --tabnum=1 --dnd | while read line2
do
echo "$line2" >&3
echo "$line2" >> "/tmp/MetabolomicProgramdnd2.log"
done &

yad --plug="$id" --tabnum=2 --text-info --tail <&3>> "/tmp/MetabolomicProgramdnd2.log" &

yad --title="Import files" --center --paned --key="$id" --text="Drag and drop your files into the grey
section of the window to import them" --width="800" --height="500" --splitter="150" --button="gtk-
quit:1" --button="gtk-ok:0"
#out of the script if close buttons are clicked
case $? in
1)
rm "/tmp/MetabolomicProgramdnd.log" "/tmp/MetabolomicProgramdnd2.log"
exit;;
252)
rm "/tmp/MetabolomicProgramdnd.log" "/tmp/MetabolomicProgramdnd2.log"
exit;;
esac

cat /tmp/MetabolomicProgramdnd2.log > ~/CCRACD/tmp/dnd.txt
NumberOfFilesToCopy=$(grep -v -c "ThisIsMyAntiMatch" ~/CCRACD/tmp/dnd.txt)
PercentCompleteWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfFilesToCopy" | bc )

(if [ -f ~/CCRACD/tmp/dnd.txt ];then
line=1
TotallingPercentage=0
for i in $(seq $NumberOfFilesToCopy);do

file=$(awk -v x=$line 'NR==x {print}' ~/CCRACD/tmp/dnd.txt | sed 's/.*\///')
path=$(awk -v x=$line 'NR==x {print}' ~/CCRACD/tmp/dnd.txt | sed 's/file:\///')

```

```

    cp "$path" ~/CCRACD/MetabolomicProgram_Input/"$file"
    echo $TotallingPercentage
    echo "#Importing files    ${TotallingPercentage}% Complete"
    TotallingPercentage=$( echo "$TotallingPercentage + $PercentCompleteWorthOfEachFile" | bc )
    line=$(( $line + 1 ))
done

rm "/tmp/MetabolomicProgramdnd.log" "/tmp/MetabolomicProgramdnd2.log"
if [ -f ~/CCRACD/tmp/dnd.txt ];then
rm ~/CCRACD/tmp/dnd.txt
fi

fi

) | yad --progress --auto-close --auto-kill --center --width=700 --image=$ICON --image-on-top --
title="Importing files into MetabolomicProgram: CRACD" \
--percentage=0

thelist=$(ls ~/CCRACD/MetabolomicProgram_Input/)
thechoice=$(yad --title="Imported Files" --width=800 --height=600 --center --button="Continue":0 --
button="Remove" --list --multiple --column="Files present for analysis" --separator="" $thelist)

mode="$?"
    case $mode in
        0)OpenMainMenu=1 ;;
    esac

fi

if [[ "$RunMenu" = 1 ]] || [[ "$ByPassToStage1" = 1 ]] || [[ "$ByPassToStage2" = 1 ]] || [[
"$ByPassToStage3" = 1 ]];then

    if [[ "$ByPassToStage1" = 2 ]] && [[ "$ByPassToStage2" = 2 ]] && [[ "$ByPassToStage3" = 2
]];then
Stage2_SelectMetabFix_Menu=2
Stage3_MassMetabTrackFromRaw=2
    yad --title="Metabolomics PROGRAM -- Main Menu
Created by Giles Holt" --text="STEP 1:
IDENTIFY COMPOUNDS OF INTEREST
    Identify 'compounds of interest'
    based on user specified factors

STEP 2:
CLEANUP COMPOUNDS OF INTEREST
    Match compounds that are the same,
    and identalise their names

STEP 3:
COMPOUND FLUX THROUGH CONDITIONS
    Find compound matches to the 'compounds
of interest' within mass datasets,

```

```

    profiling transgression" --text-align=left --center --image=$ICON --image-on-top --size=fit --
button="Previous":5 --button="
                Step 1 - Identify
                ":6 --button="
                Step 3 - Profile
                ":4 --button="
2 - Cleanup
layout=start

```

```

mode="$?"
case $mode in
  3)Stage2_SelectMetabFix_Menu=1 ;;
  4)Stage3_MassMetabTrackFromRaw=1 ;;
  5)OpenMainMenu=1 ;;
  6)Stage1_IdentifyCompounds=1 ;;
esac
fi

```

```

if [[ "$Stage1_IdentifyCompounds" = 1 ]] || [[ "$ByPassToStage1" = 1 ]];then
  yad --title="Metabolomics PROGRAM -- Step 1 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button="
Previous
                ":2 --button="Filter settings and input files":1 --button="
                Run
":0 --button-layout=center

```

```

mode="$?"
case $mode in
  0)~/CCRACD/Scripts/MetabolomicsScript.sh ;;
  1)MetabRunFileSelect_Stage1=1 ;;
  2)echo "1" > ~/CCRACD/tmp/RunStage1and2Menu.txt && OpenMainMenu=1 ;;
esac
fi

```

```

if [[ "$MetabRunFileSelect_Stage1" = 1 ]];then

```

```

ls ~/CCRACD/MetabolomicProgram_Input/ > ~/CCRACD/tmp/AllInputDataFiles.txt

```

```

NumberOfChoices=$( grep -c -v "HelloThisIsMyAntiMatch" ~/CCRACD/tmp/AllInputDataFiles.txt )

```

```

Choice1=$(sed -n 1p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice2=$(sed -n 2p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice3=$(sed -n 3p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice4=$(sed -n 4p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice5=$(sed -n 5p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice6=$(sed -n 6p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice7=$(sed -n 7p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice8=$(sed -n 8p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice9=$(sed -n 9p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice10=$(sed -n 10p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice11=$(sed -n 11p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice12=$(sed -n 12p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice13=$(sed -n 13p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice14=$(sed -n 14p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice15=$(sed -n 15p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice16=$(sed -n 16p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice17=$(sed -n 17p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice18=$(sed -n 18p ~/CCRACD/tmp/AllInputDataFiles.txt)

```

```

if [[ $NumberOfChoices = 1 ]];then
mode="$?"
case $mode in
1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2):" \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!Multiple files to select!" '-' 'N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~ /CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

if [[ $NumberOfChoices = 2 ]];then

mode="$?"
case $mode in
1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2):" \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!Multiple files to select!" '-' 'N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~ /CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else
if [[ $NumberOfChoices = 3 ]];then
mode="$?"
case $mode in
1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2):" \
--field="          p value":CB \

```

```

--field="                                Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!Multiple files to select!" '-' "'N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

        else
            if [[ $NumberOfChoices = 4 ]];then
mode="$?"
        case $mode in
            1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
        esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                                Select file(s):":CB \
--field="                                mz range (e.g. 121.213 1032.142):": \
--field="                                Retention time range (e.g. 1.6 19.2):" \
--field="                                p value:":CB \
--field="                                Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!Multiple files to select!" '-' "'N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

        else
            if [[ $NumberOfChoices = 5 ]];then
mode="$?"
        case $mode in
            1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
        esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                                Select file(s):":CB \
--field="                                mz range (e.g. 121.213 1032.142):": \
--field="                                Retention time range (e.g. 1.6 19.2):" \
--field="                                p value:":CB \
--field="                                Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!Multiple files to select!" '-' "'N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

        else
            if [[ $NumberOfChoices = 6 ]];then
mode="$?"
        case $mode in

```

```

1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!Multiple files to select!"
'- '- "N/A!0.05!0.01!0.005!0.001" '- 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else
    if [[ $NumberOfChoices = 7 ]];then
        mode="$?"
        case $mode in
            1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
            esac | \
            yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!Multiple
files to select!" '- '- "N/A!0.05!0.01!0.005!0.001" '- 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

        else
            if [[ $NumberOfChoices = 8 ]];then
                mode="$?"
                case $mode in
                    1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
                    esac | \
                    yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8
}!Multiple files to select!" '- '- "N/A!0.05!0.01!0.005!0.001" '- 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else
    if [[ $NumberOfChoices = 9 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!Multiple files to select!" '-' '-' "N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else
    if [[ $NumberOfChoices = 10 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!Multiple files to select!" '-' '-' "N/A!0.05!0.01!0.005!0.001" '-'
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else
    if [[ $NumberOfChoices = 11 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;

```

```

    esac | \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!Multiple files to select!" '-' 'N/A!0.05!0.01!0.005!0.001"
-' 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt
else

```

```

        if [[ $NumberOfChoices = 12 ]];then

```

```

mode="$?"
case $mode in
1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!Multiple files to select!" '-' 'N/A!0.05!0.01!0.005!0.001"
-' 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt
else

```

```

        if [[ $NumberOfChoices = 13 ]];then

```

```

mode="$?"
case $mode in
1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

```

```

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8
}!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!Multiple files to select!" '-' '-'
"N/A!0.05!0.01!0.005!0.001" '-' 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

    if [[ $NumberOfChoices = 14 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
    Select file(s):" :CB \
--field="
    mz range (e.g. 121.213 1032.142):" : \
--field="
    Retention time range (e.g. 1.6 19.2):" \
--field="
    p value:" :CB \
--field="
    Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8
}!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!Multiple files to
select!" '-' '-' "N/A!0.05!0.01!0.005!0.001" '-' 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

    if [[ $NumberOfChoices = 15 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac \
yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
    Select file(s):" :CB \
--field="
    mz range (e.g. 121.213 1032.142):" : \
--field="
    Retention time range (e.g. 1.6 19.2):" \
--field="
    p value:" :CB \
--field="
    Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8
}!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!M
ultiple files to select!" '-' '-' "N/A!0.05!0.01!0.005!0.001" '-' 'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

```

```

        if [[ $NumberOfChoices = 16 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\
  yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!${
Choice16}!Multiple files to select!" '- '- "N/A!0.05!0.01!0.005!0.001" '-
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

        if [[ $NumberOfChoices = 17 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\
  yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Select file(s)":CB \
--field="          mz range (e.g. 121.213 1032.142)": \
--field="          Retention time range (e.g. 1.6 19.2)": \
--field="          p value":CB \
--field="          Max CV%:" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!${
Choice16}!${Choice17}!Multiple files to select!" '- '- "N/A!0.05!0.01!0.005!0.001" '-
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

else

        if [[ $NumberOfChoices = 18 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage1Menu.txt && OpenMainMenu=1 ;;
esac |\

```

```

yad --title="Metabolomics PROGRAM -- Step 1 Settings Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                Select file(s)":CB \
--field="                mz range (e.g. 121.213 1032.142)": \
--field="                Retention time range (e.g. 1.6 19.2)": \
--field="                p value":CB \
--field="                Max CV%": \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!${
Choice16}!${Choice17}!${Choice18}!Multiple files to select!" '- ' "N/A!0.05!0.01!0.005!0.001" '-
'Stage1_FilterCompounds.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt

```

else

```

echo "There is an unexpected high number of files in the input, only 18 are shown"
notify-send "There is an unexpected high number of files in the input, only 18 are
shown"

```

```

fi

```

```

echo "1" > ~/.CCRACD/tmp/RunStage1Menu.txt

```

```

OpenMainMenu=1

```

```

fi

```

```

if [[ "$Stage2_SelectMetabFix_Menu" = 1 ]] || [[ "$ByPassToStage2" = 1 ]];then
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button="

```

```
Previous          ":2 --button="Settings and input files":1 --button="          Run
":0 --buttons-layout=center
```

```
mode="$?"
case $mode in
  0)~/CCRACD/Scripts/MetabolomicsScript.sh ;;
  1)MetabRunFileSelect_Stage2=1 ;;
  2)echo "1" > ~/CCRACD/tmp/RunStage1and2Menu.txt && OpenMainMenu=1 ;;
esac
fi
```

```
if [[ "$MetabRunFileSelect_Stage2" = 1 ]];then
```

```
ls ~/CCRACD/MetabolomicProgram_Input/ > ~/CCRACD/tmp/AllInputDataFiles.txt
```

```
NumberOfChoices=$( grep -c -v "HelloThisIsMyAntiMatch" ~/CCRACD/tmp/AllInputDataFiles.txt )
```

```
Choice1=$(sed -n 1p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice2=$(sed -n 2p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice3=$(sed -n 3p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice4=$(sed -n 4p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice5=$(sed -n 5p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice6=$(sed -n 6p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice7=$(sed -n 7p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice8=$(sed -n 8p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice9=$(sed -n 9p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice10=$(sed -n 10p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice11=$(sed -n 11p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice12=$(sed -n 12p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice13=$(sed -n 13p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice14=$(sed -n 14p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice15=$(sed -n 15p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice16=$(sed -n 16p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice17=$(sed -n 17p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice18=$(sed -n 18p ~/CCRACD/tmp/AllInputDataFiles.txt)
```

```
if [[ $NumberOfChoices = 1 ]];then
```

```
mode="$?"
```

```
case $mode in
```

```
  1)echo "1" > ~/CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
```

```
esac | \
```

```
yad --title="Metabolomics PROGRAM -- Step 2 Menu
```

```
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
```

```
--field="          Chosen compounds file (+ion):":CB \
```

```
--field="          Chosen compounds file (-ion):":CB \
```

```
--field="          mz minimum decimal place match":CB \
```

```
--field="Retention time +/- minute/s (Example:0.5):" \
```

```
--field="Experiment name (no spaces, if spaces are needed use _):" \
```

```
"${Choice1}!-" "${Choice1}!-" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
```

```
--text-info --show-uri --width=600 --height=500 --center --wrap \
```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

else

```
if [[ $NumberOfChoices = 2 ]];then
```

```

mode="$?"
case $mode in
  1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
  esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice1}!${Choice2}!-" "${Choice1}!${Choice2}!-" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

else

```
if [[ $NumberOfChoices = 3 ]];then
```

```

mode="$?"
case $mode in
  1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
  esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice3}!${Choice2}!${Choice1}!-" "${Choice3}!${Choice2}!${Choice1}!-"
"1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

else

```
if [[ $NumberOfChoices = 4 ]];then
```

```

mode="$?"
case $mode in
  1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
  esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 5 ]];then

```

```

mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac \

```

```

yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 6 ]];then

```

```

mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac \

```

```

yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 7 ]];then
        mode="$?"
        case $mode in
            1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
        esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \
"${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 8 ]];then
        mode="$?"
        case $mode in
            1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
        esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}
}!-!"
"${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}
}!-!" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 9 ]];then
        mode="$?"
        case $mode in
            1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
        esac | \

```

```

    yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}
!${Choice1}!-!"
"${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}
!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

    else

                if [[ $NumberOfChoices = 10 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
    yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice
3}!${Choice2}!${Choice1}!-!"
"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice
3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

    else

                if [[ $NumberOfChoices = 11 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
    yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

```

```

"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

else

```

      if [[ $NumberOfChoices = 12 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
  esac | \
  yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

else

```

      if [[ $NumberOfChoices = 13 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
  esac | \
  yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"

```

```

ice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

    else

        if [[ $NumberOfChoices = 14 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

    else

        if [[ $NumberOfChoices = 15 ]];then
mode="$?"
case $mode in
    1)echo "1" > ~/.CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

```

```

else

        if [[ $NumberOfChoices = 16 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
  yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${
Choice1}!-"
"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${
Choice1}!-" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~ /CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

else

        if [[ $NumberOfChoices = 17 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac | \
  yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!$
{Choice2}!${Choice1}!-"
"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!$
{Choice2}!${Choice1}!-" "1!2!3!4!5!6!7!8!" '1' 'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~ /CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

else

```

```

        if [[ $NumberOfChoices = 18 ]];then
mode="$?"
case $mode in
  1)echo "1" > ~/CCRACD/tmp/RunStage2Menu.txt && OpenMainMenu=1 ;;
esac |\

  yad --title="Metabolomics PROGRAM -- Step 2 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Chosen compounds file (+ion):":CB \
--field="          Chosen compounds file (-ion):":CB \
--field="          mz minimum decimal place match:":CB \
--field="Retention time +/- minute/s (Example:0.5):" \
--field="Experiment name (no spaces, if spaces are needed use _):" \

"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${
Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!
${Choice3}!${Choice2}!${Choice1}!-"
"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${
Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!
${Choice3}!${Choice2}!${Choice1}!-" "1!2!3!4!5!6!7!8!" '1'
'Stage2_CleanupCompoundsOfInterest.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt >
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt

else

  echo "There is an unexpected high number of files in the input, only 18 are shown"
  notify-send "There is an unexpected high number of files in the input, only 18 are
shown"

fi

```

```

echo "1" > ~/CCRACD/tmp/RunStage2Menu.txt

OpenMainMenu=1

fi

if [[ "$Stage3_MassMetabTrackFromRaw" = 1 ]] || [[ "$ByPassToStage3" = 1 ]];then
  yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button="
Previous          ":2 --button="Settings and File assignment":1 --button="          Run
":0 --buttons-layout=center

  mode="$?"
  case $mode in
    0)~/CCRACD/Scripts/MetabSig_In_Raw_Enhanced.sh ;;
    1)MetabRunFileSelect_Stage3=1 ;;
    2)echo "1" > ~/CCRACD/tmp/RunStage1and2Menu.txt && OpenMainMenu=1 ;;
  esac
fi

if [[ "$MetabRunFileSelect_Stage3" = 1 ]];then

ls ~/CCRACD/MetabolomicProgram_Input/ > ~/CCRACD/tmp/AllInputDataFiles.txt

NumberOfChoices=$( grep -c -v "HelloThisIsMyAntiMatch" ~/CCRACD/tmp/AllInputDataFiles.txt )

Choice1=$(sed -n 1p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice2=$(sed -n 2p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice3=$(sed -n 3p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice4=$(sed -n 4p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice5=$(sed -n 5p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice6=$(sed -n 6p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice7=$(sed -n 7p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice8=$(sed -n 8p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice9=$(sed -n 9p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice10=$(sed -n 10p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice11=$(sed -n 11p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice12=$(sed -n 12p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice13=$(sed -n 13p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice14=$(sed -n 14p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice15=$(sed -n 15p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice16=$(sed -n 16p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice17=$(sed -n 17p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice18=$(sed -n 18p ~/CCRACD/tmp/AllInputDataFiles.txt)

if [[ $NumberOfChoices = 1 ]];then
mode="$?"
  case $mode in
    1)echo "1" > ~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
  esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!-!" "${Choice1}!-!" "${Choice1}!-!" "${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw
abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
if [[ $NumberOfChoices = 2 ]];then
mode="$?"
case $mode in
1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!${Choice2}!-!" "${Choice1}!${Choice2}!-!" "${Choice1}!${Choice2}!-!"
"${Choice1}!${Choice2}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
if [[ $NumberOfChoices = 3 ]];then
mode="$?"
case $mode in
1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \

```

```

--field="                Total raw positive file":CB \
--field="                mz minimum decimal place match":CB \
--field="                Retention time +/- minute/s (Example:0.5):" \
--field="                Abundance":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice3}!${Choice2}!${Choice1}!-!" "${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice1}!${Choice2}!${Choice3}!-!" "${Choice2}!${Choice1}!${Choice3}!-!"
"1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 4 ]];then
mode="$?"
    case $mode in
        1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="                Total raw negative file":CB \
--field="                Total raw positive file":CB \
--field="                mz minimum decimal place match":CB \
--field="                Retention time +/- minute/s (Example:0.5):" \
--field="                Abundance":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!-!"
"${Choice2}!${Choice1}!${Choice4}!${Choice3}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw
abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 5 ]];then
mode="$?"
    case $mode in
        1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="                Total raw negative file":CB \
--field="                Total raw positive file":CB \
--field="                mz minimum decimal place match":CB \

```

```

--field="          Retention time +/- minute/s (Example:0.5):" \
--field="          Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!-!"
"${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice5}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw
abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 6 ]];then
mode="$?"
    case $mode in
        1)~/.CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="          Total raw negative file":CB \
--field="          Total raw positive file":CB \
--field="          mz minimum decimal place match:":CB \
--field="          Retention time +/- minute/s (Example:0.5):" \
--field="          Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!-!"
"${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${Choice5}!-!" "1!2!3!4!5!6!7!8!"
'1' "Raw abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

```

else
    if [[ $NumberOfChoices = 7 ]];then
mode="$?"
    case $mode in
        1)~/.CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="          Total raw negative file":CB \
--field="          Total raw positive file":CB \
--field="          mz minimum decimal place match:":CB \
--field="          Retention time +/- minute/s (Example:0.5):" \

```

```

--field="                                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!-!"
"${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${Choice5}!-!"
"1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

else

    if [[ $NumberOfChoices = 8 ]];then
        mode="$?"
    case $mode in
        1)~/.CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="                                Total raw negative file":CB \
--field="                                Total raw positive file":CB \
--field="                                m/z minimum decimal place match:":CB \
--field="                                Retention time +/- minute/s (Example:0.5):" \
--field="                                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}
!-!"
"${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}
!-!"
"${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}
!-!"
"${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${Choice5}
!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

else

    if [[ $NumberOfChoices = 9 ]];then
        mode="$?"
    case $mode in
        1)~/.CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac | \

```

```

                yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}
!${Choice1}!-!"
"${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}
!${Choice1}!-!"
"${Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}
!${Choice6}!-!"
"${Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}
!${Choice5}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

        if [[ $NumberOfChoices = 10 ]];then
            mode="$?"
        case $mode in
            1)~/.CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
        esac | \

```

```

                yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice
3}!${Choice2}!${Choice1}!-!"
"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice
3}!${Choice2}!${Choice1}!-!"
"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice
4}!${Choice5}!${Choice6}!-!"
"${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice
3}!${Choice6}!${Choice5}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

if [[ $NumberOfChoices = 11 ]];then
    mode="$?"
    case $mode in
        1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac |\
        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice
e4}!${Choice3}!${Choice2}!${Choice1}!-"
"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice
e4}!${Choice3}!${Choice2}!${Choice1}!-"
"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice
e3}!${Choice4}!${Choice5}!${Choice6}!-"
"${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice
e4}!${Choice3}!${Choice6}!${Choice5}!-" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised
abundance!-" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

if [[ $NumberOfChoices = 12 ]];then
    mode="$?"
    case $mode in
        1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
    esac |\
        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance":CB \

```

```

--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

if [[ $NumberOfChoices = 13 ]];then
mode="$?"
case $mode in
  1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
  esac | \
      yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="          Total raw negative file":CB \
--field="          Total raw positive file":CB \
--field="          m/z minimum decimal place match":CB \
--field="          Retention time +/- minute/s (Example:0.5):" \
--field="          Abundance":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

```

```

"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

if [[ $NumberOfChoices = 14 ]];then
mode="$?"

```

```

case $mode in
  1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
esac |\
        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!-!"
"${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${Choice8}!${Ch
oice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${Choice5}!-!"
"1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
        --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

else

if [[ $NumberOfChoices = 15 ]];then
mode="$?"
case $mode in
  1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
esac |\
        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!-!"

```

```
"${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${Choice9}!${C
hoice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${Choice5}!-!"
"1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt
```

else

```
if [[ $NumberOfChoices = 16 ]];then
mode="$?"
case $mode in
1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
esac |\
yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
Total raw negative file":CB \
--field="
Total raw positive file":CB \
--field="
mz minimum decimal place match":CB \
--field="
Retention time +/- minute/s (Example:0.5)": \
--field="
Abundance":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
```

```
"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${
Choice1}!-!"
"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!${Choice2}!${
Choice1}!-!"
"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${
Choice6}!-!"
"${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${Choice10}!${
Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!${Choice6}!${
Choice5}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt
```

else

```
if [[ $NumberOfChoices = 17 ]];then
mode="$?"
case $mode in
1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
```

```

    esac | \
        yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!$
{Choice2}!${Choice1}!-!"
"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!${Choice3}!$
{Choice2}!${Choice1}!-!"
"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!${Choice4}!$
{Choice5}!${Choice6}!-!"
"${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${Choice11}!${
Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!${Choice3}!$
{Choice6}!${Choice5}!-!" "1!2!3!4!5!6!7!8!" "1" "Raw abundance!Normalised abundance!-!"
'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

    if [[ $NumberOfChoices = 18 ]];then
        mode="$?"
        case $mode in
            1)~/CCRACD/tmp/RunStage3Menu.txt && OpenMainMenu=1 ;;
        esac | \
            yad --title="Metabolomics PROGRAM -- Step 3 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Chosen compounds file, +ion (Post step 2 recommended)":CB \
--field="Chosen compounds file, -ion (Post step 2 recommended)":CB \
--field="
                Total raw negative file":CB \
--field="
                Total raw positive file":CB \
--field="
                mz minimum decimal place match:":CB \
--field="
                Retention time +/- minute/s (Example:0.5):" \
--field="
                Abundance:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${
Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!$
{Choice3}!${Choice2}!${Choice1}!-!"
"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${

```

```

Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice6}!${Choice5}!${Choice4}!
${Choice3}!${Choice2}!${Choice1}!-!"
"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${
Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice1}!${Choice2}!${Choice3}!
${Choice4}!${Choice5}!${Choice6}!-!"
"${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!${
Choice11}!${Choice10}!${Choice9}!${Choice8}!${Choice7}!${Choice2}!${Choice1}!${Choice4}!
${Choice3}!${Choice6}!${Choice5}!-!" "1!2!3!4!5!6!7!8!" '1' "Raw abundance!Normalised
abundance!-!" 'MetabData_CrossConditionTable.txt' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt >
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt

```

else

```

echo "There is an unexpected high number of files in the input, only 18 are shown"
notify-send "There is an unexpected high number of files in the input, only 18 are

```

shown"

```

fi

```

```

echo "1" > ~/.CCRACD/tmp/RunStage3Menu.txt

```

```

OpenMainMenu=1

```

```

fi

```

fi

```

if [[ "$DataAndPlotMenu" = 1 ]];then

```

```

SelectionOfR_Script=2

```

```

MetabPlotAndFileSelect=2

```

```

yad --title="Metabolomics PROGRAM -- Step 4 Menu

```

```

Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button="

```

```
Previous          ":3 --button="Data and Plot selection":1 --button="          Run
":0 --buttons-layout=center
```

```
mode="$?"
case $mode in
  0)SelectionOfR_Script=1 ;;
  1)MetabPlotAndFileSelect=1 ;;
  3)OpenMainMenu=1 ;;
esac

if [[ "$SelectionOfR_Script" = 1 ]];then
  InputTable=$(awk -F "|" '{print $1}' ~/CCRACD/tmp/MetabDataAndPlotSelected.txt)
  PlotType=$(awk -F "|" '{print $2}' ~/CCRACD/tmp/MetabDataAndPlotSelected.txt)
  #number of variable in the input table
  NumberOfCompounds=$(grep -c -v "ThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgram_Input/"$InputTable")
  NumberOfCompounds=$(( $NumberOfCompounds - 1 ))
  NumberOfSamples=$(awk -F $'\t' 'BEGIN { OFS = FS }; NR==1 {print}'
~/CCRACD/MetabolomicProgram_Input/"$InputTable" | cut -f 2- | awk -F $'\t' 'BEGIN { OFS = FS }
; { print NF } ')
  if [ "$PlotType" == "Heat Map" ];then
    echo "          Heat Map selected
```

Creating Heat Map now using \$InputTable table file

Number of Compounds (including their multiplication from conditions) to account for: \$NumberOfCompounds

```
Number of Samples to account for: $NumberOfSamples"
#change the variable numbers in R script to the variables created from table
#Sample number
sed -i -e "s/rowsep=(1:9)\+/rowsep=(1:$NumberOfSamples)/g"
~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R
#Compound Number
sed -i -e "s/colsep=(1:48)\+/colsep=(1:${NumberOfCompounds})/g"
~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R
#make a copy of the table.csv file and change to .txt
cp ~/CCRACD/MetabolomicProgram_Input/"$InputTable"
~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt
#change the input file and variable number back to default
#Sample number
sed -i -e "s/rowsep=(1:$NumberOfSamples)\+/rowsep=(1:9)/g"
~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R
#Compound Number
sed -i -e "s/colsep=(1:${NumberOfCompounds})\+/colsep=(1:48)/g"
~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R
#Run the heatmap script
yad --title="CCRACD-- Heat Map Generation          Created by Giles Holt" --
width=400 --center --sticky --on-top --no-buttons --no-escape --text-align=center --text="
Generating Heat Map

" & R < ~/CCRACD/R_Scripts/HEATMAP_GILES_METAB.R --no-save

#closes the window
pkill yad
```

```

        #convert the pdf to the plot type output selected
        convert ~/CCRACD/MetabolomicProgramOutput/HeatMap.pdf
~/CCRACD/MetabolomicProgramOutput/HeatMap.jpeg
        convert ~/CCRACD/MetabolomicProgramOutput/Heatmaps_dendo.test.pdf
~/CCRACD/MetabolomicProgramOutput/Heatmaps_dendo.test.jpeg

        gnome-open ~/CCRACD/MetabolomicProgramOutput/HeatMap.pdf
        #delete the copy .txt file
        rm ~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt
    fi

    if [ "$PlotType" == "PCA" ];then
        #change the variable numbers in R script to the variables created from table
        #make a copy of the table.csv file and change to .txt
        cp ~/CCRACD/MetabolomicProgram_Input/"$InputTable"
~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt
        #Run the PCA script
        yad --title="CCRACD -- PCA Generation          Created by Giles Holt" --width=400 -
-center --sticky --on-top --no-buttons --no-escape --text-align=center --text="          Generating
PCA

" & R < ~/CCRACD/R_Scripts/PCA_GILES_metab.R --no-save

        #Closes window
        pkill yad

        #make jpegs as well as pdf's
        convert -density 400 ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.all.scre.pdf -
quality 150 ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.all.scre.jpeg
        convert ~/CCRACD/MetabolomicProgramOutput/Correlations.pdf
~/CCRACD/MetabolomicProgramOutput/Correlations.jpeg
        convert -density 400 ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.Dotplot.pdf -
quality 150 ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.Dotplot.jpg
        convert ~/CCRACD/MetabolomicProgramOutput/Metab.Biplot.pdf
~/CCRACD/MetabolomicProgramOutput/Metab.Biplot.jpeg
        convert ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.pdf
~/CCRACD/MetabolomicProgramOutput/Metab.PCA.jpeg

        gnome-open ~/CCRACD/MetabolomicProgramOutput/Metab.PCA.pdf
        #change the input file and variable number back to default
        #delete the copy .txt file
        rm ~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt
    fi
    echo "
                $PlotType for $InputTable is complete
"

    OpenMainMenu=1

    if [ "$PlotType" == "PLSDA" ];then
        #make a copy of the table.csv file and change to .txt
        cp ~/CCRACD/MetabolomicProgram_Input/"$InputTable"
~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt
        #Run the PCA script

```

```

yad --title="CCRACD -- PLS-DA Generation          Created by Giles Holt" --
width=400 --center --sticky --on-top --no-buttons --no-escape --text-align=center --text="
Generating PLS-DA

```

```

" & R < ~/CCRACD/R_Scripts/PLSDA_GILES_metab.R --no-save

```

```

#Closes window
pkill yad

```

```

#make jpegs as well as pdf's
convert ~/CCRACD/MetabolomicProgramOutput/Metab.PLSDA.pdf
~/CCRACD/MetabolomicProgramOutput/Metab.PLSDA.jpeg

```

```

gnome-open ~/CCRACD/MetabolomicProgramOutput/Metab.PLSDA.pdf
#change the input file and variable number back to default
#delete the copy .txt file
rm ~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt

```

```

fi
fi

```

```

if [[ "$MetabPlotAndFileSelect" = 1 ]];then
ls ~/CCRACD/MetabolomicProgram_Input/ > ~/CCRACD/tmp/AllInputDataFiles.txt
NumberOfChoices=$( grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/tmp/AllInputDataFiles.txt )
Choice1=$(sed -n 1p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice2=$(sed -n 2p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice3=$(sed -n 3p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice4=$(sed -n 4p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice5=$(sed -n 5p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice6=$(sed -n 6p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice7=$(sed -n 7p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice8=$(sed -n 8p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice9=$(sed -n 9p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice10=$(sed -n 10p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice11=$(sed -n 11p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice12=$(sed -n 12p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice13=$(sed -n 13p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice14=$(sed -n 14p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice15=$(sed -n 15p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice16=$(sed -n 16p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice17=$(sed -n 17p ~/CCRACD/tmp/AllInputDataFiles.txt)
Choice18=$(sed -n 18p ~/CCRACD/tmp/AllInputDataFiles.txt)

if [[ $NumberOfChoices = 1 ]];then
mode="$?"
case $mode in
1) echo "1" > ~/CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
esac | \
yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Choose table file:":CB \
--field="          Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!-" "Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \

```

```

--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt

  else
    if [[ $NumberOfChoices = 2 ]];then
      mode="$?"
      case $mode in
        1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
        esac | \
          yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
          Choose table file:":CB \
--field="
          Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!${Choice2}!-!" "Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt

    else
      if [[ $NumberOfChoices = 3 ]];then
        mode="$?"
        case $mode in
          1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
          esac | \
            yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
            Choose table file:":CB \
--field="
            Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!${Choice2}!${Choice3}!-!" "Heat Map!PCA!PLSDA"
'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt

        else
          if [[ $NumberOfChoices = 4 ]];then
            mode="$?"
            case $mode in
              1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1
;;
              esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!-!" "Heat Map!PCA!PLSDA"
'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/CCRACD/tmp/MetabDataAndPlotSelected.txt

```

```

else

```

```

if [[ $NumberOfChoices = 5 ]];then
mode="$?"

```

```

case $mode in

```

```

1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&

```

```

OpenMainMenu=1 ;;

```

```

esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 4 Menu

```

```

Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \

```

```

--field="
Choose table file:":CB \

```

```

--field="
Choose plot/graph type:":CB \

```

```

--field="Experiment name (no spaces, if spaces are needed please use _):" \

```

```

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!-" "Heat Map!PCA!PLSDA"

```

```

'MetabData_CrossConditionGraph.jpeg' \

```

```

--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/CCRACD/tmp/MetabDataAndPlotSelected.txt

```

```

else

```

```

if [[ $NumberOfChoices = 6 ]];then
mode="$?"

```

```

case $mode in

```

```

1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&

```

```

OpenMainMenu=1 ;;

```

```

esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 4 Menu

```

```

Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \

```

```

--field="
Choose table file:":CB \

```

```

--field="
Choose plot/graph type:":CB \

```

```

--field="Experiment name (no spaces, if spaces are needed please use _):" \

```

```

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!-" "Heat

```

```

Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \

```

```

--text-info --show-uri --width=600 --height=500 --center --wrap \

```

```

--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/CCRACD/tmp/MetabDataAndPlotSelected.txt

```

```

else

```

```

if [[ $NumberOfChoices = 7 ]];then
mode="$?"

```

```

case $mode in

```

```

1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&

```

```

OpenMainMenu=1 ;;

```

```

esac | \

```

```

yad --title="Metabolomics PROGRAM -- Step 4 Menu

```

```

Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \

```

```

--field="
Choose table file:":CB \

```

```

--field="
Choose plot/graph type:":CB \

```

```

--field="Experiment name (no spaces, if spaces are needed please use _):" \

```

```

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!-" "Heat
Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 8 ]];then
            mode="$?"
            case $mode in
                1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&
OpenMainMenu=1 ;;
                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
}!-" "Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 9 ]];then
            mode="$?"
            case $mode in
                1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&
OpenMainMenu=1 ;;
                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
}!${Choice9}!-" "Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 10 ]];then
            mode="$?"
            case $mode in
                1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt &&
OpenMainMenu=1 ;;
                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \

```

```

--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!-!" "Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 11 ]];then
            mode="$?"
            case $mode in
                1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt

&& OpenMainMenu=1 ;;

                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!-!" "Heat Map!PCA!PLSDA"
'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 12 ]];then
            mode="$?"
            case $mode in
                1)echo "1" > ~/.CCRACD/tmp/DataAndPlotMenu.txt

&& OpenMainMenu=1 ;;

                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="
                Choose table file:":CB \
--field="
                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!-!" "Heat Map!PCA!PLSDA"
'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
    else
        if [[ $NumberOfChoices = 13 ]];then
            mode="$?"
            case $mode in
                1)echo "1" >
~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
                esac | \

```

```

                                yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                                Choose table file:":CB \
--field="                                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!-!" "Heat Map!PCA!PLSDA"
'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
                                else
                                    if [[ $NumberOfChoices = 14 ]];then
                                        mode="$?"
                                        case $mode in
                                            1)echo "1" >
~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
                                        esac |\
                                        yad --title="Metabolomics PROGRAM -- Step 4 Menu
Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                                Choose table file:":CB \
--field="                                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!-!" "Heat
Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
                                else
                                    if [[ $NumberOfChoices = 15 ]];then
                                        mode="$?"
                                        case $mode in
                                            1)echo "1" >
~/.CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
                                        esac |\
                                        yad --title="Metabolomics PROGRAM -- Step 4
Menu
Created by Giles
Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                                Choose table file:":CB \
--field="                                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!-!"
"Heat Map!PCA!PLSDA" 'MetabData_CrossConditionGraph.jpeg' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/.CCRACD/tmp/MetabDataAndPlotSelected.txt
                                else

```

```

        if [[ $NumberOfChoices = 16 ]];then
            mode="$?"
            case $mode in
                1)echo "1" >
~/CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
            esac | \
            yad --title="Metabolomics PROGRAM -- Step 4
Menu                                     Created by Giles
Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                Choose table file:":CB \
--field="                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!${
Choice16}!-!" "Heat Map!PCA!PLSDA" "MetabData_CrossConditionGraph.jpeg" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/CCRACD/tmp/MetabDataAndPlotSelected.txt
        else
            if [[ $NumberOfChoices = 17 ]];then
                mode="$?"
                case $mode in
                    1)echo "1" >
~/CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
                esac | \
                yad --title="Metabolomics PROGRAM -- Step 4
Menu                                     Created by Giles
Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                Choose table file:":CB \
--field="                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice9}!${Choice10}!${Choice11}!${Choice12}!${Choice13}!${Choice14}!${Choice15}!${
Choice16}!${Choice17}!-!" "Heat Map!PCA!PLSDA" "MetabData_CrossConditionGraph.jpeg" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.CCRACD/tmp/MetabDataAndPlotSelected.txt >
~/CCRACD/tmp/MetabDataAndPlotSelected.txt
        else
            if [[ $NumberOfChoices = 18 ]];then
                mode="$?"
                case $mode in
                    1)echo "1" >
~/CCRACD/tmp/DataAndPlotMenu.txt && OpenMainMenu=1 ;;
                esac | \
                yad --title="Metabolomics PROGRAM -- Step
4 Menu                                     Created by
Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="                Choose table file:":CB \
--field="                Choose plot/graph type:":CB \
--field="Experiment name (no spaces, if spaces are needed please use _):" \

```



```

fi
fi
echo "1" > ~/CCRACD/tmp/DataAndPlotMenu.txt
OpenMainMenu=1
fi
fi
if [[ $OpenFile1 = 1 ]];then
    echo | ls ~/CCRACD/MetabolomicProgramOutput > ~/CCRACD/tmp/MetabSelectFile.txt

    cat ~/CCRACD/tmp/MetabSelectFile.txt | yad --title="                Metabolomics PROGRAM -
file selection                Created by Giles Holt" --list --column="Select files you wish to Open" --multiple
--width 800 --height 600 --center --align=center --button="Open all":1 --button="Open selected":0 --
button="Previous":2 --separator=" > ~/CCRACD/tmp/MetabSelectedFile.txt

mode="$?"
case $mode in
    0)OpenFile1=1 ;;
    1)OpenFile1=1 ;;
    2)OpenMainMenu=1 ;;
esac

if [[ $OpenFile1 = 1 ]];then
FileToOpen=$(head -n1 ~/CCRACD/tmp/MetabSelectedFile.txt)
FileToOpentxt=$(echo $FileToOpen | grep -c ".txt")

if [ $FileToOpentxt = 1 ];then
    gedit ~/CCRACD/MetabolomicProgramOutput/$FileToOpen
fi

FileToOpentxt=$(echo $FileToOpen | grep -c ".csv")

if [ $FileToOpentxt = 1 ];then
    gnome-open ~/CCRACD/MetabolomicProgramOutput/$FileToOpen
fi

FileToOpentxt=$(echo $FileToOpen | grep -c ".pdf")

if [ $FileToOpentxt = 1 ];then
    gnome-open ~/CCRACD/MetabolomicProgramOutput/$FileToOpen
fi

OpenMainMenu=1
fi

fi

## Re-Open script

if [[ "$OpenMainMenu" = 1 ]];then
~/CCRACD/MetabolomicsProgramMenu.sh
fi

```

10.8.3 Scripts

The following sub-sections contain all the scripts written in order to carry out the functions of CRACCD

10.8.3.1 Identifying compounds of interest

```
#!/bin/bash

InputFile=$(awk -F '|' '{print $1}'
~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt)
mz=$(awk -F '|' '{print $2}' ~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt)
mzMin=$(echo "$mz" | awk -F '|' '{print $1}')
mzMax=$(echo "$mz" | awk -F '|' '{print $2}')
rt=$(awk -F '|' '{print $3}' ~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt)
rtMin=$(echo "$rt" | awk -F '|' '{print $1}')
rtMax=$(echo "$rt" | awk -F '|' '{print $2}')
Pvalue=$(awk -F '|' '{print $4}' ~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt)
CV=$(awk -F '|' '{print $5}' ~/CCRACD/tmp/Metab_Stage1_FilterCompounds_FilesSelected.txt)

#make a copy of the input file, and make it a temp file while filtering
cp ~/CCRACD/MetabolomicProgram_Input/"$InputFile" ~/CCRACD/tmp/TempFilteringFile.csv

#Filter out non significant compounds

if [ "$Pvalue" = "N/A" ];then
echo "No P value filtering selected"
else

    awk -v n=$Pvalue '
function abs(x) {return x < 0 ? -x : x}
{print abs($0 - n) "\t" $0}' < | sort -n | head -n 1 ~/CCRACD/tmp/TempFilteringFile.csv >
~/CCRACD/tmp/tmp1_TempFilteringFile.csv

    #above creates a odd new column, so this removes that
    awk -F "$\t" 'BEGIN { OFS = FS } { print $2 }' ~/CCRACD/tmp/tmp1_TempFilteringFile.csv >
~/CCRACD/tmp/TempFilteringFile.csv
fi

#Filter out inconsistant data
if [ "$CV" = "-" ];then
echo "No CV% filtering selected"
else
    awk -v n=$CV '
function abs(x) {return x < 0 ? -x : x}
{print abs($0 - n) "\t" $0}' < | sort -n | head -n 1 ~/CCRACD/tmp/TempFilteringFile.csv >
~/CCRACD/tmp/tmp1_TempFilteringFile.csv

    #above creates a odd new column, so this removes that
```

```
awk -F $'\t' 'BEGIN { OFS = FS } { print $2 }' ~/CCRACD/tmp/tmp1_TempFilteringFile.csv >
~/CCRACD/tmp/TempFilteringFile.csv
fi
```

```
#Filtering for those withing retention range
if [ "$rt" = "-" ];then
echo "No retention range set for filtering"
else
awk -F $'\t' -v h=$rtMin -v l=$rtMax 'BEGIN { OFS = FS } $3 > h && $3 < l {print}'
~/CCRACD/tmp/TempFilteringFile.csv > ~/CCRACD/tmp/tmp1_TempFilteringFile.csv
mv ~/CCRACD/tmp/tmp1_TempFilteringFile.csv ~/CCRACD/tmp/TempFilteringFile.csv

fi
```

```
#Filtering for those withing retention range
if [ "$mz" = "-" ];then
echo "No mz range set for filtering"
else
awk -F $'\t' -v h=$mzMin -v l=$mzMax 'BEGIN { OFS = FS } $2 > h && $2 < l {print}'
~/CCRACD/tmp/TempFilteringFile.csv > ~/CCRACD/tmp/tmp1_TempFilteringFile.csv
mv ~/CCRACD/tmp/tmp1_TempFilteringFile.csv ~/CCRACD/tmp/TempFilteringFile.csv
fi
```

```
#change name back to edited version of input name and place back into input
mv ~/CCRACD/tmp/TempFilteringFile.csv
~/CCRACD/MetabolomicProgram_Input/"Filtered_$(InputFile)"
```

10.8.3.2 Clean-up compounds of interest

```
#!/bin/bash

# Set the retention time range given from settings
rtVar=$(awk -F "|" '{print $4}' ~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt)

#takes the name of the pos ion csv file selected
MetabInputData=$(awk -F "|" '{print $1}'
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt)

#takes the name of the neg ion csv file selected
MetabInputData_NegIon=$(awk -F "|" '{print $2}'
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt)

#sets the variable for m/z decimal place limit
mzDecVar=$(awk -F "|" '{print $3}'
~/CCRACD/tmp/Metab_Stage1_SelectMetabFix_FilesSelected.txt)

TrueTime1min=$(date +"%M")
TrueTime1hourtmp=$(date +"%H")
cd ~/

#This removes the output file if it already exists
if [[ -f ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData" ]];then
rm ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"
fi

#This removes the output file if it already exists
if [[ -f ~/CCRACD/tmp/CompoundThruConditionStats.txt ]];then
rm ~/CCRACD/tmp/CompoundThruConditionStats.txt
fi

if [[ -f ~/CCRACD/tmp/InputConditionRepeatStats.txt ]];then
rm ~/CCRACD/tmp/InputConditionRepeatStats.txt
fi

echo | (date +"

Metabolomics: Cross compound comparison through conditions - Start Date: %d-%m-%y Time: %T

"
)

echo | (date +"          Metabolomics: Cross compound comparison through conditions - Start
Date: %d-%m-%y Time: %T
```

```

"
)>> ~/CCRACD/tmp/InputConditionRepeatStats.txt

echo "

    Calculating the number of conditions, condition names, number of compounds per condition, and
    mz's and retention times...

"

echo "
The number of conditions, condition names, number of compounds per condition, and mz's and
retention times have been calculated

" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt

#creates a variable with a number in if no file was selected
WasAPosFileSelected=$(echo $MetabInputData | grep -c -x "-")

#creates a variable with a number in if no file was selected
WasANegFileSelected=$(echo $MetabInputData_NegIon | grep -c -x "-")

if [[ $WasAPosFileSelected = 0 ]];then
echo "A positive ion input file was selected"
fi
if [[ $WasANegFileSelected = 0 ]];then
echo "A Negative ion input file was selected"
fi

NumberOfFilesToRun=$( echo "$WasAPosFileSelected + $WasANegFileSelected" | bc )

#if no files were selected it would = 0
if [[ $NumberOfFilesToRun > 1 ]];then
notify-send "No files selected to run"
echo "No files selected to run"
echo "No files selected to run" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt
else

#loop to run positive ion file if there and negative ion file if there

for i in $(seq 1 2);do

if [[ $WasAPosFileSelected = 0 ]];then

echo "

#####                               #####

                Running the Positive ion file

#####                               #####

"

```

```

RunningFiletype="positive"
fi

if [[ $WasANegFileSelected = 3 ]];then
echo "

#####                               #####

                Running the negative ion file

#####                               #####

"

RunningFiletype="negative"
#Change WasAPosFileSelected to equal WasANegFileSelected
WasAPosFileSelected=$WasANegFileSelected

#Change WasAPosFileSelected to equal WasANegFileSelected
MetabInputData=$MetabInputData_NegIon

fi

#mz file with number cut to 1 decimal place - first grab the column containing mz's
awk -F $'\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt

if [[ $mzDecVar = 1 ]];then
#then cut to 1 decimal place without rounding up
grep -o '^[0-9]*\.[0-9]' ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
else
  if [[ $mzDecVar = 2 ]];then
  #then cut to 2 decimal place without rounding up
  grep -o '^[0-9]*\.[0-9][0-9]' ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
  else
    if [[ $mzDecVar = 3 ]];then
    #then cut to 3 decimal place without rounding up
    grep -o '^[0-9]*\.[0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
    else
      if [[ $mzDecVar = 4 ]];then
      #then cut to 4 decimal place without rounding up
      grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
      else
        if [[ $mzDecVar = 5 ]];then
        #then cut to 5 decimal place without rounding up

```

```

    grep -o '[0-9]*\.[0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
    else
        if [[ $mzDecVar = 6 ]];then
            #then cut to 6 decimal place without rounding up
            grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
        else
            if [[ $mzDecVar = 7 ]];then
                #then cut to 6 decimal place without rounding up
                grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
            else
                if [[ $mzDecVar = 8 ]];then
                    #then cut to 6 decimal place without rounding up
                    grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
                else
                    echo "No mz decimal place limit chosen"
fi
fi
fi
fi
fi
fi
fi
fi

```

```

#create file containing just retention times
awk -F $'\t' 'BEGIN { OFS = FS } {print $3}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt

```

```

#retention file with number cut to whole number
#awk -F $'\t' 'BEGIN { OFS = FS } $1=int($1)'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt

```

#####Compounds of interest Data file ##### ascertain condition name and number #####

#counts how many rows per condition - by looking for when the condition column begins to repeat and actually contains text

```

#fixes file into tab delimited if its comma delimited
cat ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | tr ',' '\t' >
~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabInputData}"
mv ~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabInputData}"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

```



```

#File for the table making script
echo "$ConditionNames" > ~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt

#use the condition list and grab all the lines for each condition and create a condition file with each
loop through
touch ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
ConditionLine=1
echo "The conditions are..."
echo "The conditions are..." >> ~/CCRACD/tmp/InputConditionRepeatStats.txt

for i in $(seq 1 $NumberOfConditions);do
Condition=$(echo "$ConditionNames" | awk -v x=$ConditionLine 'NR==x {print}' | tr -d ' ')
echo "$ConditionLine:${Condition}"
echo "$ConditionLine:${Condition}" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt
#creates each file for compounds of each condition
awk -v x=$Condition '$4==x {print}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${Condition}.txt
#create file containing number of compounds for each condition
grep -c "$Condition" ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >>
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
ConditionLine=$(( $ConditionLine + 1 ))
done

#####

##### ----- Quick minor pre-condition searching prep

#####

#make copy of original file
cp ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"
~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"

echo "

Retention time (rt) range set: +/- $rtVar minute/s"

echo "

mz - minimum number of matching decimal places: $mzDecVar"

##### ----- MAIN SECTION

##### Matches accross conditions #####

```

```
##### Start of Loop creation requirements
LoopNumber=1
CompoundOfInterestConditionNumber=1
CompoundConditionToCheckIn=$NumberOfConditions

NumberOfConditionsToCheckB4ChangingCompoundInterestCondition=$(( $NumberOfConditions -
1 ))
#

CalculatedNumberOfRepeatsForLoop=$(echo "(( $NumberOfConditions - 1 ) * 0.5) *
$NumberOfConditions" | bc | awk -F '.' '{print $1}')

for i in $(seq 1 $CalculatedNumberOfRepeatsForLoop);do
CompoundOfInterestCondition=$(awk -v x=$CompoundOfInterestConditionNumber 'NR==x'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
CompoundOfInterestCondition=$(echo "$CompoundOfInterestCondition" | tr -d ' ')
NameOfCompoundConditionToCheckIn=$(awk -v x=$CompoundConditionToCheckIn 'NR==x'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
NameOfCompoundConditionToCheckIn=$(echo "$NameOfCompoundConditionToCheckIn" | tr -d '
')
NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition=$(awk -v
x=${CompoundOfInterestCondition} 'NR==x'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt)

echo "
```

```
In $RunningFiletype ion file: $MetabInputData
Looking for Identical compounds from
$CompoundOfInterestCondition condition in $NameOfCompoundConditionToCheckIn condition...
```

```
"
```

```
echo "
```

```
Looked for Identical compounds from $CompoundOfInterestCondition condition in
$NameOfCompoundConditionToCheckIn condition...
```

```
" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt
```

```
#grabs mz column, file with number cut to 1 decimal place - first grab the column containing mz's
awk -F $'\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${CompoundOfInterestCo
ndition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec.txt
```

```
#ensures only mzs are in file (no titles or empty lines etc)
```

```
grep -o '[0-9]*\.[0-9]*'
```

```
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
```

```

dec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt

#####removes any lines where numbers from the list to the left of the decimal place that
are low or higher than the highest and lowest mz's in the compound to search for file
# same is done for retention time
#####

#Set the mz and RT's limits as variables
#get highest and lowest mz in condition input file
#order lines/rows by mz in input mz file and take last line - highest, take left of decimal place
mzMaxNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt | tail -n1 | awk -F '.' '{print $1}')
#add 1
mzMaxNumber=$(( $mzMaxNumber_prep + 1 ))
#take first line - lowest, take left of decimal place
mzMinNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt | awk -F '.' 'NR==1 {print $1}')
#minus 1
mzMinNumber=$(( $mzMinNumber_prep - 1 ))
#get highest and lowest RT in input file
awk -F '$\t' 'BEGIN { OFS = FS } {print $3}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${CompoundOfInterestCo
ndition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec.txt
grep -o '[0-9]*\.[0-9]*'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt
#order lines/rows by RT in input rt file, take last line - highest, take left of decimal place
rtMaxNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt | tail -n1 | awk -F '.' '{print $1}')
#add the variable for rt range
rtMaxNumber=$(( $rtMaxNumber_prep + $rtVar ))
#take first line - lowest, take left of decimal place
rtMinNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt | awk -F '.' 'NR==1 {print $1}')
#minus the variable for rt range
rtMinNumber=$(( $rtMinNumber_prep - $rtVar ))
#set to zero if its a negative
if (( "$rtMinNumber" < "0" ));then
rtMinNumber=0
fi

#create new file of those that fit mz range
awk -F '$\t' -v h=$mzMinNumber -v l=$mzMaxNumber 'BEGIN { OFS = FS } $2 > h && $2 < l
{print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCond

```

```

itionToCheckIn}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmed.txt

#create new file of those that fit rt range
awk -F '$\t' -v h=$rtMinNumber -v l=$rtMaxNumber 'BEGIN { OFS = FS } $3 > h && $3 < l {print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmed.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmedPrep.txt

mv
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmedPrep.txt
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmed.txt

awk -F '$\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfCompoundCon
ditionToCheckIn}_trimmed.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec.txt

#####
#adaptable for decimal number chosen
#####

#then cut to 1 decimal place without rounding up
#grep -o '^[0-9]*\.[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec.txt > ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con3mzdec1.txt

if [[ $mzDecVar = 1 ]];then
#then cut to 1 decimal place without rounding up
grep -o '^[0-9]*\.[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec1.txt
else
  if [[ $mzDecVar = 2 ]];then
    #then cut to 2 decimal place without rounding up
    grep -o '^[0-9]*\.[0-9][0-9]'
    ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec.txt >
    ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec1.txt
  else
    if [[ $mzDecVar = 3 ]];then
      #then cut to 3 decimal place without rounding up
      grep -o '^[0-9]*\.[0-9][0-9][0-9]'
      ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec.txt >
      ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCh
eckIn}mzdec1.txt
    else

```

```

else
  if [[ $mzDecVar = 4 ]];then
    #then cut to 4 decimal place without rounding up
    grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec1.txt
  else
    if [[ $mzDecVar = 5 ]];then
      #then cut to 5 decimal place without rounding up
      grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec1.txt
    else
      if [[ $mzDecVar = 6 ]];then
        #then cut to 6 decimal place without rounding up
        grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec1.txt
      else
        if [[ $mzDecVar = 7 ]];then
          #then cut to 6 decimal place without rounding up
          grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec1.txt
        else
          if [[ $mzDecVar = 8 ]];then
            #then cut to 6 decimal place without rounding up
            grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con${NameOfCompoundConditionToCheckIn}mzdec1.txt
          else
            echo "No mz decimal place limit chosen"
fi
  fi
fi
fi
fi
fi
fi
fi
fi

```

#add empty lines back in above the mz 1 dec file

```

#add spaces back in
EmptyLineSpace=$(( $SelectedCompounds_Condition1LineNumberStart - 1 ))

#loop to total number rows within condition
linenumber=$SelectedCompounds_Condition1LineNumberStart

for i in $(seq 1 $Condition1_NumberOfCompounds);do
#Setting the mzToMatch variable to the given line number until the condition is complete
mzToMatch=$(awk -v x=$linenumber '{FS=" "; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt)

echo "mz to match: $mzToMatch"

NumberOfmzToFindMatchFrom=$(grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con3mzdec1.txt)

linenumber_1=$SelectedCompounds_Condition1LineNumberStart

for i in $(seq 1 $NumberOfmzToFindMatchFrom);do

if [[ ! -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt ]];then
touch ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt
fi

#grab the $linenumber_1 from the mzlist to check from
mzToCheckForMatch=$(awk -v x=$linenumber_1 '{FS=" "; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con3mzdec1.txt)

#this subtracts one from the other, if they are equal the answer will be 0, i run it through tr because it
collects any numbers from the output, removing problems of negatives etc
PerfectMatch=$(echo "$mzToCheckForMatch - $mzToMatch" | bc | tr -dc '0-9')

if [[ "$PerfectMatch" == 0 ]];then
echo "Found initial match for mz: $mzToMatch"

sed -n ${linenumber_1}p
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon3.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt

fi

linenumber_1=$(( $linenumber_1 + 1 ))

done

#resets line number after the above mini-loop
linenumber_1=$SelectedCompounds_Condition1LineNumberStart

if [[ ! -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt ]];then
touch ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt
fi

mzmatched=$(grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt)

```

```

if [[ "$mzmatched" > 0 ]];then

echo "Checking retention time (rt) of the mz match"

#set rt to match variable to the given line number until the condition is complete
rtToMatch=$(awk -v x=$linenumber '{FS=" " } ; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt)

echo "rt to match to: $rtToMatch"

#1 decimal lower than match
rtminHigher=$(echo "scale=2; $rtToMatch -$rtVar" | bc)

if [[ $(echo "$rtminHigher < 0" | bc) -eq 1 ]];then
rtminHigher=0
fi

echo "Lower rt limit: $rtminHigher"

#1 decimal higher than match
rtminLower=$(echo "scale=2; $rtToMatch +$rtVar" | bc)

echo "Upper rt limit: $rtminLower"

#find matches within 1 minute of the rt to match
awk -v h=$rtminHigher -v l=$rtminLower -F $'\t' 'BEGIN { OFS = FS } $3 > h && $3 < l {print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt

NumberMzRtMatches=$( grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt )

echo "For mz: $mzToMatch - Number of matches with rt between $rtminHigher and $rtminLower :
$NumberMzRtMatches "

else
if [[ ! -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt ]];then
touch ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt
fi

if [[ ! -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt ]];then
touch ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt
fi

fi

#cut to closest mz value by first taking the numbers after the decimal place and picking the ones with
the closest match to original mz decimals

#number of matches
NumberMzRtMatches=$( grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt )

echo "Number of matches after initial mz and RT filter: $NumberMzRtMatches"

#if number of matches > 1 then cut to closest to exact mz

```

```

if [[ $NumberMzRtMatches > 1 ]] && [[ $NumberMzRtMatches != 0 ]];then

echo "Filtering for the absolute closest mz match"
#set mz to match as just the full number after the decimal point
mzToMatch=$(awk -v x=$linenumber 'NR==x'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt | cut -f2 -d ".")

#create a file containing the numbers after the decimal place for possible matches thus far
awk -F $'\t' 'BEGIN { OFS = FS } { print $2 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt | cut -f2 -d "."
> ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp1.txt

#find the closest match to the above file
awk -v n=$mzToMatch '
function abs(x) {return x < 0 ? -x : x}
{print abs($0 - n) "\t" $0}' <
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp1.txt |
sort -n |
head -n 1 > ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp2.txt

#above creates a odd new column, so this removes that
awk -F $'\t' 'BEGIN { OFS = FS } { print $2 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp2.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp3.txt

#set above file first option as variable
ExactDecMatch=$(head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp3.txt)

echo "Final mz match: $ExactDecMatch"

#re-grabs the full line for the one that was an exact match
grep "$ExactDecMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch.txt

#same as above but only stores the current loops output - this is for creating the compound number ID
adjustment at the end of each loop
grep "$ExactDecMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt

#where there was instantly only one match it puts it straight into the final collected file of repeat
compounds
else

echo "No further filtering required"

head -n1 ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch.txt

#same as above but only stores the current loops output - this is for creating the compound number ID
adjustment at the end of each loop
head -n1 ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt

```

```

fi

#this is a zero if no match is found which messes up the rest, to ensure this doesn't happen i re-set the
match number to the final file and check that it equals 1 in the if function
NumberMzRtMatches=$( grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt )

if [[ $NumberMzRtMatches = 1 ]];then

#sets the single match as variable
RepeatCompoundInAlternativeCondition=$(head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt)

echo "Matching compounds data: $RepeatCompoundInAlternativeCondition"

#sets the found matches incorrect compound number as a variable
RepeatCompoundNumber=$(awk -F $'\t' 'BEGIN { OFS = FS } { print $1 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt)

echo "Matching compounds number is: $RepeatCompoundNumber"

#gets the metabolite names from condition 1
CompoundNumbersInConditionSearchingFrom=$(awk -F $'\t' 'BEGIN { OFS = FS } { print $1 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon1.txt)

#counts if the compound name is already from condition 1 i.e. has it already been changed
HasItAlreadyBeenChanged=$( echo "$CompoundNumbersInConditionSearchingFrom" | grep -o
"$RepeatCompoundNumber" | wc -l )

#checks if the compound has already been changed before changing it now
if [[ "$HasItAlreadyBeenChanged" = 0 ]]
then

#sets the compound number you've been searching for as a variable
OriginalCompoundNumber=$(awk -v x=$linenumber -F $'\t' 'BEGIN { OFS = FS } NR==x { print $1
}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

echo "Original compounds number: $OriginalCompoundNumber"

#make the actual line number of the found match a variable
RepeatLineNumber=$(grep -n "$RepeatCompoundInAlternativeCondition"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | cut -f1 -d ":")

echo "The match was found on line number: $RepeatLineNumber"

awk -v a=$RepeatLineNumber -v b=$OriginalCompoundNumber -F $'\t' 'BEGIN { OFS = FS }
NR==a{$1=b}1' ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData" > tmp && mv tmp
~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"

echo "Matching compound number correctly adjusted to: $OriginalCompoundNumber "

else echo "This compound number has been previously adjusted/corrected"

fi

```

```

fi

#set a variable to let the user know what compounds were searched for
SearchForCompound=$(awk -v x=$linenumber -F $'\t' 'BEGIN { OFS = FS } NR==x {print $1}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

NumberMzRtMatches=$( grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt )

if [[ $NumberMzRtMatches = 1 ]]
then

echo "A successful match was found and adjusted as stated above"

else

echo "No match was found"

fi

echo "Compound $SearchForCompound check complete"

"

linenumber=$(( $linenumber + 1 ))

if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp3.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp3.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp2.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp2.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp1.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_tmp.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt ]
then rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch_NumbFix.txt
fi

done

NumberInCondition=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMz_RtMatch.txt)

echo "

```

Search for Identical compounds from \$CompoundOfInterestCondition condition in \$ConditionName3 condition is complete, there are \$NumberInCondition repeat compounds found. There given names have been adjusted appropriately

"

echo "

The search for Identical compounds from \$CompoundOfInterestCondition condition in \$ConditionName3 condition was complete, there are \$NumberInCondition repeat compounds found. Their given names have been adjusted appropriately

" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt

#Refresh the comparison files with the most uptodate and changed file

```
if [ ! -f ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData" ]
then echo ""
else
```

```
#make a copy of the original data file and remove lines that contain condition 1
cp ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
```

```
#removing the lines that contain the conditions not wanting to compare against
sed "$CompoundOfInterestCondition/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
sed "$ConditionName2/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon3.txt
```

```
cp ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
#removing the lines that contain the conditions not wanting to compare against
sed "$CompoundOfInterestCondition/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
sed "$ConditionName3/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon2.txt
```

```
cp ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
```

```
#removing the lines that contain the conditions not wanting to compare against
sed "$ConditionName2/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
sed "$ConditionName3/d"
~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon1.txt
fi
```

```

cp ~/CCRACD/MetabolomicProgramOutput/1_"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/1_"$MetabInputData"

rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon2.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon3.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataCon2CloseMz_RtMatch_tmp
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_Con2mzdec1.txt

##remember to change the coulmsn through out this for the associated numbers

WasANegFileSelected=$(( $WasANegFileSelected + 3 ))

WasAPosFileSelected=$(( $WasAPosFileSelected + 1 ))

done

fi

echo "

Giles Holt Metabolomics Program has COMPLETE the following process- cross comparison for
identifying identical compounds accross conditions"

echo "

Giles Holt Metabolomics Program COMPLETE the following process - cross comparison for
identifying identical compounds accross conditions" >>
~/CCRACD/tmp/InputConditionRepeatStats.txt

echo | (date +"          Metabolomics: Cross compound comparison through conditions -
Completion Date: %d-%m-%y Time: %T

"
)

echo | (date +"          Metabolomics: Cross compound comparison through conditions -
Completion Date: %d-%m-%y Time: %T

"
) >> ~/CCRACD/tmp/InputConditionRepeatStats.txt

echo          |          ~/CCRACD/Scripts/InputConditionRepeatStatsPostRunWindow.sh          &&

~/CCRACD/MetabolomicsProgramMenu.sh

```

10.8.3.3 Compound flux through conditions

```
#!/bin/bash

# Set the retention time range given from settings
rtVar=$(awk -F "|" '{print $6}' ~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

#sets the variable for mz decimal place limit
mzDecVar=$(awk -F "|" '{print $5}' ~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

#takes the name of the csv file selected
MetabInputData=$(awk -F "|" '{print $1}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

#takes the name of the neg ion csv file selected
MetabInputData_NegIon=$(awk -F "|" '{print $2}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

MetabRawPosInputData=$(awk 'BEGIN {FS = "|"} {print $4}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

MetabRawNegInputData=$(awk 'BEGIN {FS = "|"} {print $3}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

TrueTime1min=$(date +%M)
TrueTime1hourtmp=$(date +%H)
cd ~/

echo "
Calculating number of conditions, condition names, number of compounds per condition, and mz's
and retention times, in the data file and the Raw data file...

"

#This removes the output file if it already exists
if [[ -f ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData" ]];then
rm ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData"
fi

if [[ -f ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawNegInputData" ]];then
rm ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawNegInputData"
fi

#second removal seems unnecessary# check if needed
if [[ -f ~/CCRACD/tmp/CompoundThruConditionStats.txt ]];then
rm ~/CCRACD/tmp/CompoundThruConditionStats.txt
fi

#This removes the output file if it already exists
if [[ -f ~/CCRACD/tmp/CompoundThruConditionStats.txt ]];then
rm ~/CCRACD/tmp/CompoundThruConditionStats.txt
fi
```

```
echo | (date +"          Metabolomics: Cross compound comparison through conditions - Start
Date: %d-%m-%y Time: %T
```

```
"
)
```

```
echo | (date +"          Metabolomics: Cross compound comparison through conditions - Start
Date: %d-%m-%y Time: %T
```

```
"
```

```
) >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
echo "
```

```
Calculating the number of conditions, condition names, number of compounds per condition, and
mz's and retention times...
```

```
"
```

```
echo "
```

```
The number of conditions, condition names, number of compounds per condition, and mz's and
retention times have been calculated
```

```
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
#creates a variable with a number in if no file was selected
```

```
WasAPosFileSelected=$(echo $MetabInputData | grep -c -x "-")
```

```
#creates a variable with a number in if no file was selected
```

```
WasANegFileSelected=$(echo $MetabInputData_NegIon | grep -c -x "-")
```

```
if [[ $WasAPosFileSelected = 0 ]];then
```

```
echo "A positive ion input file was selected"
```

```
fi
```

```
if [[ $WasANegFileSelected = 0 ]];then
```

```
echo "A Negative ion input file was selected"
```

```
fi
```

```
NumberOfFilesToRun=$(( $WasAPosFileSelected + $WasANegFileSelected ))
```

```
#if no files were selected it would = 0
```

```
if [[ $NumberOfFilesToRun > 1 ]];then
```

```
notify-send "No files selected to run"
```

```
echo "No files selected to run"
```

```
echo "No files selected to run" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt
```

```
else
```

```
#loop to run positive ion file if there and negative ion file if there
```

```
for i in $(seq 1 2)
```

```
do
```

```
if [[ $WasAPosFileSelected = 0 ]]
```

```
then
```

```
echo "
```

```

#####
Running the Positive ion file
#####

"
RunningFiletype="positive"
fi

if [[ $WasANegFileSelected = 3 ]];then
echo "

#####

Running the negative ion file
#####

"
RunningFiletype="Negative"
#Change WasAPosFileSelected to equal WasANegFileSelected
WasAPosFileSelected=$(( $WasANegFileSelected ))

#Change WasAPosFileSelected to equal WasANegFileSelected
MetabInputData=$MetabInputData_NegIon

#Change MetabRawPosInputData to equal MetabRawNegInputData
MetabRawPosInputData=$MetabRawNegInputData

fi

#mz file with number cut to desired decimal place - first grab the column containing mz's
awk -F $'\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt

if [[ $mzDecVar = 1 ]];then
#then cut to 1 decimal place without rounding up
grep -o '[0-9]*\.[0-9]' ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
else
if [[ $mzDecVar = 2 ]];then
#then cut to 2 decimal place without rounding up
grep -o '[0-9]*\.[0-9][0-9]' ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
else
if [[ $mzDecVar = 3 ]];then
#then cut to 3 decimal place without rounding up
grep -o '[0-9]*\.[0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt

```

```

else
  if [[ $mzDecVar = 4 ]];then
    #then cut to 4 decimal place without rounding up
    grep -o '[0-9]*\.[0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
  else
    if [[ $mzDecVar = 5 ]];then
      #then cut to 5 decimal place without rounding up
      grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
    else
      if [[ $mzDecVar = 6 ]];then
        #then cut to 6 decimal place without rounding up
        grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
      else
        if [[ $mzDecVar = 7 ]];then
          #then cut to 6 decimal place without rounding up
          grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
        else
          if [[ $mzDecVar = 8 ]];then
            #then cut to 6 decimal place without rounding up
            grep -o '[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
          else
            echo "No mz decimal place limit chosen"
          fi
        fi
      fi
    fi
  fi

```

```

#####-----
#####

#####----- Prep for Compounds of interest file -----
#####

#####-----
#####

```

#retention file with number cut to whole number

```

awk -F $'\t' 'BEGIN { OFS = FS } {print $3}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt

#####Compounds of interest Data file ##### ascertain condition name and number #####

#counts how many rows per condition - by looking for when the condition column begins to repeat
and actually contains text

#fixes file into tab delimited if its comma delimited
cat ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | tr '[,]' '\t' >
~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabInputData}"
mv ~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabInputData}"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

StopCheckAsFoundAnswer=2
Find1stCompound1stRow=1
for i in $(seq 1 15);do
    if [[ "$StopCheckAsFoundAnswer" = "2" ]];then
        Find1stCompound2ndRow=$(( $Find1stCompound1stRow + 1 ))
        SelectedCompounds_CheckRows1="$(awk -F $'\t' -v x=$Find1stCompound1stRow 'BEGIN {
OFS = FS } ; NR==x {print $4}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")"
        SelectedCompounds_CheckRows2="$(awk -F $'\t' -v x=$Find1stCompound2ndRow 'BEGIN
{ OFS = FS } ; NR==x {print $4}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")"
        echo "SelectedCompounds_CheckRows1: $SelectedCompounds_CheckRows1"
        echo "SelectedCompounds_CheckRows2: $SelectedCompounds_CheckRows2"
        if [[ "$SelectedCompounds_CheckRows1" == "$SelectedCompounds_CheckRows2" ]] && [[
! -z "$SelectedCompounds_CheckRows1" ]];then
            SelectedCompounds_Condition1LineNumberStart="$Find1stCompound1stRow"
            echo "
The first condition starts on line: $SelectedCompounds_Condition1LineNumberStart
"
            echo "
The first condition started on line: $SelectedCompounds_Condition1LineNumberStart
">> ~/CCRACD/tmp/CompoundThruConditionStats.txt
            StopCheckAsFoundAnswer=1
        fi
        Find1stCompound1stRow=$(( $Find1stCompound1stRow + 1 ))
    fi
done

#set the number of empty lines to additional
EmptyLineSpace=$(( $SelectedCompounds_Condition1LineNumberStart - 1 ))

#creates the total number of rows containing data as a variable
TotalRowsOfData=$(awk -F $'\t' 'BEGIN { OFS = FS } ; {print $1}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | sed -n
"$((($SelectedCompounds_Condition1LineNumberStart)),$p' | grep -c '[0-9]')

echo "The input $MetabInputData file contains $TotalRowsOfData compounds in which to search for
matches in the $MetabRawPosInputData"

#add empty lines back in above the mz 1 dec file #loop
for i in $(seq 1 $EmptyLineSpace);do
    sed -i '1i\\' ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt

```

```

done

##-----
##### create files for each condition from the compounds of interest file
##-----

#make a list of condition names
ConditionNames=$(awk -F $'\t' -v x=$SelectedCompounds_Condition1LineNumberStart 'NR>x
{print $4}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | uniq)
#count number of conditions
NumberOfConditions=$(echo "$ConditionNames" | wc -l)
echo "$NumberOfConditions conditions have been identified"
echo "$NumberOfConditions conditions have been identified" >>
~/CCRACD/tmp/CompoundThruConditionStats.txt

#File for the table making script
echo "$ConditionNames" > ~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt

#use the condition list and grab all the lines for each condition and create a condition file with each
loop through
touch ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
ConditionLine=1
echo "The conditions are..."
echo "The conditions are..." >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
for i in $(seq 1 $NumberOfConditions);do
Condition=$(echo "$ConditionNames" | awk -v x=$ConditionLine 'NR==x {print}' | tr -d ' ')
echo "$ConditionLine:${Condition}"
echo "$ConditionLine:${Condition}" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
awk -v x=$Condition '$4==x {print}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${Condition}.txt
#create file containing number of compounds for each condition
grep -c "$Condition" ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >>
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
ConditionLine=$(( $ConditionLine + 1 ))
#dont see need to calculate where conditions start - but thats something previous version did

done

#####

##### ----- Raw Data file Prep info: Calculation of
compound numbers and Conditions, and seperation of raw data files -----
#####

#####

echo "$MetabRawPosInputData" > ~/CCRACD/tmp/Metab_RawCrossCondition_PosFileSelected.txt

CheckingRawFileSelected=$( grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/tmp/Metab_RawCrossCondition_PosFileSelected.txt )
if [[ $CheckingRawFileSelected = 0 ]];then
notify-send "No $RunningFiletype ion file selected to run"
echo "No $RunningFiletype ion file selected to run"

```

```

fi

#####

#####-----counts how many rows per condition - by looking for when the condition
column begins to repeat and actually contains text

#fixes file into tab delimited if its comma delimited
cat ~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" | tr '[,]' '[\t]' >
~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabRawPosInputData}"
mv ~/CCRACD/MetabolomicProgram_Input/tmp_"${MetabRawPosInputData}"
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData"

#identifies when the first condition begins in the raw file
StopCheckAsFoundAnswer=2
Find1stCompound1stRow=1
for i in $(seq 1 15);do
    if [[ "$StopCheckAsFoundAnswer" = "2" ]];then
        Find1stCompound2ndRow=$(( $Find1stCompound1stRow + 1 ))
        SelectedCompounds_CheckRows1="$(awk -F '$\t' -v x=$Find1stCompound1stRow 'BEGIN {
OFS = FS } ; NR==x {print $4}'
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData")"
        SelectedCompounds_CheckRows2="$(awk -F '$\t' -v x=$Find1stCompound2ndRow 'BEGIN
{ OFS = FS } ; NR==x {print $4}'
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData")"
        if [[ "$SelectedCompounds_CheckRows1" == "$SelectedCompounds_CheckRows2" ]] && [[
! -z "$SelectedCompounds_CheckRows1" ]]
        then
            Condition1LineNumberStart="$Find1stCompound1stRow"
            echo "
            The first condition in the raw $RunningFiletype ion file starts on line:
            $Condition1LineNumberStart
            "
            echo "
            The first condition in the raw $RunningFiletype ion file starts on line: $Condition1LineNumberStart
            " >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
            StopCheckAsFoundAnswer=1
        fi
        Find1stCompound1stRow=$(( $Find1stCompound1stRow + 1 ))
    fi
done

##-----
##### create files for each condition from the raw compounds file
##-----

#set the number of empty lines to additional
EmptyLineSpaceRaw=$(( $Condition1LineNumberStart - 1 ))

#creates the total number of rows containing data as a variable
TotalRowsOfRawData=$(awk -F '$\t' 'BEGIN { OFS = FS } ; {print $1}'
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" | sed -n
"$(( $Condition1LineNumberStart ))", $p' | grep -c '[0-9]')

echo "

```

The \$MetabRawPosInputData \$RunningFiletype ion file contains \$TotalRowsOfRawData compounds in which to search for matches

"

#make a list of condition names

```
RawConditionNames=$(awk -F '$\t' -v x=$Condition1LineNumberStart 'NR>x {print $4}'  
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" | uniq)
```

#count number of conditions

```
NumberOfRawConditions=$(echo "$RawConditionNames" | wc -l)  
echo "${NumberOfRawConditions} conditions have been identified"  
echo "${NumberOfRawConditions} conditions have been identified" >>  
~/CCRACD/tmp/CompoundThruConditionStats.txt
```

#File for the table making script

```
echo "$RawConditionNames" > ~/CCRACD/MetabolomicProgramOutput/RawConditionlist.txt
```

#use the condition list and grab all the lines for each condition and create a condition file with each loop through

```
touch ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_RawEachConditionNo.txt
```

```
RawConditionLine=1
```

```
echo "The conditions are..."
```

```
echo "The conditions are..." >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
for i in $(seq 1 $NumberOfRawConditions);do
```

```
RawCondition=$(echo "$RawConditionNames" | awk -v x=$RawConditionLine 'NR==x {print}' | tr -  
d ' ')
```

```
echo "${RawConditionLine}:${RawCondition}"
```

```
echo "${RawConditionLine}:${RawCondition}" >>
```

```
~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
awk -v x=$RawCondition '$4==x {print}'
```

```
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" >
```

```
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCondition}.txt
```

#create file containing number of compounds for each condition

```
grep -c "$RawCondition" ~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" >>
```

```
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_RawEachConditionNo.txt
```

```
RawConditionLine=$(( $RawConditionLine + 1 ))
```

#dont see need to calculate where conditions start - but thats something previous version did

done

#####

----- Quick minor pre-condition searching prep

#####

#make copy of original file

```
cp ~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData"
```

```
~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData"
```

#NumberOfConditions

```
NumberOfConditions=$(grep -c -v "HelloThisIsMyAntiMatch"
```

```
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
```

```
echo "
```

Retention time (rt) range set: +/- \$rtVar minute/s"

echo "

mz - minimum number of matching decimal places: \$mzDecVar"

----- MAIN SECTION

#####----- Compound search through Raw files

-----#####

Matching the selected metabolites back to different conditions in the raw data

#

#####

Start of Loop creation requirements

LoopNumber=1

CompoundOfInterestConditionNumber=1

RawCompoundConditionNumber=\$NumberOfRawConditions

NumberOfRawConditionsToCheckB4ChangingCompoundInterestCondition=\$((

\$NumberOfConditions - 1))

CalculatedNumberOfRepeatsForLoop=\$(echo "(\$NumberOfRawConditions - 1) *

\$NumberOfConditions" | bc)

for i in \$(seq 1 \$CalculatedNumberOfRepeatsForLoop);do

#skips to the next condition in the raw file if the condition matches the compounds of interest condition

if [["\$RawCompoundConditionNumber" = "\$CompoundOfInterestConditionNumber"]];then

RawCompoundConditionNumber=\$((\$RawCompoundConditionNumber - 1))

fi

CompoundOfInterestCondition=\$(awk -v x=\$CompoundOfInterestConditionNumber 'NR==x'

~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)

CompoundOfInterestCondition=\$(echo "\$CompoundOfInterestCondition" | tr -d ' ')

echo "CompoundOfInterestCondition: \$CompoundOfInterestCondition"

RawCompoundCondition=\$(awk -v x=\$RawCompoundConditionNumber 'NR==x'

~/CCRACD/MetabolomicProgramOutput/RawConditionlist.txt)

RawCompoundCondition=\$(echo "\$RawCompoundCondition" | tr -d ' ')

NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition=\$(awk -v

x=\${CompoundOfInterestConditionNumber} 'NR==x'

~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt)

echo "NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition:

\$NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition"

echo "

Looking for Identical compounds from \$MetabInputData \$RunningFiletype ion file

(\$CompoundOfInterestCondition condition) in \$MetabRawPosInputData file

(\$RawCompoundCondition condition)...

"

echo "

Looked for Identical compounds from \$MetabInputData \$RunningFiletype ion file

(\$CompoundOfInterestCondition condition) in \$MetabRawPosInputData file

(\$RawCompoundCondition condition)...

" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt

End of Loop creation requirements

```

#sets rt range allowed - not sure i need to set this again
rtVar=$(awk -F "|" '{print $6}' ~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)
#grabs mz column
awk -F $'\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${CompoundOfInterestCo
ndition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec.txt
#ensures only mzs are in file (no titles or empty lines etc)
grep -o '[0-9]*\.[0-9]*'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt

#####removes any lines where numbers from the list to the left of the decimal place that
are low or higher than the highest and lowest mz'z in the compound to search for file
# same is done for retention time
#####

#Set the mz and RT's limits as variables
#get highest and lowest mz in condition input file
#order lines/rows by mz in input mz file and take last line - highest, take left of decimal place
mzMaxNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt | tail -n1 | awk -F '.' '{print $1}')
#add 1
mzMaxNumber=$(( $mzMaxNumber_prep + 1 ))
#take first line - lowest, take left of decimal place
mzMinNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_mz
dec1.txt | awk -F '.' 'NR==1 {print $1}')
#minus 1
mzMinNumber=$(( $mzMinNumber_prep - 1 ))
#get highest and lowest RT in input file
awk -F $'\t' 'BEGIN { OFS = FS } {print $3}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${CompoundOfInterestCo
ndition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec.txt
grep -o '[0-9]*\.[0-9]*'
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt
#order lines/rows by RT in input rt file, take last line - highest, take left of decimal place
rtMaxNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt | tail -n1 | awk -F '.' '{print $1}')
#add the variable for rt range
rtMaxNumber=$(( $rtMaxNumber_prep + $rtVar ))
#take first line - lowest, take left of decimal place

```

```

rtMinNumber_prep=$(sort -b -n
~/CCRACD/MetabolomicProgramOutput/MetabDataCon${CompoundOfInterestCondition}_tmp_rtd
ec1.txt | awk -F ' ' 'NR==1 {print $1}')
#minus the variable for rt range
rtMinNumber=$(( $rtMinNumber_prep - $rtVar ))
#set to zero if its a negative
if (( "$rtMinNumber" < "0" ));then
rtMinNumber=0
fi
#create new file of those that fit mz range
awk -F '$\t' -v h=$mzMinNumber -v l=$mzMaxNumber 'BEGIN { OFS = FS } $2 > h && $2 < l
{print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt

#awk -F ' ' -v h=$mzMinNumber -v l=$mzMaxNumber -F '$\t' 'BEGIN { OFS = FS } $2 > h && $2
< l {print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt

#create new file of those that fit rt range
awk -F '$\t' -v h=$rtMinNumber -v l=$rtMaxNumber 'BEGIN { OFS = FS } $3 > h && $3 < l {print}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmedPrep.txt

mv
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmedPrep.txt
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt

awk -F '$\t' 'BEGIN { OFS = FS } {print $2}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt

#then cut to user specified decimal place without rounding up
if [[ $mzDecVar = 1 ]];then
#then cut to set decimal place without rounding up
grep -o '^[0-9]*\.[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
else
if [[ $mzDecVar = 2 ]];then
#then cut to 2 decimal place without rounding up

```

```

    grep -o '^[0-9]*\.[0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
    else
        if [[ $mzDecVar = 3 ]];then
            #then cut to 3 decimal place without rounding up
            grep -o '^[0-9]*\.[0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
        else
            if [[ $mzDecVar = 4 ]];then
                #then cut to 4 decimal place without rounding up
                grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
            else
                if [[ $mzDecVar = 5 ]];then
                    #then cut to 5 decimal place without rounding up
                    grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
                else
                    if [[ $mzDecVar = 6 ]];then
                        #then cut to 6 decimal place without rounding up
                        grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
                    else
                        if [[ $mzDecVar = 7 ]];then
                            #then cut to 6 decimal place without rounding up
                            grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
                        else
                            if [[ $mzDecVar = 8 ]];then
                                #then cut to 6 decimal place without rounding up
                                grep -o '^[0-9]*\.[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt
                            else

```

```

                echo "No mz decimal place limit chosen"
            fi
        fi
    fi
fi

##### loop until total number rows within condition 1 of the input data file are complete
EmptyLineSpace=$(( $Condition1LineNumberStart - 1 ))

if [[ "$PreviousLineNumber" > 0 ]] && [ ! "$CompoundOfInterestCondition" ==
"$RememberCompoundOfInterestCondition" ];then
#echo "linenumber: $linenumber"
linenumber=$PreviousLineNumber
#echo "linenumber: $linenumber"
else
    if [[ "$RememberLineNumber" > 0 ]];then
        if [[ "$RememberLineNumber" = "$SelectedCompounds_Condition1LineNumberStart" ]];then
            linenumber=$SelectedCompounds_Condition1LineNumberStart
        else
            linenumber=$RememberLineNumber
        fi
    else
        linenumber=$SelectedCompounds_Condition1LineNumberStart
    fi
fi
RememberCompoundOfInterestCondition=$CompoundOfInterestCondition
RememberLineNumber=$linenumber

echo "
#####
After

PreviousLineNumber: $PreviousLineNumber
CompoundOfInterestCondition: $CompoundOfInterestCondition
RememberCompoundOfInterestCondition: $RememberCompoundOfInterestCondition
linenumber: $linenumber
RememberLineNumber: $RememberLineNumber

#####"

# date started
TimeStartmin=$(date +%M")
TimeStarthourtmp=$(date +%H")
echo "The program run was started at: ${TrueTime1hourtmp}:${TrueTime1min}"
echo "Searching for comparable compounds in condition ${RawCompoundCondition}. Time started:
${TimeStarthourtmp}:${TimeStartmin}"

TimeStarthour=$( echo "$TimeStarthourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$TimeStarthour + $TimeStartmin" | bc )

```

```

NumberTimesThrough=0

for i in $(seq 1 $NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition);do
#####
#TIME CALCULATION
Time1min=$(date +%M)
Time1hourtmp=$(date +%H)
echo "Current time: ${Time1hourtmp}:${Time1min}"

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

#this version is for the continued calculation of total time left
if (( ${TimeTaken} >= 1 ));then
echo "Time taken to complete $NumberTimesThrough run through/s: ${TimeTaken} minutes"
else echo "Time taken to complete $NumberTimesThrough run through/s: < 1 minute"
fi

Condition1_NumberOfCompoundstmp=$( echo
"$NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition - $NumberTimesThrough" |
bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $NumberTimesThrough) *
$Condition1_NumberOfCompoundstmp" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

if (( ${TimeTaken} >= 1 ));then
echo "
Searching for comparable compounds in condition ${RawCompoundCondition}...
Estimated time left until completion of this section: $EstimatedtimetoFinishMins minutes
(${EstimatedtimetoFinishHours} hours)"
else echo "Searching for comparable compounds in condition ${RawCompoundCondition}...
Still calculating the estimated time for the completion of this section"
fi

##### estimate the time it will take in total for an ion Run

if (( "$TimeTaken" > 1 ));then
EstimatedCompleteFinishMins=$( echo "(( $EstimatedtimetoFinishMins + $TimeTaken) *
$CalculatedNumberOfRepeatsForLoop) - $TimeTaken" | bc )
EstimatedCompleteFinishHours=$( echo "$EstimatedCompleteFinishMins / 60" | bc )
echo "Estimated time left until total completion: $EstimatedCompleteFinishMins minutes
(${EstimatedCompleteFinishHours} hours)"
fi

#####

#finds compound name (number)
CompoundNumber1MatchCheck=$(awk -F $'\t' 'BEGIN { OFS = FS } {print $1}'
~/CCRACD/MetabolomicProgram_Input/$MetabInputData | sed -n ${linenumber}p)
echo "

```

```
Searching for compound $CompoundNumber1MatchCheck from $MetabInputData $RunningFiletype
ion file in $MetabRawPosInputData $RunningFiletype ion file (condition
${RawCompoundCondition})
"
```

```
#looks for a match of the compound number/name in the condition (that will have its raw data
searched) of the compounds of interest file (basically helps make sure compound insint already
sorted)
```

```
CompoundNumber2MatchCheck=$(awk -F $'\t' 'BEGIN { OFS = FS } {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${RawCompoundConditio
n}.txt | grep "\b$CompoundNumber1MatchCheck\b")
```

```
#looks for a match of the compound number in condition 3 of the raw files
```

```
CompoundNumber3MatchCheck=$(awk -F $'\t' 'BEGIN { OFS = FS } {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt | grep "\b$CompoundNumber1MatchCheck\b")
```

```
#check if the compound has already been found in the other condition before looking into the raw
positive file
```

```
if [[ $CompoundNumber1MatchCheck != $CompoundNumber2MatchCheck ]] && [[
$CompoundNumber1MatchCheck != $CompoundNumber3MatchCheck ]];then
```

```
echo "
```

```
Compound Name '${CompoundNumber1MatchCheck}' is not already present in the
${RawCompoundCondition} condition of the $MetabInputData $RunningFiletype ion file, nor has it
been previously found and adjusted in the ${RawCompoundCondition} condition of the
$MetabRawPosInputData $RunningFiletype ion file.
```

```
Searching for it in the ${RawCompoundCondition} condition of the $MetabRawPosInputData file
"
```

```
echo "
```

```
Compound Name '${CompoundNumber1MatchCheck}' is not already present in the
${RawCompoundCondition} condition of the $MetabInputData $RunningFiletype ion file, nor has it
been previously found and adjusted in the ${RawCompoundCondition} condition of the
$MetabRawPosInputData $RunningFiletype ion file.
```

```
Searching for it in the ${RawCompoundCondition} condition of the $MetabRawPosInputData file
```

```
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
#Setting the mzToMatch variable to the given line number until the condition is complete
```

```
mzToMatch=$(awk -v x=$linenumber 'FS=" "; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt)
```

```
echo "mz to match to: $mzToMatch"
```

```
echo "mz to match to: $mzToMatch" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
NumberOfmzToFindMatchFrom=$(grep -c '[0-9]'
```

```
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt)
```

```
countmzMatch=0
```

```
linenumber_1=1
```

```
for i in $(seq 1 $NumberOfmzToFindMatchFrom);do
```

```

    if [[ ! -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt ]];then
    touch
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt
    fi
    #moved further up as otherwise this gets reset in the loop - touch
~/CCRACD/MetabolomicProgramOutput/MetabDataCon3CloseMzMatch.txt

    #grab the $linenumber_1 from the mzlist to check from
    mzToCheckForMatch=$(awk -v x=$linenumber_1 '{FS=" "; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${RawCompoundCondition}mzd
ec1.txt)

    #this subtracts one from the other, if they are equal the answer will be 0, i run it through tr because
it collects any numbers from the output, removing problems of negatives etc
    PerfectMatch=$(echo "$mzToCheckForMatch - $mzToMatch" | bc | tr -dc '0-9')

    if [[ "$PerfectMatch" == 0 ]];then
    countmzMatch=$(( $countmzMatch + 1 ))
    sed -n ${linenumber_1}p
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawCompoundCon
dition}_trimmed.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt
    fi

    linenumber_1=$(( $linenumber_1 + 1 ))

done

echo "
Found $countmzMatch initial match/es for mz: $mzToMatch
"

#resets line number after the above mini-loop
linenumber_1=$Condition1LineNumberStart

if [[ ! -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt ]];then
touch
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt
fi

mzmatched=$(grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt)

if [[ "$mzmatched" > 0 ]];then
echo "Checking retention time (rt) of the mz match"
#set rt to match variable to the given line number until the condition is complete

```

```

rtToMatch=$(awk -v x=$linenumber '{FS=" "; FNR == x {print $1}'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt)

echo "Retention time to match to = $rtToMatch"
#1 decimal lower than match
rtminHigher=$(echo "scale=2; $rtToMatch -$rtVar" | bc)

if [[ $(echo "$rtminHigher < 0" | bc) -eq 1 ]];then
rtminHigher=0
fi

echo "Retention time lower limit: $rtminHigher"

#1 decimal higher than match
rtminLower=$(echo "scale=2; $rtToMatch +$rtVar" | bc)

echo "Retention time upper limit: $rtminLower"

#find matches within user provided range of the rt to match

awk -F '$\t' -v h=$rtminHigher -v l=$rtminLower 'BEGIN { OFS = FS } $3 > h && $3 < l { print }'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt

#cut to closest mz value by first taking the numbers after the decimal place and picking the ones with
the closest match to original mz decimals

#number of matches
NumberMzRtMatches=$( grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt )
echo "
Initial filtered number of mz matches found for compound $CompoundNumber1MatchCheck from
$MetabInputData $RunningFiletype ion file in $MetabRawPosInputData $RunningFiletype ion file =
$NumberMzRtMatches
"

else
if [[ ! -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt ]];then
touch
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz
Match.txt
fi
if [[ ! -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt ]];then
touch
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt
fi
NumberMzRtMatches=0

```

fi

```
#if number of matches > 1 then cut to closest to exact mz
if [[ $NumberMzRtMatches > 1 ]] && [[ $NumberMzRtMatches != 0 ]];then
    #set mz to match as just the full number after the decimal point
    mzToMatch=$(awk -v x=$linenumber 'NR==x'
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt | cut -f2 -d ".")
    echo "
As more than one match passed the first mz and rt filter, the program is now looking for closest match
to the post decimal place numbers of the mz. The post decimal place numbers for compound
$CompoundNumber1MatchCheck are: $mzToMatch
"

    #create a file containing the numbers after the decimal place for possible matches thus far
    awk -F $'\t' 'BEGIN { OFS = FS } { print $2 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt | cut -f2 -d "." >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp1.txt

    awk -v n=$mzToMatch '
function abs(x) {return x < 0 ? -x : x}
{print abs($0 - n) "\t" $0}' <
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp1.txt | sort -n | head -n 1 >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp2.txt

    #above creates a odd new column, so this removes that
    awk -F $'\t' 'BEGIN { OFS = FS } { print $2 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp2.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp3.txt

    #set above file first option as variable
    ExactDecMatch=$(head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp3.txt)
    echo "
The best match for compound $CompoundNumber1MatchCheck is: $ExactDecMatch
"

    #re-grabs the full line for the one that was an exact match
    grep "$ExactDecMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch.txt

    #same as above but only stores the current loops output - this is for creating the compound number
ID adjustment at the end of each loop
    grep "$ExactDecMatch"
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumFix.txt
```

```

#where there was instantly only one match it puts it straight into the final collected file of repeat
compounds
else
  if [[ $NumberMzRtMatches != 0 ]];then
    echo "No further filtering required, single match found"
    head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt >>
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch.txt
  #same as above but only stores the current loops output - this is for creating the compound number
ID adjustment at the end of each loop
  head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt
  else
    echo "No match found"
    touch
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt
  fi
fi
#####
#####

### looks in the copy of the pos raw original file and goes to the $RepeatCompoundNumber in
column 1 and changes it to the compound at the line number

NumberMzRtMatches=$( grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt )

if [[ $NumberMzRtMatches = 1 ]];then
#sets the single match as variable
RepeatCompoundInAlternativeCondition=$(head -n1
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt)

echo "Matching compounds full row data: $RepeatCompoundInAlternativeCondition"

#sets the found matches incorrect compound number as a variable
RepeatCompoundNumber=$(awk -F $'\t' 'BEGIN { OFS = FS } { print $1 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt)

echo "Matching compounds number is: $RepeatCompoundNumber"
#this will make the repeat of a compound change to the same arbitrary number it was associated to first
#at line $RepeatLineNumber replace column 1 $RepeatLineNumber with $linenumber in
~/CCRACD/MetabolomicProgramOutput/MetabData_2

```

```
#runs thru each one adding up, if variable is greater than zero at the end then dont do the below if,
while running through them, if variable increases make list of conditions that contain it
ConditionToCheckIfCompoundNameAlreadyChanged=1
```

```
HasItAlreadyBeenChangedPrevious=0
```

```
for i in $(seq $NumberOfConditions);do
```

```
#grabs condition name
```

```
NameOfConditionToCheckIfCompoundNameAlreadyChanged=$(awk -v
x=$ConditionToCheckIfCompoundNameAlreadyChanged 'NR==x'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt | tr -d " \t")
```

```
#gets the metabolite names from condition of the compounds of interest file
```

```
CompoundNumbersInConditionSearchingFrom=$(awk -F $'\t' 'BEGIN { OFS = FS } { print $1 }'
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${NameOfConditionToCh
eckIfCompoundNameAlreadyChanged}.txt)
```

```
#counts if the compound name is already from condition i.e. has it already been changed
```

```
HasItAlreadyBeenChanged=$( echo "$CompoundNumbersInConditionSearchingFrom" | grep -o
"$RepeatCompoundNumber" | wc -l )
```

```
##### this creates a file saying which condition had already changed that compound name and
if it was more than once (which it shouldt have) # # might not need/use this bit, but built it just incase
if [[ "$HasItAlreadyBeenChanged" > 0 ]];then
```

```
echo
```

```
"${NameOfConditionToCheckIfCompoundNameAlreadyChanged},${HasItAlreadyBeenChanged}"
```

```
>>
```

```
~/CCRACD/MetabolomicProgramOutput/CompoundAndNumberOfChangesForACompoundSearchF
rom_Compound${CompoundOfInterestCondition}.txt
```

```
fi
```

```
if [[ "$HasItAlreadyBeenChanged" > "$HasItAlreadyBeenChangedPrevious" ]];then
```

```
NumberOfChanges=$(echo "$HasItAlreadyBeenChanged - $HasItAlreadyBeenChangedPrevious" |
bc)
```

```
echo "${NameOfConditionToCheckIfCompoundNameAlreadyChanged},${NumberOfChanges}" >>
```

```
~/CCRACD/MetabolomicProgramOutput/CompoundAndNumberOfChangesForACompoundSearchF
rom_Compound${CompoundOfInterestCondition}.txt
```

```
fi
```

```
#####
```

```
HasItAlreadyBeenChangedPrevious=$HasItAlreadyBeenChanged
```

```
ConditionToCheckIfCompoundNameAlreadyChanged=$((
```

```
$ConditionToCheckIfCompoundNameAlreadyChanged + 1 ))
```

```
done
```

```
#####
```

```
#checks if the compound has already been changed before changing it now (does this by checking
conditions from the selected file and making sure the number thats about to be changed isn't one of
the selected file numbers)
```

```
if [[ "$HasItAlreadyBeenChanged" = 0 ]];then
```

```
#sets the found matches incorrect compound number as a variable
```

```
NumberOfTheCompoundMatchingTo=$(awk -v l=$linenumber -F $'\t' 'BEGIN { OFS = FS } NR==1
{print $1}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")
```

```
echo "Original compounds number: $NumberOfTheCompoundMatchingTo"
```

```

#make the actual line number of the found match a variable
RepeatLineNumber=$(grep -n "$RepeatCompoundInAlternativeCondition"
~/CCRACD/MetabolomicProgram_Input/"$MetabRawPosInputData" | cut -f1 -d ":")

echo "The match was found on line number '${RepeatLineNumber}' in the $MetabRawPosInputData
$RunningFiletype ion file"

##### This is the file the table making script uses #####

awk -v a=$RepeatLineNumber -v b=$NumberOfTheCompoundMatchingTo -F '\t' 'BEGIN { OFS =
FS } NR==a{$1=b}1' ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData" > tmp
&& mv tmp ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData"

echo "Matching compound number correctly adjusted to: $NumberOfTheCompoundMatchingTo "
#RepeatLineNumber
else echo "This compound number has been previously adjusted/corrected"
fi

fi

#set a variable to let the user know what compounds were searched for
SearchForCompound=$(awk -v x=$linenumber -F '\t' 'BEGIN { OFS = FS } NR==x {print $1}'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

SuccessfulMatch=$( grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt )

if [[ $SuccessfulMatch = 1 ]];then
echo "
Completed the search for compound $CompoundNumber1MatchCheck in the
${RawCompoundCondition} condition of the $MetabRawPosInputData $RunningFiletype ion file.
Best match for compound $CompoundNumber1MatchCheck in the ${RawCompoundCondition}
condition of the $MetabRawPosInputData $RunningFiletype ion file is:
$RepeatCompoundInAlternativeCondition
"
echo "
Completed the search for compound $CompoundNumber1MatchCheck in the
${RawCompoundCondition} condition of the $MetabRawPosInputData $RunningFiletype ion file.
Best match for compound $CompoundNumber1MatchCheck in the ${RawCompoundCondition}
condition of the $MetabRawPosInputData $RunningFiletype ion file is:
$RepeatCompoundInAlternativeCondition
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt

else
echo "
Completed the search for compound $CompoundNumber1MatchCheck in the
${RawCompoundCondition} condition of the $MetabRawPosInputData $RunningFiletype ion file.
No match found
"
echo "
Completed the search for compound $CompoundNumber1MatchCheck in the
${RawCompoundCondition} condition of the $MetabRawPosInputData $RunningFiletype ion file.
No match found

```

```

" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
fi
echo "
Compound $SearchForCompound check complete

"
NumberTimesThrough=$(( $NumberTimesThrough + 1 ))
linenumber=$(( $linenumber + 1 ))
PreviousLineNumber=$linenumber

if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
Match.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
Match.txt
fi
if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp3.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp3.txt
fi
if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp2.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp2.txt
fi
if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp1.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_tmp1.txt
fi
if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_
${RawCompoundCondition}CloseMz_RtMatch_tmp.txt
fi
if [ -f
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt ]
then rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${RawCompoundCondition}CloseMz_
RtMatch_NumbFix.txt
fi

else

```

```
echo "  
Compound $CompoundNumber1MatchCheck in condition ${CompoundOfInterestCondition} has  
already been found in condition ${RawCompoundCondition} within the original selected metabolite  
dataset. Therefore it will not be searched for within the raw $RunningFiletype ion file  
"
```

```
echo "  
Compound $CompoundNumber1MatchCheck in condition ${CompoundOfInterestCondition} was  
already found in condition ${RawCompoundCondition} within the original selected metabolite  
dataset. Therefore it was not searched for within the raw $RunningFiletype ion file  
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
NumberTimesThrough=$(( $NumberTimesThrough + 1 ))  
linenumber=$(( $linenumber + 1 ))  
PreviousLineNumber=$linenumber  
fi  
done
```

```
if [ -f  
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_  
${RawCompoundCondition}CloseMz_RtMatch.txt ];then  
NumberInCondition=$(grep -c -v "HelloThisIsMyAntiMatch"  
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon${CompoundOfInterestCondition}_in_  
${RawCompoundCondition}CloseMz_RtMatch.txt)  
else  
NumberInCondition=0  
echo "Note zero matches found may mean compounds were already identified within the compounds  
of interest file"  
fi
```

```
NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition=$(awk -v  
x=${CompoundOfInterestConditionNumber} 'NR==x'  
~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt)
```

```
echo "
```

The search for Identical compounds from \${CompoundOfInterestCondition} condition in
\${RawCompoundCondition} condition is complete, there are \$NumberInCondition compound
matches found of the \$NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition searched
for in the raw file. Those found have had their compound identities changed correctly.

```
#####  
#####  
"
```

```
echo "  
The search for Identical compounds from ${CompoundOfInterestCondition} condition in  
${RawCompoundCondition} condition was complete, there are $NumberInCondition compound  
matches found of the $NumberOfCompoundsInCompoundOfInterestFile_InGivenCondition searched  
for. Those found have had their compound identities changed correctly.
```

```
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
```

```
#####
```

```

#Refresh the comparison files with the most up-to-date and changed file
if [ -f ~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData" ];then
RawConditionLine=1
  for i in $(seq 1 $NumberOfRawConditions);do
    RawCondition=$(echo "$RawConditionNames" | awk -v x=$RawConditionLine 'NR==x {print}'
| tr -d "\t")
    awk -v x=$RawCondition '$4==x {print}'
~/CCRACD/MetabolomicProgramOutput/1_"$MetabRawPosInputData" >
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${RawConditionLine}
.txt
    RawConditionLine=$(( $RawConditionLine + 1 ))
  done
fi

```

```
#####
```

```
RawCompoundConditionNumber=$(( $RawCompoundConditionNumber - 1 ))
```

```

#changes the compound of interest condition
if [[ "$LoopNumber" =
"$NumberOfRawConditionsToCheckB4ChangingCompoundInterestCondition" ]];then
  LoopNumber=0
  #resets the raw condition
  RawCompoundConditionNumber=$NumberOfRawConditions
  #resets the compounds of interest condition
  CompoundOfInterestConditionNumber=$(( $CompoundOfInterestConditionNumber + 1 ))
fi
LoopNumber=$(( $LoopNumber + 1 ))

```

```

done
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
fi

```

```

SelectConditionNameForFileCleanUp=1
  for i in $(seq $NumberOfConditions);do
    #grabs condition name
    ConditionNameForFileCleanUp=$(awk -v x=$SelectConditionNameForFileCleanUp 'NR==x'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${ConditionNameForFileC
leanUp}.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${ConditionNameFor
FileCleanUp}.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${ConditionNameForFileCleanU
p}mzdec1.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${ConditionNameForFileCleanU
p}mzdec.txt
    SelectConditionNameForFileCleanUp=$(( $SelectConditionNameForFileCleanUp + 1 ))
  done
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_RawEachConditionNo.txt

WasANegFileSelected=$(( $WasANegFileSelected + 3 ))
WasAPosFileSelected=$(( $WasAPosFileSelected + 1 ))
done
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_rt.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_mzdec.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabData_tmp_EachConditionNo.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval1.txt
fi
if [ -f ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt ];then
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataConditionRemoval.txt
fi

```

```

SelectConditionNameForFileCleanUp=1
  for i in $(seq $NumberOfConditions);do
    #grabs condition name
    ConditionNameForFileCleanUp=$(awk -v x=$SelectConditionNameForFileCleanUp 'NR==x'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt | tr -d " \t")
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataCompareAgainstCon${ConditionNameForFileC
leanUp}.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRawCompareAgainstCon${ConditionNameFor
FileCleanUp}.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${ConditionNameForFileCleanU
p}mzdec1.txt
    rm
~/CCRACD/MetabolomicProgramOutput/MetabDataRaw_tmp_Con${ConditionNameForFileCleanU
p}mzdec.txt
    SelectConditionNameForFileCleanUp=$(( $SelectConditionNameForFileCleanUp + 1 ))
  done

echo "

```

Giles Holt Metabolomics Program has COMPLETE the following process- cross comparison for identifying identical compounds accross conditions"

```

rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon2_vs_3CloseMz_RtMatch.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon2CloseMz_RtMatch.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon3CloseMz_RtMatch.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon3_vs_1CloseMz_RtMatch.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon2_vs_1CloseMz_RtMatch.txt
rm ~/CCRACD/MetabolomicProgramOutput/MetabDataRawCon3_vs_2CloseMz_RtMatch.txt

echo | ~/CCRACD/Scripts/MetabTableMakingScript.sh

```

10.8.3.4 Compounds of interest through conditions: Table creation

```
#!/bin/bash
```

```
echo "
```

```
    Building a table from the data...
```

```
    Building a table from the data...
```

```
    Building a table from the data...
```

```
"
```

```
#Grab the file names and locations required for table construction
```

```
#takes the name of the csv file selected
```

```
MetabInputData=$(awk -F "|" '{print $1}'
```

```
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)
```

```
#takes the name of the neg ion csv file selected
```

```
MetabInputData_NegIon=$(awk -F "|" '{print $2}'
```

```
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)
```

```
MetabRawPosInputData=1_$(awk 'BEGIN {FS = "|"} {print $4}'
```

```
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)
```

```
MetabRawNegInputData=1_$(awk 'BEGIN {FS = "|"} {print $3}'
```

```
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)
```

```
#creates a variable with a number in if no file was selected
```

```
WasAPosFileSelected=$(echo $MetabInputData | grep -c -x "-")
```

```
#creates a variable with a number in if no file was selected
```

```
WasANegFileSelected=$(echo $MetabInputData_NegIon | grep -c -x "-")
```

```
if [[ $WasAPosFileSelected = 0 ]];then
```

```
echo "
```

```
A positive ion input file was selected"
```

```
fi
```

```
if [[ $WasANegFileSelected = 0 ]];then
```

```
echo "
```

```

A Negative ion input file was selected"
fi

NumberOfFilesToRun=$(( $WasAPosFileSelected + $WasANegFileSelected ))

#if no files were selected it would = 2
if [[ $NumberOfFilesToRun > 1 ]]
then
notify-send "No files selected to run"
echo "No files selected to run"
echo "No files selected to run" >> ~/CCRACD/tmp/InputConditionRepeatStats.txt
else

#loop to run positive ion file if there and negative ion file if there

for i in $(seq 1 2);do

if [[ $WasAPosFileSelected = 0 ]];then

echo "

#####                               #####

                Running the Positive ion file

#####                               #####

"

#####

# below - selecting abundance type

#####
#select the raw or norm data depending on settings if 'not applicable' is chosen then run all that data
AbundanceChoiceType=$(awk 'BEGIN {FS = "|"} {print $7}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

if [[ "$AbundanceChoiceType" == "-" ]];then
echo "No Abundance type given, constructing table as appropriate"

else

echo "Abundance type setting is: ${AbundanceChoiceType}. Preparing for table based solely on
$AbundanceChoiceType data, trimming $MetabInputData to the selected abundance"

cp ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/Copy_"$MetabInputData"

MetabInputData="Copy_"$MetabInputData"

```

```

#find the column $AbundanceChoiceType was on
AbundanceChoiceTypeColumnNumber=$(awk -v A="$AbundanceChoiceType" -F $'\t' 'BEGIN {
OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i } }'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

#if that phrase isnt on column 3 delete the columns between 2 and the column number it is found on
if [[ "$AbundanceChoiceTypeColumnNumber" != 5 ]];then

    echo "abundance type chosen is not the first abundance type in the file"

    cut -f 1,2,3,4,"$AbundanceChoiceTypeColumnNumber"-
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"

    mv ~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

#else it is column 3, so this works out what the abundance that hasn't been selected is called, so it can
be removed
else

    echo "abundance type chosen is the first abundance type in the file"

    if [[ "$AbundanceChoiceType" == "Normalised abundance" ]];then
        NotTheAbundanceChoiceType="Raw abundance"
    else
        NotTheAbundanceChoiceType="Normalised abundance"
    fi

#look for string $NotTheAbundanceChoiceType, if not found then no further columns need removing.
if found delete columns from and including $AbundanceChoiceTypeColumnNumber

    AbundanceChoiceTypeColumnNumber=$(awk -v A="$NotTheAbundanceChoiceType" -F
$'\t' 'BEGIN { OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i } }'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

#this avoids the no number problem if there is only one abundance name in the file and they select
that name
    AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber + 1 ))

    if [[ "$AbundanceChoiceTypeColumnNumber" == 1 ]]

        then
            echo "$AbundanceChoiceType data was selected and is the only type of abundance present,
running now "

        else

#remove 2 rather than 1 to make up for the additional 1 added earlier that ensured it would be a
number regardless
        AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber - 2 ))
        echo "$AbundanceChoiceTypeColumnNumber"

```

```

        cut -f 1-"$AbundanceChoiceTypeColumnNumber"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"

        mv ~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

        echo "$NotTheAbundanceChoiceType that was not selected has been found and removed"

fi

        fi

                fi

#####selecting abundance type from the Raw file

#select the raw or norm data depending on settings if 'not applicable' is chosen then run all that data
AbundanceChoiceType=$(awk 'BEGIN {FS = "|"} {print $7}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

if [[ "$AbundanceChoiceType" == "-" ]];then
echo "No Abundance type given, constructing table as appropriate"

else

echo "Abundance type setting is: $AbundanceChoiceType . Preparing for table based solely on
$AbundanceChoiceType data, trimming $MetabRawPosInputData to the selected abundance"

cp ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"
~/CCRACD/MetabolomicProgramOutput/Copy_"$MetabRawPosInputData"
MetabRawPosInputData="Copy_"$MetabRawPosInputData"

#find the column $AbundanceChoiceType was on
AbundanceChoiceTypeColumnNumber=$(awk -v A="$AbundanceChoiceType" -F $'\t' 'BEGIN {
OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData")

#if that phrase isnt on column 3 delete the columns between 2 and the column number it is found on
if [[ "$AbundanceChoiceTypeColumnNumber" != 5 ]];then

        echo "abundance type chosen is not the first abundance type in the file"

                cut -f 1,2,3,4,"$AbundanceChoiceTypeColumnNumber"-
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" >
~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"

                        mv ~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"

#else it is column 3, so this works out what the abundance that hasn't been selected is called, so it can
be removed
        else

```

```

echo "abundance type chosen is the first abundance type in the file"

if [[ "$AbundanceChoiceType" == "Normalised abundance" ]]
then
NotTheAbundanceChoiceType="Raw abundance"
else
NotTheAbundanceChoiceType="Normalised abundance"
fi

#look for string $NotTheAbundanceChoiceType, if not found then no further columns need removing.
if found delete columns from and including $AbundanceChoiceTypeColumnNumber

AbundanceChoiceTypeColumnNumber=$(awk -v A="$NotTheAbundanceChoiceType" -F
$\t 'BEGIN { OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData")

#this avoids the no number problem if there is only one abundance name in the file and they select
that name
AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber + 1 ))

if [[ "$AbundanceChoiceTypeColumnNumber" == 1 ]];then
echo "$AbundanceChoiceType data was selected and is the only type of abundance present,
running now "

else

#remove 2 rather than 1 to make up for the additional 1 you added earlier that ensured it would be a
number regardless
AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber - 2 ))
echo "$AbundanceChoiceTypeColumnNumber"
cut -f 1-"$AbundanceChoiceTypeColumnNumber"
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" >
~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"

mv ~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"

echo "$NotTheAbundanceChoiceType that was not selected has been found and removed"

fi

fi

fi

#####

# above - selecting abundance type

#####

IonType=P_

```

```

fi

if [[ $WasANegFileSelected = 3 ]];then

echo "

#####                               #####

                Running the negative ion file

#####                               #####

"

#Change WasAPosFileSelected to equal WasANegFileSelected
WasAPosFileSelected=$WasANegFileSelected

#Change WasAPosFileSelected to equal WasANegFileSelected
MetabInputData=$MetabInputData_NegIon

#Change MetabRawPosInputData to equal MetabRawNegInputData - removed the 1_ from here and
added earlier in
MetabRawPosInputData=$MetabRawNegInputData

IonType=N_

#####

# below - selecting abundance type

#####
#select the raw or norm data depending on settings if they choose not applicable then run all that data
as good to go
AbundanceChoiceType=$(awk 'BEGIN {FS = "|"} {print $7}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

if [[ "$AbundanceChoiceType" == "-" ]];then
echo "No Abundance type given, constructing table as appropriate"

else

echo "Abundance type setting is: $AbundanceChoiceType . Preparing for table based solely on
$AbundanceChoiceType data, trimming $MetabInputData to the selected abundance"

cp ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/Copy_"$MetabInputData"

MetabInputData="Copy_"$MetabInputData"

#find the column $AbundanceChoiceType was on
AbundanceChoiceTypeColumnNumber=$(awk -v A="$AbundanceChoiceType" -F '\t' 'BEGIN {
OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

```

```

#if that phrase isnt on column 3 delete the columns between 2 and the column number it is found on
if [[ "$AbundanceChoiceTypeColumnNumber" != 5 ]];then

    echo "abundance type chosen is not the first abundance type in the file"

    cut -f 1,2,3,4,"$AbundanceChoiceTypeColumnNumber"-
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"

    mv ~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

#else it is column 3, so this works out what the abundance that hasn't been selected is called, so it can
be removed
else

    echo "abundance type chosen is the first abundance type in the file"

    if [[ "$AbundanceChoiceType" == "Normalised abundance" ]]
    then
        NotTheAbundanceChoiceType="Raw abundance"
    else
        NotTheAbundanceChoiceType="Normalised abundance"
    fi

#look for string $NotTheAbundanceChoiceType, if not found then no further columns need removing.
if found delete columns from and including $AbundanceChoiceTypeColumnNumber

    AbundanceChoiceTypeColumnNumber=$(awk -v A="$NotTheAbundanceChoiceType" -F
    $\t 'BEGIN { OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

#this avoids the no number problem if there is only one abundance name in the file and they select
that name
    AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber + 1 ))

    if [[ "$AbundanceChoiceTypeColumnNumber" == 1 ]]

    then
        echo "$AbundanceChoiceType data was selected and is the only type of abundance present,
running now "

    else

#remove 2 rather than 1 to make up for the additional 1 you added earlier that ensured it would be a
number regardless
        AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber - 2 ))
        echo "$AbundanceChoiceTypeColumnNumber"
        cut -f 1-"$AbundanceChoiceTypeColumnNumber"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"

```

```

mv ~/CCRACD/MetabolomicProgram_Input/tmp"$MetabInputData"
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData"

echo "$NotTheAbundanceChoiceType that was not selected has been found and removed"

fi

fi

fi

#####selecting abundance type from the Raw file

#select the raw or norm data depending on settings if 'not applicable' is chosen then run all that data
AbundanceChoiceType=$(awk 'BEGIN {FS = "|"} {print $7}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

if [[ "$AbundanceChoiceType" == "-" ]];then
echo "No Abundance type given, constructing table as appropriate"

else

echo "Abundance type setting is: $AbundanceChoiceType . Preparing for table based solely on
$AbundanceChoiceType data, trimming $MetabRawPosInputData to the selected abundance"

cp ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"
~/CCRACD/MetabolomicProgramOutput/Copy_"$MetabRawPosInputData"

MetabRawPosInputData="Copy_"$MetabRawPosInputData"

#find the column $AbundanceChoiceType was on
AbundanceChoiceTypeColumnNumber=$(awk -v A="$AbundanceChoiceType" -F $'\t' 'BEGIN {
OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData")

#if that phrase isnt on column 3 delete the columns between 2 and the column number it is found on
if [[ "$AbundanceChoiceTypeColumnNumber" != 5 ]];then

echo "abundance type chosen is not the first abundance type in the file"

cut -f 1,2,3,4,"$AbundanceChoiceTypeColumnNumber"-
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" >
~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"

mv ~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"

#else it is column 3, so this works out what the abundance that hasn't been selected is called, so it can
be removed
else

echo "abundance type chosen is the first abundance type in the file"

```

```

if [[ "$AbundanceChoiceType" == "Normalised abundance" ]];then
    NotTheAbundanceChoiceType="Raw abundance"
else
    NotTheAbundanceChoiceType="Normalised abundance"
fi

#look for string $NotTheAbundanceChoiceType, if not found then no further columns need removing.
if found delete columns from and including $AbundanceChoiceTypeColumnNumber

    AbundanceChoiceTypeColumnNumber=$(awk -v A="$NotTheAbundanceChoiceType" -F
    $\t 'BEGIN { OFS = FS }; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
    ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData")

#this avoids the no number problem if there is only one abundance name in the file and they select
that name
    AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber + 1 ))

    if [[ "$AbundanceChoiceTypeColumnNumber" == 1 ]]

        then
            echo "$AbundanceChoiceType data was selected and is the only type of abundance present,
            running now "
        else

#remove 2 rather than 1 to make up for the additional 1 you added earlier that ensured it would be a
number regardless
        AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber - 2 ))
        echo "$AbundanceChoiceTypeColumnNumber"
        cut -f 1-"$AbundanceChoiceTypeColumnNumber"
        ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" >
        ~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"

        mv ~/CCRACD/MetabolomicProgramOutput/tmp"$MetabRawPosInputData"
        ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData"

        echo "$NotTheAbundanceChoiceType that was not selected has been found and removed"

    fi

fi

fi

fi

fi

MetabInputData_Name=$(echo "$MetabInputData" | awk -F '.' '{ print $1 }')

MetabInputData_NegIon_Name=$(echo "$MetabInputData_NegIon" | awk -F '.' '{ print $1 }')

StopCheckAsFoundAnswer=2
Find1stCompound1stRow=1

```

```

for i in $(seq 1 15);do
  if [[ "$StopCheckAsFoundAnswer" = "2" ]];then
    Find1stCompound2ndRow=$(( $Find1stCompound1stRow + 1 ))
    SelectedCompounds_CheckRows1="$(awk -F '$\t' -v x=$Find1stCompound1stRow 'BEGIN {
OFS = FS } ; NR==x {print $4}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")"
    SelectedCompounds_CheckRows2="$(awk -F '$\t' -v x=$Find1stCompound2ndRow 'BEGIN
{ OFS = FS } ; NR==x {print $4}' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")"
    if [[ "$SelectedCompounds_CheckRows1" == "$SelectedCompounds_CheckRows2" ]] && [[
! -z "$SelectedCompounds_CheckRows1" ]];then
      SelectedCompounds_Condition1LineNumberStart="$Find1stCompound1stRow"
      echo "
The first condition starts on line: $SelectedCompounds_Condition1LineNumberStart
"
      echo "
The first condition started on line: $SelectedCompounds_Condition1LineNumberStart
" >> ~/CCRACD/tmp/CompoundThruConditionStats.txt
      StopCheckAsFoundAnswer=1
    fi
    Find1stCompound1stRow=$(( $Find1stCompound1stRow + 1 ))
  fi
done

TitlesLineEnd=$(( $SelectedCompounds_Condition1LineNumberStart - 1 ))

      echo "
The titles precluding data are from line 1 to line $TitlesLineEnd
"

if [[ $WasAPosFileSelected = 0 ]];then

### changed to _Input
  #Step 2 takes the lines that prelude the data and makes the beginning of the table text file
  awk -v x=$TitlesLineEnd -F '$\t' 'BEGIN { OFS = FS } ; NR==1,NR==x { print }'
  ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
  ~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv
fi

# an if for making the table if the positive file wasn't selected and the table file therefore doesn't exist
if [[ $WasAPosFileSelected != 0 ]] && [[ $WasANegFileSelected = 3 ]];then
  if [[ -f
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv ]];then
    ##### ONLY USE THE HASHED BELOW BIT IF YOU WISH TO HAVE 2 SEPERATE
    TABLES FOR POS AND NEG ION #####
    #Step 2 takes the lines that prelude the data and makes the beginning of the table text file
    #awk -v x=$TitlesLineEnd -F '$\t' 'B#EGIN { OFS = FS } ; NR==1,NR==x { print }'
    ~/CCRACD/#MetabolomicProgramOutput/"$MetabInputData" >>
    ~/CCRACD/#MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_N
ame"_Table.csv
    echo ""
  else
    ##changed to _Input
    #Step 2 takes the lines that prelude the data and makes the beginning of the table text file

```

```

    awk -v x=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS } ; NR==1,NR==x { print }'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv

```

```

fi
fi

```

#####now start adding the compound info:

```

#find compound 1 in positive file - take each line that matches - take the AVERAGE column from
each sub condition changing for norm or raw depending on what was selected

```

```

awk -v x=$SelectedCompounds_Condition1LineNumberStart -F $'\t' 'BEGIN { OFS = FS } ; NR>=x
{ print $1 }' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | sort | uniq | awk 'NF' >
~/CCRACD/MetabolomicProgramOutput/UniqueCompoundList.csv
## check the unique compound list, as sometimes it seems to be including the compound title
'compound ID'

```

```

NumberOfCompounds=$(grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/UniqueCompoundList.csv)

```

```

CompoundLineNumber=1

```

```

echo "Number of compounds to gather from varing conditions to create the metabolite table:
$NumberOfCompounds"

```

```

#Adding in the compounds from sig input and raw input
for i in $(seq 1 $NumberOfCompounds);do

```

```

#sets the compound name variable - changed to _Input
CompoundName=$(awk -v x=$CompoundLineNumber -F $'\t' 'BEGIN { OFS = FS } ; NR==x { print
$1 }' ~/CCRACD/MetabolomicProgramOutput/UniqueCompoundList.csv)

```

```

echo "
Compound name: $CompoundName , line number of compound $CompoundName was found on:
$CompoundLineNumber"

```

```

#Checks that this particular compound in that specific condition hasn't already been added by
checking if there is already an exact match for that name
CompoundLineAboutToBeAdded=$(awk -v x=$CompoundName -F $'\t' 'BE#GIN { OFS = FS } ; $1
== x' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | sed "s/^/$IonType/")

```

```

#added a second one to cover if theres a name without the _
CompoundLineAboutToBeAdded2=$(awk -v x=$CompoundName -F $'\t' 'B#EGIN { OFS = FS } ;
$1 == x' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData")

```

```

#gives this variable a number greater than 0 if the exact CompoundLineAboutToBeAdded is already
present in the table
PrePresenceOfCompoundLineAboutToBeAdded=$(grep -c "$CompoundLineAboutToBeAdded"
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv)

```

```

#added a second one to cover if theres a name without the _

```

```
PrePresenceOfCompoundLineAboutToBeAdded2=$(grep -c "$CompoundLineAboutToBeAdded2"
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv)
```

```
#makes sure no duplicate lines are added to the table
if [[ $PrePresenceOfCompoundLineAboutToBeAdded > 0 ]] || [[
$PrePresenceOfCompoundLineAboutToBeAdded2 > 0 ]]
then
echo ""
else
```

```
#outputs all the compounds from inputSig that match the compound name into the table
awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS = FS } ; $1 == x'
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | sed "s/^\$IonType/" >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv
```

```
#kept as output because the most uptodate version of the file is being refreshed constantly in the
output
```

```
#outputs all the compounds from inputRaw that match the compound name into the table
awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS = FS } ; $1 == x'
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" | sed "s/^\$IonType/" >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv
```

```
fi
```

```
#Checks that this particular compound in that specific condition hasn't already been added by
checking if there is already an exact match for that name
```

```
CompoundLineAboutToBeAdded=$(awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS = FS } ; $1
== x' ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" | sed "s/^\$IonType/")
```

```
echo "$CompoundLineAboutToBeAdded" >
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv
```

```
NumberOfMatchesFromRaw=$(awk -F $'\t' '{print $1}'
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv | grep -c "[0-9]")
```

```
ConditionMatchesFrom=$(awk -F $'\t' '{print $4}'
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv)
```

```
echo "Compound: $CompoundName : Found in raw: $NumberOfMatchesFromRaw : Matches from
conditions: $ConditionMatchesFrom"
```

```
#CompoundLineFromFileToBeAddedTo=$(awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS =
FS } ; $1 == x'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv | sed "s/^\$IonType/")
```

```
#echo "$CompoundLineFromFileToBeAddedTo" >
~/CCRACD/MetabolomicProgramOutput/CompoundLineFromFileToBeAddedTo.csv
```

```
#compare the two - output any line that can be found in raw but not in table
```

```

grep -F -x -v -f ~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv
~/CCRACD/MetabolomicProgramOutput/CompoundLineFromFileToBeAddedTo.csv >
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.txt

#number of lines in the lines to be added file
NumberOfLinesToAdd=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.txt)

if [[ $NumberOfLinesToAdd > 0 ]];then
LineNumberToAdd=1
find this line in raw and put into table
    #loop to number of lines in the lines to be added file
    for i in $(seq 1 $NumberOfLinesToAdd);do
    awk -v x=$LineNumberToAdd -F '$\t' 'BEGIN { OFS = FS } ; NR==x'
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv

    LineNumberToAdd=$(( $LineNumberToAdd + 1 ))
done
fi

#Checks that this particular compound in that specific condition hasn't already been added by
checking if there is already an exact match for that name

CompoundLineAboutToBeAdded=$(awk -v x=$CompoundName -F '$\t' 'BEGIN { OFS = FS } ; $1
== x' ~/CCRACD/MetabolomicProgram_Input/"$MetabInputData" | sed "s/^\$IonType/")

echo "$CompoundLineAboutToBeAdded" >>
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv

CompoundLineFromFileToBeAddedTo=$(awk -v x=$CompoundName -F '$\t' 'BEGIN { OFS = FS
} ; $1 == x'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv | sed "s/^\$IonType/")

echo "$CompoundLineFromFileToBeAddedTo" >
~/CCRACD/MetabolomicProgramOutput/CompoundLineFromFileToBeAddedTo.csv

#compare the two - output any line that can be found in sig input but not in table

grep -F -x -v -f ~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv
~/CCRACD/MetabolomicProgramOutput/CompoundLineFromFileToBeAddedTo.csv >
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.txt

awk -F '$\t' 'BEGIN { OFS = FS } ; !seen[$0]++'
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv >
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.txt

##### at this point it has any compound matches from both sig and raw in 1 file

#sort the lines in the file and remove any duplicate lines
sort ~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.csv | uniq | awk -F
'\t' 'BEGIN { OFS = FS } ; NF' >
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv

```

```

#remove lines in the file that are identical to lines in the table file
awk -F $'\t' 'BEGIN { OFS = FS } ; NR==FNR{a[$0];next} !($0 in a)'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table.csv ~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv >
~/CCRACD/tmp/tmpLinesToBeAddedIntoTable.csv

mv ~/CCRACD/tmp/tmpLinesToBeAddedIntoTable.csv
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv

#number of lines in the lines to be added file
NumberOfLinesToAdd=$(grep -c '[0-9]'
~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv)

if [[ "$NumberOfLinesToAdd" > 0 ]];then
cat ~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table.csv
fi

#NumberOfMatchesForCompoundName=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded.txt)

#CompoundLineAboutToBeAdded2=$(awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS = FS } ;
$1 == x' ~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData")

#echo "$CompoundLineAboutToBeAdded2" >
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded2.txt

#NumberOfMatchesForCompoundName2=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/CompoundLineAboutToBeAdded2.txt)

#gives this variable a number greater than 0 if the exact CompoundLineAboutToBeAdded is already
present in the table

if [[ "$NumberOfMatchesForCompoundName" == 1 ]];then
PrePresenceOfCompoundLineAboutToBeAdded=$(grep -c "$CompoundLineAboutToBeAdded"
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table.csv)

PrePresenceOfCompoundLineAboutToBeAdded2=$(grep -c "$CompoundLineAboutToBeAdded2"
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table.csv)

if [[ $PrePresenceOfCompoundLineAboutToBeAdded > 0 ]] || [[
$PrePresenceOfCompoundLineAboutToBeAdded2 > 0 ]]
then
echo ""
else
#kept as output because the most uptodate version of the file is being refreshed constantly in the
output
#outputs all the compounds from inputRaw that match the compound name into the table
awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS = FS } ; $1 == x'
~/CCRACD/MetabolomicProgramOutput/"$MetabRawPosInputData" | sed "s/^\$IonType/" >>

```

```

~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table.csv

fi

CompoundLineNumber=$(( $CompoundLineNumber + 1 ))

done

#negative
if [[ $WasAPosFileSelected = 0 ]];then

#makes it a generic name so that it can be combined in the second time through
mv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table.csv ~/CCRACD/MetabolomicProgramOutput/tmpTable.csv

echo "

#####

Completed creating Positive ion part of the table

#####

"
fi

if [[ $WasANegFileSelected = 3 ]];then

#Remove the titles from the pos file to prevent a doubling of the titles and further downstream
problems and combines the pos ion and neg ion files that are made in the loop
if [ -f ~/CCRACD/MetabolomicProgramOutput/tmpTable.csv ];then

awk -v x=$SelectedCompounds_Condition1LineNumberStart -F '$\t' 'BEGIN { OFS = FS } ; NR>=x
{ print }' ~/CCRACD/MetabolomicProgramOutput/tmpTable.csv >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table.csv
fi

#removes the now unnecessary generic named pos file
rm ~/CCRACD/MetabolomicProgramOutput/tmpTable.csv

echo "

#####

Completed creating Negative ion part of the table

#####

"
fi

```

```

WasANegFileSelected=$(( $WasANegFileSelected + 3 ))

WasAPosFileSelected=$(( $WasAPosFileSelected + 1 ))

done

fi

if [[ $WasAPosFileSelected = 1 ]];then

#if only the pos file has run through the loop it will change the generic named file back into its proper
name
mv ~/CCRACD/MetabolomicProgramOutput/tmpTable.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv

fi

##### Trimming table down / calculating to core information #####

#Make a new table for graphing from, with only column 1 and 4 onwards
cut -f2,3 --complement
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_Graphing.csv

#####

# below - selecting abundance type

#####
#select the raw or norm data depending on settings if 'not applicable' is chosen then run all that data
as good to go
AbundanceChoiceType=$(awk 'BEGIN {FS = "|"} {print $7}'
~/CCRACD/tmp/Metab_RawCrossCondition_FilesSelected.txt)

if [[ "$AbundanceChoiceType" == "-" ]];then
echo "No Abundance type given, constructing table as appropriate"

else

echo "Abundance type setting is: $AbundanceChoiceType . Creating table based solely on
$AbundanceChoiceType data"

#find the column $AbundanceChoiceType was on
AbundanceChoiceTypeColumnNumber=$(awk -v A="$AbundanceChoiceType" -F $'\t' 'BEGIN {
OFS = FS } ; { for (i=1; i<=NF; i++) if ($i==A) { print i } }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_Graphing.csv)

#if that phrase isnt on column 3 delete the columns between 2 and the column number it is found on
if [[ "$AbundanceChoiceTypeColumnNumber" != 3 ]];then

```

```

echo "abundance type chosen is not the first abundance type in the file"

cut -f 1,2,"$AbundanceChoiceTypeColumnNumber"-
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphingtmp.csv

mv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphingtmp.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv

#else it is column 3, so this works out what the abundance that hasn't been selected is called, so it can
be removed
else

echo "abundance type chosen is the first abundance type in the file"

if [[ "$AbundanceChoiceType" == "Normalised abundance" ]];then
    NotTheAbundanceChoiceType="Raw abundance"
else
    NotTheAbundanceChoiceType="Normalised abundance"
fi

#look for string $NotTheAbundanceChoiceType, if not found then no further columns need removing.
if found delete columns from and including $AbundanceChoiceTypeColumnNumber

AbundanceChoiceTypeColumnNumber=$(awk -v A="$NotTheAbundanceChoiceType" -F
$'\t' 'BEGIN { OFS = FS }; { for (i=1; i<=NF; i++) if ($i==A) {print i} }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv)

#this avoids the no number problem if there is only one abundance name in the file and they select
that name
AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber + 1 ))

if [[ "$AbundanceChoiceTypeColumnNumber" == 1 ]];then
echo "$AbundanceChoiceType data was selected and is the only type of abundance present,
running now "

else

#remove 2 rather than 1 to make up for the additional 1 you added earlier that ensured it would be a
number regardless
AbundanceChoiceTypeColumnNumber=$(( $AbundanceChoiceTypeColumnNumber - 2 ))
echo "$AbundanceChoiceTypeColumnNumber"
cut -f 1-"$AbundanceChoiceTypeColumnNumber"
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphingtmp.csv

```

```

mv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphingtmp.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv

echo "$NotTheAbundanceChoiceType that was not selected has been found and removed"

fi

fi

fi

#####

# above - selecting abundance type

#####

#Find the name of the first sample (or sub condition)
FirstSubSampleName=$(awk -v g=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS } ; NR==g {print $3}'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv)

echo "
first Sample/SubCondition is: $FirstSubSampleName"

#now count how many columns in that line from given column start contain that specific name
NumberOfFirstSubSampleName=$(awk -v g=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS } ; NR==g
{print $0}'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv | grep -o "$FirstSubSampleName" | wc -l )

echo "
The first Sample/SubCondition ("$FirstSubSampleName") contains $NumberOfFirstSubSampleName
replicates"

#Variable for how many different sample titles there are
#count how many columns from col 3

NumberOfCols=$(cut --complement -f 1,2
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv | awk -v g=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS } ; NR==g {print
$0}' | awk -F $'\t' 'BEGIN { OFS = FS } ; {print NF; exit} ')

echo "
Number of columns associated to samples/sub-conditions and their replicates is: $NumberOfCols
"

#loop
#take all the columns from the title line starting from col 3 > each column as a new line (can do this
by looping to maximum number of column - grabbing each column individually and amending it to
the same temp file)
touch ~/CCRACD/MetabolomicProgramOutput/tmp.csv

```

```

col=1
for i in $(seq 1 $NumberOfCols)
do

cut --complement -f 1,2
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_Graphing.csv | awk -v e=$TitlesLineEnd -v g=$col -F '$\t' 'BEGIN { OFS = FS } ;
NR==e {print $g}' >> ~/CCRACD/MetabolomicProgramOutput/tmp.csv

col=$(( $col + 1 ))

done

#count number of lines
NumberOfSamplesAndReps=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/tmp.csv)

#loop to number of lines - where each line is grepped for in the file, if a match is found the line you're
searching from is removed'

echo "#####

Creating list of each sample type

#####"

line=1
for i in $(seq 1 $NumberOfCols)
do

#Gets the sample name that has replicates but you need to get rid of the replicate names
SampleCheckAndRemoveRepeats=$(awk -v e=$line -F '$\t' 'BEGIN { OFS = FS } ; NR==e'
~/CCRACD/MetabolomicProgramOutput/tmp.csv)

echo "SampleCheckAndRemoveRepeats $SampleCheckAndRemoveRepeats"

#ensures the cutting process doesn't remove the exact/line string in question
line_1=$(( $line + 1 ))

#takes all the lines of names below the one you're looking at
awk -v e=$line_1 -F '$\t' 'BEGIN { OFS = FS } ; NR>=e {print $0}'
~/CCRACD/MetabolomicProgramOutput/tmp.csv >
~/CCRACD/MetabolomicProgramOutput/tmp1.csv

#Takes all the names that don't match that string and makes a new file with them
grep -v "$SampleCheckAndRemoveRepeats" ~/CCRACD/MetabolomicProgramOutput/tmp1.csv >
~/CCRACD/MetabolomicProgramOutput/tmp.csv

#Adds the now only version of the sample name into that file
echo "$SampleCheckAndRemoveRepeats" >> ~/CCRACD/MetabolomicProgramOutput/tmp.csv

done

TypesOfSamples=$(cat ~/CCRACD/MetabolomicProgramOutput/tmp.csv)

```

```

echo "
The different types of samples/sub-conditions in your file are:
$TypesOfSamples
"

rm ~/CCRACD/MetabolomicProgramOutput/tmp1.csv

#gives the number of unique names
NumberOfUniqueSamples=$(grep -c -v "HelloThisIsMyAntiMatch"
~/CCRACD/MetabolomicProgramOutput/tmp.csv)

echo "Number of unique samples/sub-conditions: $NumberOfUniqueSamples"

#creates file ready for the averaged data in table form
touch
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

NumberRepsPerSample=$( echo "$NumberOfCols / $NumberOfUniqueSamples" | bc )

echo "
Number of replicates per sample/sub-condition: $NumberRepsPerSample
"

line=1

#loop for finding and averaging each unique sample/sub-condition
for i in $(seq 1 $NumberOfUniqueSamples);do

rm ~/CCRACD/MetabolomicProgramOutput/tmp2.csv

#gets the unique name to find the replicates of
UniqueSampleName=$(awk -v e=$line -F $'\t' 'BEGIN { OFS = FS }; NR==e {print $0}'
~/CCRACD/MetabolomicProgramOutput/tmp.csv)

echo "
#####-----
#####

          Calculating each compound average for sample/sub-condition ${UniqueSampleName}...

#####-----
#####
"

col=3
echo "Column number the data starts on: $col"
RepNumber=1
#identifies any matches to that name and outputs the column
for i in $(seq 1 $NumberOfCols);do

#takes the columns replicate data heading
NameToCheck=$(awk -v y=$col -v z=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS }; NR==z {print
$y}'

```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv)
```

```
#grabs replicate columns and combines them in one file
```

```
#First checks if the Name = the unique sample name from the list previously made  
if [[ "$NameToCheck" == "$UniqueSampleName" ]];then
```

```
#grabs the column, cuts off the titles
```

```
awk -v y=$col -F $'\t' 'BEGIN { OFS = FS }; {print $y}'  
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv | awk -v a=$SelectedCompounds_Condition1LineNumberStart -F $'\t' 'BEGIN { OFS = FS }; NR==a,NR==10000' >  
~/CCRACD/MetabolomicProgramOutput/InBetween.csv
```

```
# to be safe i've added an if - so that it only combines the two files if tmp2 file is present
```

```
#adds the column as a column rather than a line for putting the replicates together column by column
```

```
if [[ -f ~/CCRACD/MetabolomicProgramOutput/tmp2.csv ]];then  
paste ~/CCRACD/MetabolomicProgramOutput/InBetween.csv  
~/CCRACD/MetabolomicProgramOutput/tmp2.csv >  
~/CCRACD/MetabolomicProgramOutput/tmp2tmp.csv  
else cp ~/CCRACD/MetabolomicProgramOutput/InBetween.csv  
~/CCRACD/MetabolomicProgramOutput/tmp2tmp.csv  
fi
```

```
mv ~/CCRACD/MetabolomicProgramOutput/tmp2tmp.csv  
~/CCRACD/MetabolomicProgramOutput/tmp2.csv
```

```
echo "${UniqueSampleName}: Gathered all replicate $RepNumber data
```

```
"
```

```
RepNumber=$(( $RepNumber + 1 ))
```

```
fi
```

```
col=$(( $col + 1 ))
```

```
done
```

```
RepNumber=$(( $RepNumber - 1 ))
```

```
echo "
```

```
All $RepNumber replicate data for sample/sub-condition $UniqueSampleName ascertained
```

```
"
```

```
cat ~/CCRACD/MetabolomicProgramOutput/tmp2.csv
```

```
#average each line from ~/CCRACD/MetabolomicProgramOutput/tmp2.csv and output amending to  
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_GraphingAvg.csv
```

```
CompoundLineNumber=$SelectedCompounds_Condition1LineNumberStart
```

```
NumberOfCompoundsIncludingAppearanceAcrossConditions=$(awk -v  
x=$SelectedCompounds_Condition1LineNumberStart -F $'\t' 'BEGIN { OFS = FS }; NR>=x { print  
$1 }')
```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv | grep -c '[0-9]')
```

```
line2=1
for i in $(seq 1 $NumberOfCompoundsIncludingAppearanceAcrossConditions);do

#sets the compound name variable
CompoundName=$(awk -v x=$CompoundLineNumber -F $'\t' 'BEGIN { OFS = FS } ; NR==x {
print $1 }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv)

ConditionName=$(awk -v x=$CompoundLineNumber -F $'\t' 'BEGIN { OFS = FS } ; NR==x {
print $2 }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv)
```

```
echo "
```

Calculating compound $\{CompoundName\}$'s average from condition $\{ConditionName\}$ "

```
col1=1
Rep2=0
Rep=0
#loop for adding the number of replicates for a unique sample name together into 1 number and
outputting the average
for i in $(seq 1 $NumberRepsPerSample)
do

echo "
Replicate: $col1"

#if there is a previous rep
if [[ $Rep > 0 ]];then Rep2=$Rep

PreviousReplicate=$(( $col1 - 1 ))

if [[ $PreviousReplicate == 1 ]];then

Rep2=$Rep

echo "Previous replicate: ${PreviousReplicate}, abundance: $Rep2"
else
if [[ $PreviousReplicate > 1 ]]

then
Rep2=$Rep

echo "Previous replicate: ${PreviousReplicate}, total abundance of all previous replicates: $Rep2"
```

```

fi
fi
fi

#gets each rep with each loop
Rep=$(awk -v z=$col1 -v m=$line2 -F '\t' 'BEGIN { OFS = FS } ; NR==m { print $z}'
~/CCRACD/MetabolomicProgramOutput/tmp2.csv)

echo "Replicate $col1 abundance: $Rep"

#if there is a previous loop rep it adds it to the new loops rep
if [[ $Rep > 0 ]] && [[ $Rep2 > 0 ]]

then Rep=$( echo "$Rep + $Rep2" | bc )

echo "Total replicate abundance thus far: $Rep"

fi

col1=$(( $col1 + 1 ))

done

Average=$( echo "$Rep / $NumberRepsPerSample" | bc -l )

echo "$Average" >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

if [[ $line2 = $NumberOfCompoundsIncludingAppearanceAcrossConditions ]];then

if [[ -f
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv ]];then
paste
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv > ~/CCRACD/MetabolomicProgramOutput/temporaryGraphAvg.csv

mv ~/CCRACD/MetabolomicProgramOutput/temporaryGraphAvg.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv

else
mv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv

fi

```

```

fi

echo "
Sample/Sub-condition ${UniqueSampleName}, compound $CompoundName average: $Average
"

echo "
#####-----
#####

#All compounds throughout all conditions for sample/sub-condition ${UniqueSampleName} have
been complete

#####-----
#####

# "

#this ensures each compounds got an average for the sample

CompoundLineNumber=$(( $CompoundLineNumber + 1 ))

line2=$(( $line2 + 1 ))

done

if [[ -f
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv ]]
then
rm
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv
fi

#this changes the sample and starts the process again
line=$(( $line + 1 ))

echo "
#####-----
#####

All compounds throughout all conditions for sample/sub-condition ${UniqueSampleName} have
been complete

#####-----
#####

"

```

done

```
echo "
```

```
-----  
Adding titles and compound names back into the new average table  
-----  
"
```

```
#Adds the compound names from the graphing table into the data avg graphing table
```

```
awk -F $'\t' 'BEGIN { OFS = FS }; {print $1,$2}'
```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv | awk -v a=$SelectedCompounds_Condition1LineNumberStart -F $'\t' 'BEGIN { OFS = FS }; NR==a,NR==10000' >
```

```
~/CCRACD/MetabolomicProgramOutput/InBetween.csv
```

```
paste ~/CCRACD/MetabolomicProgramOutput/InBetween.csv
```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_GraphingAvgtmp.csv >
```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_GraphingAvgtmp1.csv
```

```
#make a new file with the opening line titles
```

```
#first grab compound and condition titles
```

```
awk -v x=$TitlesLineEnd -F $'\t' 'BEGIN { OFS = FS }; NR==x {print $1,$2}'
```

```
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"_Table_Graphing.csv > ~/CCRACD/MetabolomicProgramOutput/Titlestmp.csv
```

```
#then use the list of unique sample names and make a file with them, each being a new column  
line=1
```

```
for i in $(seq 1 $NumberOfUniqueSamples)  
do
```

```
awk -v x=$line -F $'\t' 'BEGIN { OFS = FS }; NR==x {print $1}'
```

```
~/CCRACD/MetabolomicProgramOutput/tmp.csv >
```

```
~/CCRACD/MetabolomicProgramOutput/tmp3.csv
```

```
if [[ -f ~/CCRACD/MetabolomicProgramOutput/tmp4.csv ]]  
then
```

```
paste ~/CCRACD/MetabolomicProgramOutput/tmp4.csv
```

```
~/CCRACD/MetabolomicProgramOutput/tmp3.csv >
```

```
~/CCRACD/MetabolomicProgramOutput/tmp5.csv
```

```
mv ~/CCRACD/MetabolomicProgramOutput/tmp5.csv
```

```
~/CCRACD/MetabolomicProgramOutput/tmp4.csv
```

```
else
```

```
mv ~/CCRACD/MetabolomicProgramOutput/tmp3.csv
```

```
~/CCRACD/MetabolomicProgramOutput/tmp4.csv
```

```

fi

line=$(( $line + 1 ))

done

mv ~/CCRACD/MetabolomicProgramOutput/tmp4.csv
~/CCRACD/MetabolomicProgramOutput/tmp2.csv

#then paste the Titlestmp.csv and tmp2.csv file together
paste ~/CCRACD/MetabolomicProgramOutput/Titlestmp.csv
~/CCRACD/MetabolomicProgramOutput/tmp2.csv >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

#add all the text from the avg graphing file to the file containing the titles
awk -F $'\t' 'BEGIN { OFS = FS } ; NR==1,NR==10000'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp1.csv >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

rm
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp1.csv

rm
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv

rm ~/CCRACD/MetabolomicProgramOutput/Titlestmp.csv

rm ~/CCRACD/MetabolomicProgramOutput/tmp.csv

rm ~/CCRACD/MetabolomicProgramOutput/tmp2.csv

rm ~/CCRACD/MetabolomicProgramOutput/tmp3.csv

rm ~/CCRACD/MetabolomicProgramOutput/InBetween.csv

#rm
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table.csv

echo "

```

```

#####-----
#####

```

Supplementing table (where compounds haven't been found) with 0's to ensure each compound has condition data

```
#####-----  
#####  
"
```

```
#Sorts it so each compound has each condition data in it by creating conditions with 0 data if presence  
wasn't found
```

```
#create a text file for each condition, adding 0's in each column next to it for the total number of  
unique samples
```

```
NumberOfConditions=$(grep -c -v "HelloThisIsMyAntiMatch"  
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
```

```
echo "Number Of Conditions: $NumberOfConditions"
```

```
condition=1  
for i in $(seq 1 $NumberOfConditions);do
```

```
echo "Condition: $condition"
```

```
ConditionNameVar=$(awk -v x=$condition -F '$\t' 'BEGIN { OFS = FS } NR==x { print $1}'  
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
```

```
echo "Condition Name: $ConditionNameVar"
```

```
echo "$ConditionNameVar" > ~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv
```

```
echo "Number Of Unique Samples/sub-conditions: $NumberOfUniqueSamples"  
#make a file containing the number of 0s in tab delimited column as the same number of unique  
samples
```

```
  #loop NumberOfUniqueSamples  
  for i in $(seq 1 $NumberOfUniqueSamples);do
```

```
    if [[ -f ~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv ]]  
    then  
      echo "0" > ~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv
```

```
      paste ~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv  
~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv >  
~/CCRACD/MetabolomicProgramOutput/ZeroFile2.csv  
      mv ~/CCRACD/MetabolomicProgramOutput/ZeroFile2.csv  
~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv
```

```
    else  
      echo "0" > ~/CCRACD/MetabolomicProgramOutput/PreZeroFile.csv  
      paste ~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv  
~/CCRACD/MetabolomicProgramOutput/PreZeroFile.csv >
```

```
~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv  
    fi  
  done
```

```
mv ~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv  
~/CCRACD/MetabolomicProgramOutput/"$ConditionNameVar"_ZeroFile.csv
```

```
echo "Condition $condition supplementary line:"
```

```
cat ~/CCRACD/MetabolomicProgramOutput/"$ConditionNameVar"_ZeroFile.csv
```

#sometimes causes problems that locks the file and prevents deletion, remaking the file with touch and then deleting works

```
touch ~/CCRACD/MetabolomicProgramOutput/ZeroFile2.csv
touch ~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv
touch ~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv
```

```
rm ~/CCRACD/MetabolomicProgramOutput/ZeroFile2.csv
rm ~/CCRACD/MetabolomicProgramOutput/ZeroFile1.csv
rm ~/CCRACD/MetabolomicProgramOutput/ZeroFile.csv
```

```
condition=$(( $condition + 1 ))
done
```

```
echo "
Finished creating the required supplementary lines
"
```

```
#makes a temp file with a list of each unique name from graph file
awk -F $'\t' 'BEGIN { OFS = FS }; !_[ $1 ]++'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_GraphingAvg.csv > ~/CCRACD/MetabolomicProgramOutput/tmp6.csv
```

```
#this loop uses the single line files made above for each condition, to add the lines to the table where
appropriate
changecondition=1
```

```
cp
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_GraphingAvg.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_GraphingAvgtmpForLoop.csv
```

```
#runs for number of compounds in graphing file
for i in $(seq 1 $NumberOfConditions);do
```

```
line=2
```

```
#grep all lines with compound name, if there is a match for condition searching for then do nothing,
otherwise make new version of that condition 000s file but with compound name added in this line to
location
```

```
echo "Number Of Compounds: $NumberOfCompounds"
```

```
NumberOfCompoundsIncludingAppearanceAcrossConditionsNew=$(awk -v x=2 -F $'\t' 'BEGIN {
OFS = FS }; NR>=x { print $1 }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_GraphingAvgtmpForLoop.csv | grep -c '[0-9]')
```

```
for i in $(seq 1 $NumberOfCompoundsIncludingAppearanceAcrossConditionsNew);do
```

```
CompoundName=$(awk -v x=$line -F $'\t' 'BEGIN { OFS = FS }; NR==x { print $1 }'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Name"
_Table_GraphingAvgtmpForLoop.csv)
```

```

echo "Compound Name: $CompoundName"

condition=$(awk -v x=$changecondition -F $'\t' 'BEGIN { OFS = FS } NR==x {print $1}'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)

echo "Condition: $condition"

DoesConditionDataExistInGivenCompound=$(awk -v x=$CompoundName -F $'\t' 'BEGIN { OFS =
FS } ; $1 == x'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv)

echo "DoesConditionDataExistInGivenCompound: $DoesConditionDataExistInGivenCompound"

#this insures that if there is no match that the conditionhit will have changed from previous times
when there was a hit
ConditionHit=0

echo "ConditionHit: $ConditionHit"

echo "$DoesConditionDataExistInGivenCompound" >
~/CCRACD/MetabolomicProgramOutput/tmp.csv

#maybe add a bit here so its only looking for the condition from the the condition column

ConditionHit=$(grep -c "$condition" ~/CCRACD/MetabolomicProgramOutput/tmp.csv)

#ConditionHit=$(echo "$DoesConditionDataExistInGivenCompound" | grep -w -c -v
"HelloThisIsMyAntiMatch")

echo "ConditionHit: $ConditionHit"

echo "Finished checking for match"

if [[ "$ConditionHit" < 1 ]];then
echo "Does Condition (${condition}) Data Already Exist For Compound ${CompoundName}:
NO

    Supplementing table to contain the condition with 0 abundance data
"

#make a copy of the template 0's file
cp ~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFile.csv
~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNo.csv

#make file for compound name
echo "${CompoundName}" > ~/CCRACD/MetabolomicProgramOutput/CompoundNameFile.csv

#add compound name to the copy of the 0's condition file

paste ~/CCRACD/MetabolomicProgramOutput/CompoundNameFile.csv
~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNo.csv >
~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNotmp1.csv

```

```

# make a variable the line form the condition text file containing the 0 data and condition in
conditiontxt=$(head -n1
~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNotmp1.csv)

#insert $conditiontxt at line number $line in graphing file

echo "$conditiontxt" >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

echo "
                Supplement Complete
"

#this line number ensures that if a line is added, the next time through its accounted for

else
echo "Does Condition (${condition}) Data Already Exist For Compound ${CompoundName}:
                YES"

fi

    line=$(( $line + 1 ))
done

    changecondition=$(( $changecondition + 1 ))

done

# sort/order by compound number

    #make a file with the titles in
awk -F '$\t' 'BEGIN { OFS = FS } ; NR==1'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv
    #grab all the lines from line 2 down, pipe into sort and sort in numerical order from compound
names, and amend (>>) to the title file

awk -F '$\t' 'BEGIN { OFS = FS } ; NR==2,NR==10000'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv | sort -V -k 1,1 >>
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv

mv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

changecondition=1

```

```

for i in $(seq 1 $NumberOfConditions);do
condition=$(awk -v x=$changecondition -F '\t' 'BEGIN { OFS = FS } NR==x {print $1}'
~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt)
rm ~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFile.csv
rm ~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNo.csv
rm ~/CCRACD/MetabolomicProgramOutput/"$condition"_ZeroFileWithCompoundNotmp1.csv

changecondition=$(( $changecondition + 1 ))
done

rm ~/CCRACD/MetabolomicProgramOutput/PreZeroFile.csv
rm ~/CCRACD/MetabolomicProgramOutput/tmp6.csv
rm ~/CCRACD/MetabolomicProgramOutput/CompoundNameFile.csv
rm ~/CCRACD/MetabolomicProgramOutput/tmp.csv

rm ~/CCRACD/MetabolomicProgramOutput/ZeroFile2.csv

#Combining compound names with there given condition name
awk -F '\t' 'BEGIN { OFS = FS } ; $1 =$1$2 {print}'
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv | cut -f2 --complement >
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

cp
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv
~/CCRACD/MetabolomicProgram_Input/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvg.csv

rm
~/CCRACD/MetabolomicProgramOutput/"$MetabInputData_Name"_"$MetabInputData_NegIon_Na
me"_Table_GraphingAvgtmp.csv

rm ~/CCRACD/MetabolomicProgramOutput/Conditionlist.txt

rm ~/CCRACD/MetabolomicProgramOutput/UniqueCompoundList.csv
rm ~/CCRACD/MetabolomicProgramOutput/LinesToBeAddedIntoTable.csv

echo | ~/CCRACD/MetabolomicsProgramMenu.sh

```

10.8.3.5 Data plotting

10.8.3.5.1 Heatmap and dendrogram

```
### READ IN giles.txt as 'data' ###
rm(list=ls())
data=read.delim("~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt", header = T,
row.names=1, check.names=F)
View(data)

library(vegan)

data.dist <- dist(t(data), method="euclidean")
data.dist

data.hr <- hclust(data.dist, method="average")

#If plottin as pdf dendrogram of all OTUs sequenced within all samples.
pdf(file "~/CCRACD/MetabolomicProgramOutput/Heatmaps_dendo.test.pdf", width=100,
height=70)

#If not then just this
#plot(data.hr)

#Finish PDF plotting
#dev.off()

### form x axis dataframe using metobolites
row.dist <- dist((data), method="euclidean")
row.dist
### cluster analysis of x axis metabolites
rows.hr <- hclust(row.dist, method="average")

plot(rows.hr)
library(gplots)

# If saving as plot as seperate PDF file
pdf(file "~/CCRACD/MetabolomicProgramOutput/HeatMap.pdf", width=60, height=50)

## otherwise this
bluered <- function(n) colorpanel(n, 'blue', 'white', 'red')
op <- par(oma=c(7,0.1,0.1,16), cex.main=1.0)
heatmap.2(as.matrix(data), col=bluered(255), Colv=as.dendrogram(data.hr), colsep=(1:48),
rowsep=(1:9), sepcolor='white', Rowv=as.dendrogram(rows.hr), scale="row", key=TRUE,
keysize=1.0, key.title=NA, symkey=FALSE, density.info=c("histogram"), margins=c(5,10),
trace="none", cexRow=4.0, cexCol=3.5)
### heatmap.2 is command from gplots package ###
# stop plotting to pdf
dev.off()
```

10.8.3.5.2 PCA, Correlation plot, Dot plot, Biplot

```
rm(list=ls())
library(ggbiplot)
library(lattice)

MetabMeta=read.delim("~/CCRACD/MetabolomicProgram_Input/TableTxtFile.txt", header=T,
row.name=1, check.names = F)

MetabMeta=t(MetabMeta)

# Look at the correlations
#install.packages("gclus")
library(gclus)
my.abs <- abs(cor(MetabMeta[,-1])) ##Can change to view correlations of other datasets
my.abs
my.colors <- dmat.color(my.abs)
my.colors
my.ordered <- order.single(cor(MetabMeta[,-1]))
my.ordered

pdf(file="~/CCRACD/MetabolomicProgramOutput/Correlations.pdf", width=50, height=50)
cpairs(MetabMeta, my.ordered, panel.colors=my.colors, gap=0.5)
dev.off()

# Do the PCA

#Make sure all values are centred and scaled accordingly by "=T"
my.prc <- prcomp(MetabMeta[,-1], center=TRUE, scale=TRUE) ##Can change to view correlations
of other datasets

#Apply Kaiser criterion (variances of >1.0) to select components

#Can also use Scree plots to decide which components to use.
#Select components until change between variances drops to <1
screeplot(my.prc, main="Scree Plot", xlab="Components")
#Sometimes easier to see with line
pdf(file="~/CCRACD/MetabolomicProgramOutput/Metab.PCA.all.scree.pdf", width=7, height=7)
screeplot(my.prc, main="Scree Plot for Metabolomics PCA", type="line" )
dev.off()
# DotChart PC1

pdf(file="~/CCRACD/MetabolomicProgramOutput/Metab.PCA.Dotplot.pdf", width=7, height=7)
#calculate loading values for dotplot
load <- my.prc$rotation
load
sorted.loadings<- load[order(load[, 1]), 1]
sorted.loadings
#Create labels for dotchart of 1st principle component
myTitle <- "Loadings Plot for PC1"
myXlab <- "Variable Loadings"
dotchart(sorted.loadings, main=myTitle, xlab=myXlab, cex=0.3, col="red")
#dotchart shows which variables have most significant effect on each PC
```

```

# DotChart PC2

#calculate loading values for dotplot
sorted.loadings <- load[order(load[, 2]), 2]
#Create labels for dotchart of 2nd principle component
myTitle <- "Loadings Plot for PC2"
myXlab <- "Variable Loadings"
dotchart(sorted.loadings, main=myTitle, xlab=myXlab, cex=0.3, col="red")
#dotchart shows which variables have most significant effect on each PC

dev.off()

# Now draw the BiPlot
pdf(file="~/CCRACD/MetabolomicProgramOutput/Metab.Biplot.pdf", width=10, height=10)
ggbiplot(my.prc, choices=1:2, circle=0.69, var.scale=1, obs.scale =1, var.axes=T, varname.size=2,
labels.size=8, labels = rownames(MetabMeta))

dev.off()

#Then Build just the PCA

pdf(file="~/CCRACD/MetabolomicProgramOutput/Metab.PCA.pdf", width=10, height=10)
# load ggplot2
library(ggplot2)

# create data frame with scores
scores = as.data.frame(my.prc$x)

# plot of observations
ggplot(data = scores, aes(x = PC1, y = PC2, label = rownames(scores))) +
  geom_hline(yintercept = 0, colour = "gray65") +
  geom_vline(xintercept = 0, colour = "gray65") +
  geom_text(colour = "tomato", alpha = 0.8, size = 4) +
  ggtitle("PCA plot of Cell wall lipids")

dev.off()

```

10.9 Appendix 9

10.9.1 GGOSS installation

```
#!/bin/sh

#if yad isn't installed, install it
YadPresent=$(which yad)
if [ $YadPresent != "/usr/bin/yad" ];then
sudo apt-get install yad
fi

if [ -f ~/GGOSSInstallType.txt ];then
rm ~/GGOSSInstallType.txt
fi

ICON=~/.GGOSS_InstallFile/Pictures/DNA_1Installer.jpg

#choose installation type
yad --title="GENOME SEQUENCING PROGRAM - GGOSS Installation" --center --size=fit --form \
Giles Holt" --text="GGOSS Installation" --center --size=fit --form \
--field="Installation type":CB \
'Complete Install (Installs the program and all associated OSS)!Careful Complete Install (Installs the
program and any associated OSS that is not already installed)!Program Install Only!' \
--text-info --show-uri --height=100 --width=400 --center --wrap \
--button="gtk-save:0" --button="gtk-close:1" --editable --filename=~/.GGOSSInstallType.txt >
~/GGOSSInstallType.txt

(InstallationType=$(awk -F '|' 'NR==1 {print $1}' ~/.GGOSSInstallType.txt)

echo 1
echo "#Installation type selected: $InstallationType"

if [ -f ~/.GGOSSInstallType.txt ];then
rm ~/.GGOSSInstallType.txt
fi

#####build directories and portion scripts out into the directories before sorting the installs#####

if [ "$InstallationType" = "Complete Install (Installs the program and all associated OSS)" ] || [
"$InstallationType" = "Careful Complete Install (Installs the program and any associated OSS that is
not already installed)" ] || [ "$InstallationType" = "Program Install Only" ]
then

##### Build InputOutput directories #####

echo 1
echo "#Building program directories 1% Complete"

if [ ! -d ~/.GGOSS_InputOutput ];then
mkdir ~/.GGOSS_InputOutput
fi
```

```

if [ ! -d ~/GGOSS_InputOutput/AdapterTrimmedFiles ];then
mkdir ~/GGOSS_InputOutput/AdapterTrimmedFiles
fi

if [ ! -d ~/GGOSS_InputOutput/Blast ];then
mkdir ~/GGOSS_InputOutput/Blast
fi

if [ ! -d ~/GGOSS_InputOutput/BWA ];then
mkdir ~/GGOSS_InputOutput/BWA
fi

if [ ! -d ~/GGOSS_InputOutput/Fastqc ];then
mkdir ~/GGOSS_InputOutput/Fastqc
fi

if [ ! -d ~/GGOSS_InputOutput/FastqFiles ];then
mkdir ~/GGOSS_InputOutput/FastqFiles
fi

if [ ! -d ~/GGOSS_InputOutput/furtherAnalysis_ComparitiveAnalysis ];then
mkdir ~/GGOSS_InputOutput/furtherAnalysis_ComparitiveAnalysis
fi

if [ ! -d ~/GGOSS_InputOutput/Khmer ];then
mkdir ~/GGOSS_InputOutput/Khmer
fi

if [ ! -d ~/GGOSS_InputOutput/MEGAN_output ];then
mkdir ~/GGOSS_InputOutput/MEGAN_output
fi

if [ ! -d ~/GGOSS_InputOutput/Mothur ];then
mkdir ~/GGOSS_InputOutput/Mothur
fi

if [ ! -d ~/GGOSS_InputOutput/PIPITS ];then
mkdir ~/GGOSS_InputOutput/PIPITS
fi

if [ ! -d ~/GGOSS_InputOutput/PRICETI ];then
mkdir ~/GGOSS_InputOutput/PRICETI
fi

if [ ! -d ~/GGOSS_InputOutput/Prokka ];then
mkdir ~/GGOSS_InputOutput/Prokka
fi

if [ ! -d ~/GGOSS_InputOutput/QUAST ];then
mkdir ~/GGOSS_InputOutput/QUAST
fi

if [ ! -d ~/GGOSS_InputOutput/Sickle ];then
mkdir ~/GGOSS_InputOutput/Sickle
fi

```

```
if [ ! -d ~/GGOSS_InputOutput/SPAdes ];then
mkdir ~/GGOSS_InputOutput/SPAdes
fi
```

```
if [ ! -d ~/GGOSS_InputOutput/Velvet ];then
mkdir ~/GGOSS_InputOutput/Velvet
fi
```

Build program directories

```
if [ ! -d ~/GGOSS ];then
mkdir ~/GGOSS
fi
```

```
if [ ! -d ~/GGOSS/Buttons ];then
mkdir ~/GGOSS/Buttons
fi
```

```
if [ ! -d ~/GGOSS/downloads ];then
mkdir ~/GGOSS/downloads
fi
```

```
if [ ! -d ~/GGOSS/downloads/R_LibraryPackages ];then
mkdir ~/GGOSS/downloads/R_LibraryPackages
fi
```

```
if [ ! -d ~/GGOSS/Fastafiles ];then
mkdir ~/GGOSS/Fastafiles
fi
```

```
if [ ! -d ~/GGOSS/FastqFiles ];then
mkdir ~/GGOSS/FastqFiles
fi
```

```
if [ ! -d ~/GGOSS/Giles_Adapter_Program ];then
mkdir ~/GGOSS/Giles_Adapter_Program
fi
```

```
if [ ! -d ~/GGOSS/LogFiles ];then
mkdir ~/GGOSS/LogFiles
fi
```

```
if [ ! -d ~/GGOSS/MEGAN_Com_txt_files ];then
mkdir ~/GGOSS/MEGAN_Com_txt_files
fi
```

```
if [ ! -d ~/GGOSS/mothur ];then
mkdir ~/GGOSS/mothur
fi
```

```
if [ ! -d ~/GGOSS/Output ];then
mkdir ~/GGOSS/Output
fi
```

```

if [ ! -d ~/GGOSS/Pictures ];then
mkdir ~/GGOSS/Pictures
fi

if [ ! -d ~/GGOSS/Scripts ];then
mkdir ~/GGOSS/Scripts
fi

if [ ! -d ~/GGOSS/Scripts/R_Scripts ];then
mkdir ~/GGOSS/Scripts/R_Scripts
fi

if [ ! -d ~/GGOSS/Scripts/FileSelection ];then
mkdir ~/GGOSS/Scripts/FileSelection
fi

if [ ! -d ~/GGOSS/Shuffled ];then
mkdir ~/GGOSS/Shuffled
fi

if [ ! -d ~/GGOSS/tmp ];then
mkdir ~/GGOSS/tmp
fi
echo 2
echo "#Program directories built    2% Complete"
fi
echo 2
echo "#Program directories built    2% Complete"
fi

#add pictures
mv -v ~/GGOSS_InstallFile/Pictures/* ~/GGOSS/Pictures/
ICON=~/.GGOSS/Pictures/DNA_1Installer.jpg

#add Scripts
mv -v ~/GGOSS_InstallFile/Scripts/* ~/GGOSS/Scripts/

#add Buttons
mv -v ~/GGOSS_InstallFile/Buttons/* ~/GGOSS/Buttons/

fi

#####---Script building/sorting section---#####

```

```

#At the end go to GGOSS directory and mass chmod all scripts
ChangeDirectory=1
for i in $(seq 1 3);do

if [ "$ChangeDirectory" = 1 ];then
ScriptsInDirectory=$(ls ~/GGOSS/*.sh | tr " " "\n")
fi

if [ "$ChangeDirectory" = 2 ];then
ScriptsInDirectory=$(ls ~/GGOSS/Buttons/*.sh | tr " " "\n")
fi

if [ "$ChangeDirectory" = 3 ];then
ScriptsInDirectory=$(ls ~/GGOSS/Scripts/*.sh | tr " " "\n")
fi

NumberOfScripts=$(echo "$ScriptsInDirectory" | grep -v -c "ThisIsMyAntiMatch")
LineForScript=1
for i in $(seq 1 "$NumberOfScripts");do
ScriptToChmod=$(echo "$ScriptsInDirectory" | awk -v n=$LineForScript 'NR==n {print}')

chmod 755 $ScriptsInDirectory

LineForScript=$(( $LineForScript + 1 ))
done

ChangeDirectory=$(( $ChangeDirectory + 1 ))
done

ScriptsInDirectory=$(ls ~/GGOSS/Buttons/*.sh)
NumberOfScripts=$(echo "$ScriptsInDirectory" | tr " " "\n")

ScriptsInDirectory=$(ls ~/GGOSS/Scripts/*.sh)
NumberOfScripts=$(echo "$ScriptsInDirectory" | tr " " "\n")

#####---Open Source Software Installation section---#####

if [ "$InstallationType" = "Complete Install (Installs the program and all associated OSS)" ] || [
"$InstallationType" = "Careful Complete Install (Installs the program and any associated OSS that is
not already installed)" ];then
#before each install check if it was set as careful, if so check if installed already, if its installed change
scripts so the path to the software is corrected

#####-----#####
#####----- SPAdes -----#####
#####-----#####

#####
#checks for presence
#matches
PresencePrep=$(find ~/ -name 'spades.py' -size +40k -size -55k 2>/dev/null)
#Best match
#Number of matches
NumberOfMatches=$(echo "$PresencePrep" | grep -c -v "ThisIsMyAntiMatch")

```

```

#If Careful version selected
if [ "$InstallationType" = "Careful Complete Install (Installs the program and any associated OSS
that is not already installed)" ] && [ "$NumberOfMatches" != "0" ];then
    line=1
    for i in $(seq 1 $NumberOfMatches)
    do
        #Grab each match individually, print third to last column (as it should be the spades file name),
strip away everything but the numbers
        echo "$PresencePrep" | awk -v x=$line 'NR==x {print}' | awk -F '/' '{print $(NF-2)}' | tr -d -c 0-9
    >> ~/MatchFind_VersionList.txt

        echo "_$line" >> ~/MatchFind_VersionList.txt

        line=$(( $line + 1 ))
    done

    #Pick the match with the latest version
    LatestVersionPrep=$(sort -t '_' -rk1 ~/MatchFind_VersionList.txt | head -n1 | awk -F '_' '{print $2}')
    LatestVersion=$(echo "$PresencePrep" | awk -v x=$LatestVersionPrep 'NR==x {print}' | awk -F '/'
'{'print $(NF-2)}')
    echo 20
    echo "#Most up-to-date version of SPAdes found on your system is: $LatestVersion    20%
Complete"
    rm ~/MatchFind_VersionList.txt

    #identifies path
    PathToSoftware=$(echo "$PresencePrep" | awk -v x=$LatestVersionPrep 'NR==x {print}')
    echo 22
    echo "#Path to $LatestVersion:$PathToSoftware    22% Complete"

    #add this SPAdes name and path to script
    sed -i -e "s|~/SPAdes-3.6.1-Linux/bin/spades.py|$PathToSoftware|g"
~/GGOSS/Scripts/SPAdesAssemblerTemplate.sh

fi

NumberOfMatches=$(( $NumberOfMatches + 1 ))
if [ "$InstallationType" = "Complete Install (Installs the program and all associated OSS)" ] || [
$NumberOfMatches = 1 ];then

####--- INSTALL SPAdes
echo 20
echo "#Installing SPAdes    20% Complete"

sudo apt-get -y install cmake
wget http://spades.bioinf.spbau.ru/release3.10.1/SPAdes-3.10.1-Linux.tar.gz
tar -xzf SPAdes-3.10.1-Linux.tar.gz

#PREFIX=/usr/local ./spades_compile.sh
#SPAdes installation directory to the PATH variable
cd SPAdes-3.10.1-Linux/bin/
sudo apt install spades

#add this SPAdes name and path to script

```

```

sed -i -e "s|~/SPAdes-3.6.1-Linux/bin/spades.py|~/SPAdes-3.10.1-Linux/bin/spades.py|g"
~/GGOSS/Scripts/SPAdesAssemblerTemplate.sh

    echo 22
    echo "#Installed SPAdes    22% Complete"

    else
    echo 22
    echo "#SPAdes is already installed    22% Complete"

    fi

#####-----#####
#####----- Artemis -----#####
#####-----#####

#If Careful version selected
    if [ "$InstallationType" = "Careful Complete Install (Installs the program and any associated OSS
that is not already installed)" ] && [ "$NumberOfMatches" != "0" ];then
        #checks for presence
        #matches
        PresencePrep=$(find ~/ -name 'artemis_sqlmap' 2>/dev/null)

        #Number of matches
        NumberOfMatches=$(echo "$PresencePrep" | grep -c -v "ThisIsMyAntiMatch")
        #Don't need Best match
        #Number of matches

        #identifies path
        PathToSoftware=$(echo "$PresencePrep" | awk -F '/' 'BEGIN {OFS=" "} NF{NF-=1};1')
        echo "Path to Artemis: $PathToSoftware"

    fi

    NumberOfMatches=$(( $NumberOfMatches + 1 ))
    if [ "$InstallationType" = "Complete Install (Installs the program and all associated OSS)" ] || [
$NumberOfMatches = 1 ];then
        echo 23
        echo "#Installing Artemis    23% Complete"
        wget ftp://ftp.sanger.ac.uk/pub/resources/software/artemis/artemis.tar.gz
        sudo gzip -d < artemis.tar.gz | tar xf -
        cd ~/artemis
        ./art
        echo 27
        echo "#Artemis Installed    27% Complete"

        else
        echo 27
        echo "#Artemis already installed    27% Complete"
        sed -i -e "s|~/artemis|$PathToSoftware|g" ~/GGOSS/Scripts/Artemis.sh

    fi

#####-----#####
#####----- Java -----#####

```

```

#####-----#####

#install regardless - wont do harm
echo 16
echo "#Installing Java"
sudo apt-get install default-jre
sudo apt-get install default-jdk
echo 20
echo "#Java Installed"

#####-----#####
#####----- git -----#####
#####-----#####

gitInstalledPrep=$(git version)

gitInstalled=$(case $gitInstalledPrep in
    "git version 2"* ) echo 1 ;;
esac)

if [[ "$gitInstalled" = 1 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the program
and any associated OSS that is not already installed)" ];then
echo 37
echo "#git is already installed    37% Complete"
else
echo 33
echo "#Installing git    33% Complete"

sudo apt-get install git

echo 37
echo "#git Installed    37% Complete"

fi

#####-----#####
#####----- pip -----#####
#####-----#####

pipInstalledPrep=$(pip -V)

pipInstalled=$(case $pipInstalledPrep in
    "pip 8"* ) echo 1 ;;
esac)

if [[ "$pipInstalled" = 1 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the program
and any associated OSS that is not already installed)" ];then
echo 42
echo "#pip is already installed    42% Complete"
sleep 1
else
echo 38
echo "#Installing pip    38% Complete"
sudo apt-get install python-pip

```

```

pip install --upgrade pip

echo 42
echo "#pip Installed    42% Complete"

fi

#####-----#####
#####----- Sickle -----#####
#####-----#####

#Sickle installer
SickleInstalledPrep=$(sickle 2>&1)

SickleInstalled=$(case $SickleInstalledPrep in
    "The program 'sickle' is currently not installed.*" ) echo 1 ;;
esac)

if [[ "$SickleInstalled" != 1 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the
program and any associated OSS that is not already installed)" ];then
echo 52
echo "#Sickle is already installed    52% Complete"
sleep 1
else
#requirements
    #Zlib
echo 43
echo "#Installing Sickle    43% Complete"
    wget http://www.zlib.net/zlib-1.2.11.tar.gz
    tar -xvzf zlib-1.2.11.tar.gz
echo 44
echo "#Installing Sickle    44% Complete"
    cd ~/zlib-1.2.11
    ./configure --prefix=/usr/local/zlib
echo 45
echo "#Installing Sickle    45% Complete"
    make

    sudo make install
echo 46
echo "#Installing Sickle    46% Complete"
    #Requests
    sudo -H pip install requests

    git clone git://github.com/kennethreitz/requests.git
echo 47
echo "#Installing Sickle    47% Complete"
    sudo python setup.py install
echo 48
echo "#Installing Sickle    48% Complete"
    #lxml
    git clone https://github.com/lxml/lxml.git lxml

```

```

    sudo apt-get install python-lxml
echo "#Installing Sickle    49% Complete"
echo 49
#Download and install sickle

sudo -H pip install sickle
echo "#Installing Sickle    50% Complete"
echo 50

sudo apt install sickle

echo 52
echo "#Sickle installed    52% Complete"
fi

#####-----#####
#####----- Fastqc -----#####
#####-----#####

FastqcInstalledPrep=$(fastqc 2>&1)

FastqcInstalled=$(case $FastqcInstalledPrep in
    "The program 'fastqc' is currently not installed.*" ) echo 1 ;;
esac)

if [[ "$FastqcInstalled" != 1 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the
program and any associated OSS that is not already installed)" ];then
echo 62
echo "#Fastqc is already installed    62% Complete"
sleep 1
else
echo 53
echo "#Installing Fastqc    53% Complete"
wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.5.zip

echo 54
echo "#Installing Fastqc    54% Complete"
sudo unzip ~/fastqc_v0.11.5.zip

#may need to cd into the fastqc file
echo 55
echo "#Installing Fastqc    55% Complete"
sudo chmod +x ~/FastQC/fastqc

echo 56
echo "#Installing Fastqc    56% Complete"
sudo ln -s ~/FastQC/fastqc /usr/local/bin/fastqc

echo 57
echo "#Installing Fastqc    57% Complete"
sudo apt install fastqc

echo 62
echo "#Fastqc installed    62% Complete"

```

fi

```
#####-----#####  
#####----- Velvet -----#####  
#####-----#####
```

```
VelvetInstalledPrep1=$(velvetg 2>&1)  
VelvetInstalledPrep2=$(velveth 2>&1)
```

```
VelvetInstalled1=$(case $VelvetInstalledPrep1 in  
  "The program 'vevletg' is currently not installed.*" ) echo 1 ;;  
esac)
```

```
VelvetInstalled2=$(case $VelvetInstalledPrep2 in  
  "The program 'velveth' is currently not installed.*" ) echo 1 ;;  
esac)
```

```
VelvetInstalled=$(( $VelvetInstalled1 + $VelvetInstalled2 ))
```

```
if [[ "$VelvetInstalled" != 2 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the  
program and any associated OSS that is not already installed)" ];then  
echo 72  
echo "#Velvet is already installed    72% Complete"  
sleep 1  
else
```

```
if [[ $VelvetInstalled1 = 1 ]];then  
echo "#Velvetg is not already installed    62% Complete"  
else  
echo "#Velvetg is already installed    62% Complete"  
fi
```

```
if [[ $VelvetInstalled2 = 1 ]];then  
echo "#Velveth is not already installed    62% Complete"  
else  
echo "#Velveth is already installed    62% Complete"  
fi
```

```
echo 63  
echo "#Installing Velvet    63% Complete"
```

```
#Velvet Install
```

```
wget https://github.com/dzerbino/velvet/archive/master.zip  
echo 65  
echo "#Installing Velvet    65% Complete"
```

```
unzip master.zip  
echo 67  
echo "#Installing Velvet    67% Complete"  
#cd into it  
cd ~/velvet-master/  
echo 68  
echo "#Installing Velvet    68% Complete"
```

```

sudo make
echo 72
echo "#Velvet installed    72% Complete"

fi

#####-----#####
#####----- QuastInstall -----#####
#####-----#####

QuastInstalledPrep=$(quast.py 2>&1)

QuastInstalled=$(case $QuastInstalledPrep in
  "QUAST: QUality ASsessment Tool for Genome Assemblies"* ) echo 1 ;;
esac)

if [[ "$QuastInstalled" = 1 ]] && [ "$InstallationType" = "Careful Complete Install (Installs the
program and any associated OSS that is not already installed)" ];then
echo 82
echo "#QUAST is already installed    82% Complete"
sleep 1
else
#QUAST install
echo 73
echo "#Installing QUAST    73% Complete"
#requirements
sudo apt-get install -y pkg-config libfreetype6-dev libpng-dev python-matplotlib

#Download QUAST source code and tarbell
echo 75
echo "#Installing QUAST    75% Complete"
wget https://downloads.sourceforge.net/project/quast/quast-4.4.tar.gz
echo 76
echo "#Installing QUAST    76% Complete"
tar -xzf ~/quast-4.4.tar.gz

echo 78
echo "#Installing QUAST    78% Complete"
cd ~/quast-4.4

sudo ./setup.py install_full

echo 82
echo "#QUAST installed    82% Complete"

fi

#####-----#####
#####----- PRICE install -----#####
#####-----#####

```

```

echo 83
echo "#Installing PRICE    83% Complete"
wget http://derisilab.ucsf.edu/software/price/PriceSource140408.tar.gz
echo 85
echo "#Installing PRICE    85% Complete"
tar -xzf PriceSource140408.tar.gz

echo 86
echo "#Installing PRICE    86% Complete"
cd ~/PriceSource14040

sudo make

echo 88
echo "#PRICE installed    88% Complete"
sed -i -e "s|PriceSource140408|PriceSource140408|g" ~/GGOSS/Scripts/PriceTItemplate.sh

#####-----#####
#####----- mothur install -----#####
#####-----#####

echo 88
echo "#Installing Mothur    88% Complete"
mkdir mothur
cd mothur
wget https://github.com/mothur/mothur/releases/download/v.1.41.1/Mothur.linux_64.zip
echo 89
echo "#Installing Mothur    89% Complete"
unzip Mothur.linux_64.zip
sudo make
echo 90
echo "#Mothur installed    90% Complete"

#####-----#####
#####----- Blast install -----#####
#####-----#####

echo 91
echo "#Installing Blast    91% Complete"
cd
mkdir BLAST
cd BLAST
wget ftp://ftp.ncbi.nih.gov/blast/executables/blast+/LATEST/ncbi-blast-2.8.1+-x64-linux.tar.gz
echo 92
echo "#Installing Blast    92% Complete"
tar -xzf ncbi-blast-2.8.1+-x64-linux.tar.gz
echo "[NCBI]
Data="/usr/local/pkg/ncbi-blast/data/"
[BLAST] BLASTDB=path_to_db" > .ncbirc

export BLASTDB=/usr/local/blastdb

echo 93

```

```
echo "#Blast installed    93% Complete"
```

```
#####-----#####  
#####----- Prokka install -----#####  
#####-----#####
```

```
echo 94  
echo "#Installing Prokka    94% Complete"  
sudo apt-get install libdatetime-perl libxml-simple-perl libdigest-md5-perl git default-jre bioperl  
echo 95  
echo "#Installing Prokka    95% Complete"  
sudo cpan Bio::Perl  
echo 96  
echo "#Installing Prokka    96% Complete"  
git clone https://github.com/tseemann/prokka.git $HOME/prokka  
$HOME/prokka/bin/prokka --setupdb  
echo 97  
echo "#Prokka Installed    97% Complete"
```

```
#####-----#####  
#####----- R install -----#####  
#####-----#####
```

```
echo 97  
echo "#Installing R... Updating system  97% Complete"  
sudo apt-get update  
echo 97  
echo "#Installing base R...  97% Complete"  
sudo apt-get install r-base  
echo 98  
echo "#Installing required R packages... vegan...  98% Complete"  
sudo apt-get build-dep r-cran-vegan  
echo 98  
echo "#Installing required R packages... gplots...  98% Complete"  
sudo apt-get build-dep r-cran-gplots  
echo 98  
echo "#Installing required R packages... ggbiplot... 98% Complete"  
sudo apt-get build-dep r-cran-ggbiplot  
echo 99  
echo "#Installing required R packages... lattice... 99% Complete"  
sudo apt-get build-dep r-cran-lattice  
echo 99  
echo "#Installing required R packages... gclus...  99% Complete"  
sudo apt-get build-dep r-cran-gclus  
echo 99  
echo "#Installing required R packages... ggplot2... 99% Complete"  
sudo apt-get build-dep r-cran-ggplot2  
echo 99  
echo "#Required R packages installed          99% Complete"  
sleep 0.8
```

```
else  
echo 20
```

```

echo "#GGOSS Install - Basic Install    20% Complete"
sleep 1

echo 30
echo "#GGOSS Install - Basic Install    30% Complete"
sleep 1

echo 40
echo "#GGOSS Install - Basic Install    40% Complete"
sleep 1

echo 50
echo "#GGOSS Install - Basic Install    50% Complete"
sleep 1

echo 60
echo "#GGOSS Install - Basic Install    60% Complete"
sleep 1

echo 70
echo "#GGOSS Install - Basic Install    70% Complete"
sleep 1

echo 80
echo "#GGOSS Install - Basic Install    80% Complete"
sleep 1

echo 90
echo "#GGOSS Install - Basic Install    90% Complete"
sleep 1

echo 99
echo "#GGOSS Install - Basic Install    99% Complete"
sleep 1

fi
echo 100
echo "#GGOSS Installation Complete    100% Complete"
)|
yad --progress --auto-close --auto-kill --center --width=700 --image=$ICON --image-on-top --
title="Installing GGOSS: |G|ui for |G|enomic analysis incorporating |O|pen |S|ource |S|oftware" \
--percentage=0

yad --title="GENOME SEQUENCING PROGRAM - GGOSS Installation" --text="GGOSS Installed:  Open GGOSS upon closing" --text-align=center --center --
height=100 --width=500 --wrap --size=fit --button="Yes":3 --button="No" --buttons-layout=center \

mode="$?"
case $mode in
    3)~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.2 GUI for genome sequencing program GGOSS

The following sub-sections contain all the scripts written in order to build the GGOSS GUI

10.9.2.1 GGOSS Main Menu code

```
#!/bin/bash
ToolSelected=2
if [ -f ~/GGOSS/tmp/CleanUpQC.txt ];then
rm ~/GGOSS/tmp/CleanUpQC.txt
fi
if [ -f ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt ];then
rm ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt
fi
if [ -f ~/GGOSS/tmp/PostAnnotation.txt ];then
rm ~/GGOSS/tmp/PostAnnotation.txt
fi
if [ -f ~/GGOSS/tmp/BacFungTaxa.txt ];then
rm ~/GGOSS/tmp/BacFungTaxa.txt
fi

ICON=~/GGOSS/Pictures/DNA_1.jpg
YADKEY=$(echo $[( $RANDOM % ($[10000 - 32000] + 1)) + 10000])
Clean_up_and_QC="Cutadapt
FastQC
Khmer
Sickle"
Assembly_Mapping_Annotation="Genome Assemblers
Annotation tools
MUMmer
PRICE
BLAST
QUAST
Ragout
BWA
SAMtools"
Post_Construction="Gepard
Mauve
Circos
CGview
GGOSS_ConservedSequenceFinder"
Community_Analysis="Mothur
PIPITS
MetaPhlAn
Community plotting
GGOSS Viral taxa finder"
OtherSelection="Edit path to tools
Stack tools
Mass file name manipulation
Primer creation
DNA/RNA conversion
Email upon completion"
```

```

echo "$Clean_up_and_QC" | yad --plug=$YADKEY --tabnum=1 --list --column="Select Clean-up or
Quality Check tool to use:" --multiple --width 800 --height 600 --center --align=center &>
~/GGOSS/tmp/CleanUpQC.txt & echo "$Assembly_Mapping_Annotation" | yad --plug=$YADKEY -
-tabnum=2 --list --column="Select Assembly or Annotation tool to use:" --multiple &>
~/GGOSS/tmp/Assembly_Mapping_Annotation.txt & echo "$Post_Construction" | yad --
plug=$YADKEY --tabnum=3 --list --column="Select Post Assembly and/or Annotation tool to use:" -
-multiple &> ~/GGOSS/tmp/PostAnnotation.txt & echo "$Community_Analysis" | yad --
plug=$YADKEY --tabnum=4 --list --column="Select Bacterial or Fungal community analysis tool to
use:" --multiple --width 800 --height 600 --center --align=center &> ~/GGOSS/tmp/BacFungTaxa.txt
& echo "$OtherSelection" | yad --plug=$YADKEY --tabnum=5 --list --column="Select GGOSS
option:" --multiple --width 800 --height 600 --center --align=center &> ~/GGOSS/tmp/StackTools.txt
& yad --notebook --title="
GENOME SEQUENCING
PROGRAM -- Main Menu
Created by Giles Holt" --skip-taskbar --image=$ICON
--image-on-top --width=1000 --height=800 --center --buttons-layout=spread --button="gtk-quit:10" --
button=" Import files ":3 --button=" Open ":2 --button=" Open Output
":4 --button=" GGOSS Manual ":2 --key=$YADKEY --tab=" Clean-up and QC "
--tab="Assembly, Alignment, Mapping and Annotation" --tab=" Post Construction Analysis " --
tab="Community Analysis" --tab=" Other options "

```

```
mode="$?"
```

```

case $mode in
2) ToolSelected=1 ;;
3)~/GGOSS/Scripts/File_Import.sh ;;
4)~/GGOSS/GenomicsProgram.sh & nautilus ~/GGOSS_InputOutput/ ;;
esac

```

```
if [ $ToolSelected = 1 ];then
```

```

if [ -f ~/GGOSS/tmp/CleanUpQC.txt ] && [ -s ~/GGOSS/tmp/CleanUpQC.txt ];then
MenuToOpen=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/CleanUpQC.txt)

```

```

if [ "$MenuToOpen" = "Cutadapt" ];then
~/GGOSS/Buttons/CutAdapter_Menu.sh
fi
if [ "$MenuToOpen" = "FastQC" ];then
~/GGOSS/Buttons/FastqcMenu.sh
fi
if [ "$MenuToOpen" = "Khmer" ];then
~/GGOSS/Buttons/ButtonKhmer.sh
fi
if [ "$MenuToOpen" = "Sickle" ];then
~/GGOSS/Buttons/SickleMenu.sh
fi

```

```
fi
```

```

if [ -f ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt ] && [ -s
~/GGOSS/tmp/Assembly_Mapping_Annotation.txt ];then
MenuToOpen=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt)
if [ "$MenuToOpen" = "Genome Assemblers" ];then
~/GGOSS/Buttons/ButtonGenomeAssembly.sh
fi
if [ "$MenuToOpen" = "Annotation tools" ];then
~/GGOSS/Buttons/ButtonAnnotation.sh
fi
if [ "$MenuToOpen" = "MUMmer" ];then
~/GGOSS/Buttons/ButtonMUMmer.sh

```

```

fi
if [ "$MenuToOpen" = "PRICE" ];then
~/GGOSS/Buttons/Button_PRICETI.sh
fi
if [ "$MenuToOpen" = "BLAST" ];then
~/GGOSS/Buttons/ButtonBlast.sh
fi
if [ "$MenuToOpen" = "QUAST" ];then
~/GGOSS/Buttons/QUASTMenu.sh
fi
if [ "$MenuToOpen" = "Ragout" ];then
~/GGOSS/Buttons/ButtonRagout.sh
fi
if [ "$MenuToOpen" = "BWA" ];then
~/GGOSS/Buttons/Button_BWA.sh
fi
fi

if [ -f ~/GGOSS/tmp/PostAnnotation.txt ] && [ -s ~/GGOSS/tmp/PostAnnotation.txt ];then
MenuToOpen=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/PostAnnotation.txt)
if [ "$MenuToOpen" = "GGOSS_ConservedSequenceFinder" ];then
~/GGOSS/Buttons/Button_ConservedGeneFinder.sh
fi

fi

if [ -f ~/GGOSS/tmp/BacFungTaxa.txt ] && [ -s ~/GGOSS/tmp/BacFungTaxa.txt ];then
MenuToOpen=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/BacFungTaxa.txt)
if [ "$MenuToOpen" = "Mothur" ];then
~/GGOSS/Buttons/Button16S_Analysis.sh
fi
if [ "$MenuToOpen" = "PIPITS" ];then
~/GGOSS/Buttons/PIPITS_Menu.sh
fi
if [ "$MenuToOpen" = "MetaPhlAn" ];then
~/GGOSS/Buttons/MetaPhlAn_Menu.sh
fi
if [ "$MenuToOpen" = "Community plotting" ];then
~/GGOSS/Buttons/CommunityPlotting_Menu.sh
fi
if [ "$MenuToOpen" = "GGOSS Viral taxa finder" ];then
~/GGOSS/Buttons/ButtonGGOSSViralTaxaFinder.sh
fi
fi

if [ -f ~/GGOSS/tmp/StackTools.txt ] && [ -s ~/GGOSS/tmp/StackTools.txt ];then
MenuToOpen=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/StackTools.txt)
if [ "$MenuToOpen" = "Edit path to tools" ];then
~/GGOSS/Buttons/ButtonToolPaths.sh
fi
if [ "$MenuToOpen" = "Stack tools" ];then
~/GGOSS/Buttons/StackToolsButtonAndRun.sh
fi
if [ "$MenuToOpen" = "Mass file name manipulation" ];then
~/GGOSS/Buttons/Button_MassFileChange.sh
fi

```

```
if [ "$MenuToOpen" = "Primer creation" ];then
~/GGOSS/Buttons/Button_PrimerDesign.sh
fi
if [ "$MenuToOpen" = "DNA/RNA conversion" ];then
~/GGOSS/Buttons/Button_DNA_RNA_Converter.sh
fi
fi

fi

if [ -f ~/GGOSS/tmp/CleanUpQC.txt ];then
rm ~/GGOSS/tmp/CleanUpQC.txt
fi
if [ -f ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt ];then
rm ~/GGOSS/tmp/Assembly_Mapping_Annotation.txt
fi
if [ -f ~/GGOSS/tmp/PostAnnotation.txt ];then
rm ~/GGOSS/tmp/PostAnnotation.txt
fi
if [ -f ~/GGOSS/tmp/BacFungTaxa.txt ];then
rm ~/GGOSS/tmp/BacFungTaxa.txt
fi
```

10.9.2.2 Genome assembly

10.9.2.2.1 Assembler selection main menu

```
#!/bin/bash

ICON=~/.GGOSS/Pictures/DNA4_900.jpg

yad --title="
Assembly          Created by Giles Holt" --width=800 --height=500 --center --image=$ICON --
image-on-top --size=fit --center --button="
Velvet            ":1 --button="
Previous          ":3 --buttons-layout=center
                                SPAdes          ":0 --button="
                                IDBA            ":2 --button="

mode="$?"
case $mode in
  0)~/GGOSS/Buttons/ButtonGenomeAssembly_SPAdes.sh ;;
  1)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
  2)~/GGOSS/Buttons/ButtonGenomeAssembly_IDBA.sh ;;
  3)~/GGOSS/GenomicsProgram.sh ;;
esac
```

10.9.2.2.2 SPAdes main menu

```
#!/bin/bash

ICON=~/.GGOSS/Pictures/SPAdes_900.jpg

yad --title="
SPAdes           Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --
center --button="Run Assembler":0 --button="Select samples":2 --button="SPAdes Settings":3 --
button="Previous":4 --buttons-layout=center

mode="$?"
case $mode in
  0)~/GGOSS/Scripts/SPAdesAssemblerPrimingScript.sh ;;
  2)~/GGOSS/Scripts/FileSelection/SPAdesfileselect.sh ;;
  3)~/GGOSS/SPAdes_settingsmenu.sh ;;
  4)~/GGOSS/Buttons/ButtonGenomeAssembly.sh ;;
esac
```

10.9.2.2.3 SPAdes settings menu

```
#!/bin/bash
```

```
ICON=~/.GGOSS/Pictures/SPAdes_900Cropped.jpg
```

```
yad --title="                                GENOME SEQUENCING PROGRAM --
SPAdes Settings      Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --
center --form \
--field="                                Data from machine type:" :CB \
--field="                                Read coverage cutoff value (optional):" \
--field="                                PHRED quality offset for input reads (optional):" \
--field="                                Specify K values (e.g. 21,33,55,77,99,127 etc):" \
--field="                                Number of threads (optional):" \
--field="                                Set Memory limit (gb) (optional):" \
--field="                                Library numeracy:" :CB \
--field="                                Single cell data:" :CB \
--field="                                Run read error correction only (optional):" :CB \
--field="                                Run assembly module only (optional):" :CB \
--field="Reduce mismatch and short indel number, and run MismatchCorrector (--careful)
(optional):" :CB \
--field="                                Trusted contigs (optional):" :CB \
--field="                                Untrusted contigs (optional):" :CB \
--field="                                Assemble with MetaSPAdes (optional):" :CB \
--field="                                Assemble plasmid (optional):" :CB \
--field="                                Assemble RNA-seq data (optional):" :CB \
"Illumina!pacBio!nanopore!sanger!iontorrent!" 'off' '-' '-' '-' '-' 'One Library!Multiple libraries!'
"No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/SPAdesSettingsChange.txt > ~/.GGOSS/tmp/SPAdesSettingsChange.txt

mode="$?"
case $mode in
0)~/.GGOSS/Buttons/SPAdes_settingsmenuLibTypeExtra.sh ;;
1)~/.GGOSS/Buttons/ButtonGenomeAssembly_SPAdes.sh ;;
esac
```

10.9.2.2.4 SPAdes file selection

```
#!/bin/sh

if [ -f ~/GGOSS/tmp/SPAdesSelectedFiles.txt ];then
rm ~/GGOSS/tmp/SPAdesSelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt ];then
rm ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="                GENOME SEQUENCING PROGRAM - SPAdes file
selection          Created by Giles Holt" --list --column="Select files you wish to run" --multiple --
width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --separator=" >
~/GGOSS/tmp/SPAdesSelectedFiles.txt

mode="$?"
case $mode in
    1)~/GGOSS/Buttons/ButtonGenomeAssembly_SPAdes.sh ;;
esac

    if [ -s ~/GGOSS/tmp/SPAdesSelectedFiles.txt ];then

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/SPAdesSelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="                GENOME SEQUENCING
PROGRAM - SPAdes file selection          Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt

mode="$?"
case $mode in
    2)~/GGOSS/Scripts/FileSelection/SPAdesfileselect.sh ;;
esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt ];then
~/GGOSS/Buttons/ButtonGenomeAssembly_SPAdes.sh
fi
```

10.9.2.2.5 Velvet main menu

```
#!/bin/bash
```

```
VelvetSettings=1
```

```
ICON=~/.GGOSS/Pictures/Velvet_900.jpg
```

```
yad --title="                                GENOME SEQUENCING PROGRAM --  
Velvet                                created by Giles Holt" --width=800 --height=500 --center --image=$ICON --  
image-on-top --size=fit --center --button="Run Velvet":0 --button="File Selection":2 --  
button="Velvet Settings":1 --button="Previous":3 --buttons-layout=center
```

```
mode="$?"
```

```
case $mode in
```

```
0)~/GGOSS/Scripts/VelvetAssembly_PrepScript.sh ;;
```

```
1)VelvetSettings=2 ;;
```

```
2)~/GGOSS/Buttons/VelvetFileSelection.sh ;;
```

```
3)~/GGOSS/Buttons/ButtonGenomeAssembly.sh ;;
```

```
esac
```

```
if [ "${VelvetSettings}" = 2 ];then
```

```
yad --title="                                GENOME SEQUENCING PROGRAM -- Velvet  
Settings Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \  
--field="Shuffle Forward and Reverse files:":CB \  
"Yes!No!" \  
--text-info --show-uri --width=600 --height=500 --center --wrap \  
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --  
filename=~/.GGOSS/tmp/VelvetSettings.txt > ~/.GGOSS/tmp/VelvetSettings.txt
```

```
mode="$?"
```

```
case $mode in
```

```
0)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
```

```
1)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
```

```
esac
```

```
fi
```

10.9.2.2.6 Velvet file selection

```
#!/bin/sh

echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

#this means in the case of script exit do 1 of the following
mode="$?"
case $mode in
  0)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
  1)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
  3)~/GGOSS/Buttons/ButtonGenomeAssembly_Velvet.sh ;;
  esac | \

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="          GENOME SEQUENCING PROGRAM
- Velvet --file selection          Created by Giles Holt" --list --column="Select files for Velvet Run" --
multiple --width 800 --height 600 --center --align=center --on-top --button="Save selection:0" --
button="Previous":3 --separator=" " --filename=~/GGOSS/tmp/VelvetSelectedFiles.txt >
~/GGOSS/tmp/VelvetSelectedFiles.txt
```

10.9.2.2.7 QUAST main menu, settings, file selection

```
#!/bin/bash

#####-----#####
##### QUAST MENU #####
#####-----#####

SettingsSelected=2
FileSelection=2

if [ -f ~/GGOSS/tmp/FileSelectionPrevious.txt ];then
FileSelectionPrevious=$(head -n1 ~/GGOSS/tmp/FileSelectionPrevious.txt)
fi

FileSelectionPrevious=$(( $FileSelectionPrevious + 1 ))
ICON=~/GGOSS/Pictures/QUASTMenu.jpg

if [ $FileSelectionPrevious != 2 ];then
FileSelectionPrevious=3

if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi

if [ -f ~/GGOSS/tmp/FileSelectionPrevious.txt ];then
rm ~/GGOSS/tmp/FileSelectionPrevious.txt
fi

yad --title="          GENOME SEQUENCING PROGRAM --
QUAST          Created by Giles Holt" --text="Quality ASsessment Tool
```

(QUAST).

The tool evaluates genome assemblies by computing various metrics. For instructions on QUAST, MetaQUAST, the extension for metagenomic datasets, and Icarus, interactive visualizer for these tools,

visit:<http://quast.bioinf.spbau.ru/manual.html>

Or GGOSS Manual sections." / --text-align=fill --center --image=\$ICON --size=fit --center --button="Previous":4 --button="Run":7 --button="Select samples":6 --button="Tabulate QUAST Output":8 --button="QUAST Settings":5 --button="QUAST Manual" --buttons-layout=center

```
mode="$?"
case $mode in
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)SettingsSelected=1 ;;
  6)FileSelection=1 ;;
  7)~/GGOSS/Scripts/QUASTScriptPrep.sh ;;
  8)~/GGOSS/Scripts/QUASTtabulate.sh ;;
esac
```

```
#####---- QUAST SETTINGS ----#####
```

```
if [ $SettingsSelected = 1 ];then
```

```
if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi
```

```
if [ -f ~/GGOSS/tmp/QUAST_SettingsChange.txt ]
then
rm ~/GGOSS/tmp/QUAST_SettingsChange.txt
fi
```

```
mode="$?"
case $mode in
  1)echo "1" > ~/GGOSS/tmp/OpenMenu.txt ;;
esac | \
```

```
yad --title="          GGOSS - QUAST -- Genome assembly quality assessment
Created by Giles Holt" --center --image-on-top --size=fit --center --form --columns=3 \
--field="          Reference genome:":CB \
--field="          Reference genome with gene positions:":CB \
--field="          Reference genome with operon positions:":CB \
--field="          Minimum contig threshold length:" \
```

```

--field="          Threads:" \
--field="          Gene finding":CB \
--field="          Gene finding tool":CB \
--field="          Gene Thresholds:" \
--field="          Genome type":CB \
--field="          Estimated reference genome size:" \
--field="          Gage mode":CB \
--field="          Contig length thresholds:" \
--field="          Assembly File Type":CB \
--field="          Use all alignments":CB \
--field="          Minimum alignment length:" \
--field="          Minimum IDY%:" \
--field="          Ambiguity usage":CB \
--field="          Ambiguity usage setting":CB \
--field="          Ambiguity score:" \
--field="          Break contigs at every misassembly":CB \
--field="          Lower threshold for misassembly relocation size:" \
--field="          Maximum scaffold gap size:" \
--field="          Lower threshold for partially unaligned contigs:" \
--field="          Reference genome is fragmented":CB \
--field="          Fragmented max indent:" \
--field="          File format for plots":CB \
--field="          Memory efficient":CB \
--field="          Space efficient: Create only primary output items":CB \
--field="          Silent mode":CB \
--field="          Structural variant calling and processing":CB \
--field="          Format of files aligned to reference genome":CB \
--field="          Effects speed: In-built input FASTA file check":CB \
--field="          Effects speed: Allow plot drawing":CB \
--field="          Effects speed: Allow HTML reports":CB \
--field="          Effects speed: Allow building of Icarus viewers":CB \
--field="          Effects speed: Allow SNPs statistics":CB \
--field="          Effects speed: Allow computation of GC%":CB \
--field="          Effects speed: Allow structural variant calling and processing":CB \
--field="          Effects speed: Allow large output file compression":CB \
--field="          MetaQUAST only: Use provided ref order in summary plots":CB \
--field="          MetaQUAST only: Reference Genome list :":CB \
--field="          MetaQUAST only: Use custom BLAST database":CB \
--field="          MetaQUAST only: Max number of reference genomes per assembly:" \
--field="          MetaQUAST only: Unique mapping:" \
"No!Yes!" "No!Yes!" "No!Yes!" '-' '-' "No!Yes!" "Glimmer!mgm!" '-' "Prokaryotic!Eukaryotic!" '-'
"No!Yes!" '-' "Contig!Scaffold!" "No!Yes!" '-' '-' "No!Yes!" "none!one!all" '-' "No!Yes!" '-' '-' '-'
"No!Yes!" '-' "Not applicable!emf!eps!pdf!png!ps!raw!rgba!svg!svgz!" "No!Yes!" "No!Yes!"
"No!Yes!" "No!Yes!" "Not applicable!bam!sam!BEDPE" "No!Yes!" "No!Yes!" "No!Yes!"
"No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" '-' '-' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~//GGOSS/tmp/QUAST_SettingsChange.txt >
~/GGOSS/tmp/QUAST_SettingsChange.txt

```

```
echo "1" > ~/GGOSS/tmp/OpenMenu.txt
```

```
fi
```

```
else
```

```

if [ -f ~/GGOSS/tmp/FileSelectionPrevious.txt ];then
rm ~/GGOSS/tmp/FileSelectionPrevious.txt
fi

if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi

FileSelection=1

fi

#####---- QUAST FILE SELECTION ----#####

if [ $FileSelection = 1 ];then

if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi

if [ -f ~/GGOSS/tmp/QUASTSelectedFiles.txt ]
then
rm ~/GGOSS/tmp/QUASTSelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/QUASTSelectedFilesP4.txt ]
then
rm ~/GGOSS/tmp/QUASTSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="          GENOME SEQUENCING PROGRAM - QUAST
sample selection          Created by Giles Holt" --list --column="Select the samples you wish to run" -
-multiple --width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --
separator=" > ~/GGOSS/tmp/QUASTSelectedFiles.txt

mode="$?"
  case $mode in
    1)echo "3" > ~/GGOSS/tmp/FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/OpenMenu.txt ;;
    esac

    if [ -s ~/GGOSS/tmp/QUASTSelectedFiles.txt ]
then
if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi

echo "Second Stage file selection opened"
FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/QUASTSelectedFiles.txt)

```

```

echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="
PROGRAM - QUAST file selection          GENOME SEQUENCING
Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/QUASTSelectedFilesP4.txt

mode="$?"
case $mode in
  2) echo "1" > ~/GGOSS/tmp/FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/OpenMenu.txt ;;
  esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/QUASTSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

if [ -s ~/GGOSS/tmp/QUASTSelectedFilesP4.txt ];then
echo "1" > ~/GGOSS/tmp/OpenMenu.txt
fi

fi

fi

#####---- Open QUAST Menu if its been set ----#####

if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/OpenMenu.txt)
fi
OpenMenu=$(( $OpenMenuPrep + 1 ))

if [ $OpenMenu = 2 ];then
~/GGOSS/Buttons/QUASTMenu.sh
fi

if [ -f ~/GGOSS/tmp/OpenMenu.txt ];then
rm ~/GGOSS/tmp/OpenMenu.txt
fi

```

10.9.2.3 File clean-up

10.9.2.3.1 Khmer main menu and settings

```
#!/bin/bash

Settings=1
KhmerMainMenu=1
Runkhmer=1

ICON=$HOME/GGOSS/Pictures/taxonomy2_1000.jpg

yad --title="                GGOSS - GENOMIC ANALYSIS PROGRAM                created
by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button="Run Khmer":0 --
button="Select Fastq File":3 --button="Select LIC File":5 --button="Peak selection":1 --
button="Khmer settings":4 --button="Previous":2 --buttons-layout=center

mode="$?"
case $mode in
  0)Runkhmer=2 ;;
  1)$HOME/GGOSS/Buttons/Khmer_Peak_settingsmenu.sh ;;
  2)$HOME/GGOSS/GenomicsProgram.sh ;;
  3)$HOME/GGOSS/Scripts/FileSelection/Khmerfileselect.sh ;;
  4)Settings=2 ;;
  5)$HOME/GGOSS/Scripts/FileSelection/Khmerkhfileselect.sh ;;
esac

if [[ $Settings = 2 ]];then

yad --title="                GGOSS - GENOMIC ANALYSIS PROGRAM --
Khmer Settings          Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --
center --form \
--field="Memory limit (e.g. if 14gb ram type 14e9)" \
--field="Number Of Processors:" \
--field="Run loading into counting":CB \
--field="Run abundance distribution":CB \
'-' '-' "No!Yes" "No!Yes" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="Previous":1 --button="gtk-save:0" --editable --
filename=$HOME/GGOSS/tmp/KhmerSettingsChange.txt >
$HOME/GGOSS/tmp/KhmerSettingsChange.txt

mode="$?"
case $mode in
  0)KhmerMainMenu=2 ;;
  1)KhmerMainMenu=2 ;;
esac
fi

if [[ $Runkhmer = 2 ]];then

#prepping the script based on settings
```

```

cp $HOME/GGOSS/Scripts/KhmerTemplate.sh $HOME/GGOSS/Scripts/Khmer.sh

#change Khmer according to settings
SettingsRamLimit=$(awk -F '[]' '{print $1}' $HOME/GGOSS/tmp/KhmerSettingsChange.txt)

if [ $SettingsRamLimit = "-" ];then
sed -i -e "s/RAM_Limit \+//g" $HOME/GGOSS/Scripts/Khmer.sh
else
sed -i -e "s/RAM_Limit\+/-M $SettingsRamLimit/g" $HOME/GGOSS/Scripts/Khmer.sh
fi

SettingsProcessorNumber=$(awk -F '[]' '{print $2}' $HOME/GGOSS/tmp/KhmerSettingsChange.txt)
if [ $SettingsProcessorNumber = "-" ];then
sed -i -e "s/Processor_Number \+//g" $HOME/GGOSS/Scripts/Khmer.sh
else
sed -i -e "s/Processor_Number\+/-N $SettingsProcessorNumber/g"
$HOME/GGOSS/Scripts/Khmer.sh
fi

SettingsProcessorNumber=$(awk -F '[]' '{print $1}' $HOME/GGOSS/tmp/KhmerSettingsChange.txt)

if [ $SettingsProcessorNumber = "-" ];then
sed -i -e "s/Processor_Number \+//g" $HOME/GGOSS/Scripts/Khmer.sh
else
sed -i -e "s/Processor_Number\+/-N $SettingsProcessorNumber/g"
$HOME/GGOSS/Scripts/Khmer.sh
fi

#sets the file path containing to those files
FilePath=$(awk 'NR==1 {print}' $HOME/GGOSS/tmp/Path2selectedFile.txt)

sed -i -e "s|FilePath|$FilePath|g" $HOME/GGOSS/Scripts/Khmer.sh

#Makes a file with settings laid out for GUI to show user whats running

#change settings tmp text into individual lines per setting
sed 's/|\n/g' $HOME/GGOSS/tmp/KhmerSettingsChange.txt >
$HOME/GGOSS/tmp/KhmerSettingsChangeList.txt

#Add specific text to start of each line
sed -i '1 s/^/Memory limit: /' $HOME/GGOSS/tmp/KhmerSettingsChangeList.txt
sed -i '2 s/^/Processor number: /' $HOME/GGOSS/tmp/KhmerSettingsChangeList.txt
sed -i '3 s/^/Run load into counting: /' $HOME/GGOSS/tmp/KhmerSettingsChangeList.txt
sed -i '4 s/^/Run abundance distribution: /' $HOME/GGOSS/tmp/KhmerSettingsChangeList.txt

#An opening yad window explaining khmer is starting under the settings they have chosen
Settings=$(cat $HOME/GGOSS/tmp/KhmerSettingsChangeList.txt)

yad --title="GILES -- Khmer - Running Khmer under the following settings" --width=400 --center --
sticky --on-top --no-buttons --no-escape --text="$Settings" --text-align=center --timeout=2

sleep 1
$HOME/GGOSS/Scripts/LoopKhmerScript.sh

```

```

fi

if [[ "$KhmerMainMenu" = 2 ]];then
$HOME/GGOSS/Buttons/ButtonKhmer.sh
fi

```

10.9.2.3.2 Khmer file selection type 1

```

#!/bin/sh

if [ -f ~/GGOSS/tmp/KhmerSelectedFiles.txt ];then
rm ~/GGOSS/tmp/KhmerSelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/KhmerSelectedFilesP4.txt ];then
rm ~/GGOSS/tmp/KhmerSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="                GENOME SEQUENCING PROGRAM - Khmer file
selection          Created by Giles Holt" --list --column="Select files you wish to run" --multiple --
width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --separator=" >
~/GGOSS/tmp/KhmerSelectedFiles.txt

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/ButtonKhmer.sh ;;
  esac

  if [ -s ~/GGOSS/tmp/KhmerSelectedFiles.txt ];then
FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/KhmerSelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="                GENOME SEQUENCING
PROGRAM - Khmer file selection          Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/KhmerSelectedFilesP4.txt

mode="$?"
case $mode in
  2)~/GGOSS/Scripts/FileSelection/Khmerfileselect.sh ;;
  esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/KhmerSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt
fi

if [ -s ~/GGOSS/tmp/KhmerSelectedFilesP4.txt ];then
~/GGOSS/Buttons/ButtonKhmer.sh
fi

```

10.9.2.3.3 Khmer file selection type 2

```
#!/bin/sh

ReturnToMenukh=1
if [ -f ~/GGOSS/tmp/KhmerkhSelectedFiles.txt ];then
rm ~/GGOSS/tmp/KhmerkhSelectedFiles.txt
fi
if [ -f ~/GGOSS/tmp/KhmerkhSelectedFilesP4.txt ];then
rm ~/GGOSS/tmp/KhmerkhSelectedFilesP4.txt
fi

SelectFilekh=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFilekh" | yad --title="          GENOME SEQUENCING PROGRAM - Khmer
file selection          Created by Giles Holt" --list --column="Select files you wish to run" --multiple -
-width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --separator="
> ~/GGOSS/tmp/KhmerkhSelectedFiles.txt

mode="$?"
case $mode in
  1)ReturnToMenukh=2 ;;
esac

if [ -s ~/GGOSS/tmp/KhmerkhSelectedFiles.txt ];then

FolderSelectedkh=$(awk 'NR=1 {print}' ~/GGOSS/tmp/KhmerkhSelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelectedkh" > ~/GGOSS/tmp/Path2selectedFilekh.txt

SelectFilekh2=$(ls ~/GGOSS_InputOutput/"$FolderSelectedkh")

echo "$SelectFilekh2" | yad --title="          GENOME SEQUENCING
PROGRAM - Khmer file selection          Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/KhmerkhSelectedFilesP4.txt

mode="$?"
case $mode in
  2)~/GGOSS/Scripts/FileSelection/Khmerkhfileselect.sh ;;
esac

FolderSelectedkh2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/KhmerkhSelectedFilesP4.txt)
echo -n "$FolderSelectedkh2" >> ~/GGOSS/tmp/Path2selectedFilekh.txt

fi
if [ $ReturnToMenukh = 2 ];then
~/GGOSS/Buttons/ButtonKhmer.sh
fi

if [ -s ~/GGOSS/tmp/KhmerkhSelectedFilesP4.txt ];then
~/GGOSS/Buttons/ButtonKhmer.sh
fi
```

10.9.2.3.4 Cutadapt main menu, settings and file selection

```
#!/bin/bash

#####-----#####
##### CutAdapt MENU #####
#####-----#####

CutAdapterFileSelection=2
AdapterCuttingSettings=2
TrimReadLengthSettings=2
RunTheChosen=2
Cutadapt_PreSavedSettingsSelected=2
DontUseAPresavedSetting=2

ICON=~/.GGOSS/Pictures/AdapterTrimming.jpg

yad --title="                               GENOME SEQUENCING PROGRAM --
Cutadapt          Created by Giles Holt" --text="          Cutadapt

Can be used to search and
remove adapters, but can
also be used to modify
and filter reads and to
redirect them to various
output files" / --center --image=$ICON --size=fit --center --button="Previous":4 --button="Select
samples":6 --button="Run":18 --button="Run CutAdapt:Nextera XT":21 --button="Trim Read
Length":9 --button="Settings":5 --button="Use Pre-saved Settings":22 --button="Manual" --buttons-
layout=center

mode="$?"
case $mode in
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)AdapterCuttingSettings=1 ;;
  6)CutAdapterFileSelection=1 ;;
  7)~/GGOSS/Scripts/AdapterInput.sh ;;
  9)TrimReadLengthSettings=1 ;;
  21)~/GGOSS/Scripts/Cutadapt_PairedEnd_Nextera.sh ;;
  18)RunTheChosen=1 ;;
  22)Cutadapt_PreSavedSettingsSelected=1 ;;
esac

#####---- Cutadapt Pre-saved SETTINGS ----#####

if [ $Cutadapt_PreSavedSettingsSelected = 1 ];then
  if [ -f ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
    rm ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt
```

fi

```
ls ~/GGOSS/tmp/PreSavedSettings/Cutadapt | yad --title="GENOME SEQUENCING PROGRAM --
Cutadapt -- Pre-saved setting options          GGOSS - Created by Giles Holt" --list --
column="Select pre-saved setting to use" --width 800 --height 600 --center --align=center --button="
Previous      ":10 --button="      Select setting type      ":1 --buttons-layout=center --text="Select
" --separator=" > ~/GGOSS/tmp/CutadaptpresavedSettingSelected.txt
```

```
mode="$?"
```

```
case $mode in
```

```
10)echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt && DontUseAPresavedSetting=1 ;;
```

```
1)echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
```

```
esac
```

```
if [ "$DontUseAPresavedSetting" != "1" ];then
```

```
#take the saved settings file name
```

```
PreSavedFileName=$(awk 'NR==1 {print}' ~/GGOSS/tmp/CutadaptpresavedSettingSelected.txt)
```

```
#copy it and rename copy to the settings file used by the run script
```

```
cp ~/GGOSS/tmp/PreSavedSettings/Cutadapt/${PreSavedFileName}
```

```
~/GGOSS/tmp/Cutadapt_SettingsChange.txt
```

```
fi
```

```
fi
```

```
#####---- CutAdapt AdapterCuttingSettings ----#####
```

```
if [ $AdapterCuttingSettings = 1 ];then
```

```
Path=~/
```

```
if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
```

```
rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
```

```
fi
```

```
if [ -f ~/GGOSS/tmp/Cutadapt_SettingsChange.txt ]
```

```
then
```

```
rm ~/GGOSS/tmp/Cutadapt_SettingsChange.txt
```

```
fi
```

```
ICON=~/GGOSS/Pictures/AdapterTrimming.jpg
```

```
mode="$?"
```

```
case $mode in
```

```
1)echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
```

```
esac | \
```

```
yad --title="          GGOSS - CutAdapter -- Base trimming          Created by
Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Adapter type:" :CB \
--field="          Bases of 3' Adapter:" \
--field="          Bases of 5' Adapter:" \
--field=" Use Adapter list (write file including path, e.g. ${Path}AdapterList.txt):" \
--field="          Run Reverse compliment of adapters as well:" :CB \
```

```

"3'adapter!5'adapter!Anchored 3'adapter!Anchored 5' adapter!5' or 3'!Linked adapter!Paired end
adapter!" '-' '-' "Yes!No" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
    --button="gtk-save:0" --button="Previous":1 --button="Re-use":3 --button="Reset":2 --
button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/Cutadapt_SettingsChange.txt > ~/.GGOSS/tmp/Cutadapt_SettingsChange.txt

```

```

mode="$?"
case $mode in
  0)echo "1" > ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
  1)echo "1" > ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
  2)echo "1" > ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
  3)echo "1" > ~/.GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
esac

```

```

if [ -s ~/.GGOSS/tmp/Cutadapt_SettingsChange.txt ];then
yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM -- Cutadapt Settings          GGOSS
- Created by Giles Holt" --on-top --center --size=fit --center --text "If you wish the group of settings
chosen to be saved as a selectable pre-saved option in the future please provide a name for the setting
choices and select the 'Make a pre-saved selectable' button. Otherwise just select ignore" --form \
--field="Save setting selection as:" \
'\ \
--text-info --show-uri --center --wrap \
--button="Make a pre-saved selectable":0 --button="Ignore":2 --editable --
filename=~/.GGOSS/tmp/Cutadapt_SettingsPreSavetmp.txt >
~/.GGOSS/tmp/Cutadapt_SettingsPreSavetmp.txt

```

```

mode="$?"
case $mode in
  0)PreSavedSelectableChosen=1 ;;
  2)PreSavedSelectableChosen=2 ;;
esac

```

```

PreSavedSettingName=$(awk -F '|' 'NR==1 {print $1}'
~/.GGOSS/tmp/Cutadapt_SettingsPreSavetmp.txt)
echo "PreSavedSettingName: $PreSavedSettingName"
if [ "$PreSavedSelectableChosen" = "1" ];then
cp ~/.GGOSS/tmp/Cutadapt_SettingsChange.txt
~/.GGOSS/tmp/PreSavedSettings/Cutadapt/${PreSavedSettingName}
fi

```

fi

fi

Run command

```

if [ $RunTheChosen = 1 ];then
echo "triggered 1st if run command"
if [ -s ~/.GGOSS/tmp/Cutadapt_SettingsChange.txt ];then
echo "triggered 2nd if run command"
TrimSettingRun=$(awk -F "|" '{print $1}' ~/.GGOSS/tmp/Cutadapt_SettingsChange.txt)
if [ $TrimSettingRun == "Start" ] || [ $TrimSettingRun == "End" ];then
echo "triggered 3rd if run command"

```

```

~/GGOSS/Scripts/cutadapter_TrimReadLength.sh
fi

#make an if for whether input has been put in the adapter list file, this will decide whats run
AdapterListPresent=$(awk -F "|" '{print $4}' ~/GGOSS/tmp/Cutadapt_SettingsChange.txt | awk '{
print length }')
if [ $AdapterListPresent = 0 ];then
  AdapterCutTypeSelected=$(awk -F "|" '{print $1}' ~/GGOSS/tmp/Cutadapt_SettingsChange.txt)

  if [ "$AdapterCutTypeSelected" == "Paired end adapter" ]
  then
    ~/GGOSS/Scripts/cutadapter_PairedEnd.sh
  fi

  else
    ~/GGOSS/Scripts/CutAdapt_CustomAdapterListRun.sh
  echo "triggered CustomRun"
  fi

  else
  echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
  fi

fi

#####---- CutAdapt TrimReadLengthSettings ----#####

if [ $TrimReadLengthSettings = 1 ];then

  if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
  rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
  fi

mode="$?"
  case $mode in
    1)echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
  esac | \

yad --title="          GGOSS - CutAdapter -- Base trimming          Created by
Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Trim from:"CB \
--field="          Number of bases to trim:" \
"Start!End!" '50' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/GGOSS/tmp/Cutadapt_SettingsChange.txt >
~/GGOSS/tmp/Cutadapt_SettingsChange.txt

fi

#####---- CutAdapt file selection ----#####

```

```

if [ $CutAdapterFileSelection = 1 ];then

    if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
        rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
    fi

    if [ -f ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt ];then
        rm ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt
    fi

echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="                GENOME SEQUENCING PROGRAM
- CutAdapter --file selection          Created by Giles Holt" --list --column="Select files to cut the
adapters from" --multiple --width 800 --height 600 --center --align=center --on-top --button="OK" --
button="Previous":1 --separator=" " --filename=~/GGOSS/tmp/Cutadapt_SelectedFiles.txt >
~/GGOSS/tmp/Cutadapt_SelectedFiles.txt

#this means in the case of script exit do 1 of the following
mode="$?"
    case $mode in
        1)echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ;;
    esac

if [ -s ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt ]
then
    if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
        rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
    fi
echo "1" > ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
fi

fi

#####---- CutAdapt return to/Open menu after selections ----#####

if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
CutAdapter_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/CutAdapter_OpenMenu.txt)
fi
CutAdapter_OpenMenu=$(( $CutAdapter_OpenMenuPrep + 1 ))

if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
fi

if [ $CutAdapter_OpenMenu = 2 ];then
~/GGOSS/Buttons/CutAdapter_Menu.sh
fi

if [ -f ~/GGOSS/tmp/CutAdapter_OpenMenu.txt ];then
rm ~/GGOSS/tmp/CutAdapter_OpenMenu.txt
fi

```

10.9.2.3.5 Cutadapt Adapter input menu

```
#!/bin/sh
```

```
mode="$?"
case $mode in
  3)~/GGOSS/Buttons/CutAdapter_Menu.sh ;;
esac | \

yad --title="          GENOME SEQUENCING PROGRAM - CutAdapter --Adapter Input
Created by Giles Holt" --form --field="Adapter (just the bases e.g. ATGC)" --width 800 --height 600 -
-center --align=center --on-top --button="Save selection:0" --button="Previous":3 --separator=" --
filename=~//GGOSS/tmp/AdapterInput.txt > ~/GGOSS/tmp/AdapterInput.txt &
```

```
ICON=~//GGOSS/Pictures/AdapterTrimming.jpg
```

```
yad --title="          GENOME SEQUENCING PROGRAM --
CutAdapter          Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit
--center --button="Select samples":6 --button="Input Adapter":7 --button="Import Fastq Files":8 --
button="Run Types ->":5 --buttons-layout=center
```

```
mode="$?"
case $mode in
  5)~/GGOSS/Buttons/CutAdapter_Menu2.sh ;;
  6)~/GGOSS/Scripts/CutAdapterFileSelection.sh ;;
  7)~/GGOSS/Scripts/AdapterInput.sh ;;
  8)~/GGOSS/Scripts/cutadapter_File_Import.sh ;;
esac
```

10.9.2.3.6 Sickle main menu, settings, and file selection

```
#!/bin/bash
SettingsSelected=2
FileSelection=2
```

```
ICON=~//GGOSS/Pictures/SickleMenu.jpg
```

```
yad --title="          GENOME SEQUENCING PROGRAM -- Sickle
Created by Giles Holt" --text="          Sickle
```

A windowed adaptive FASTQ
trimming tool using quality.

sickle addresses deteriorating
quality towards prime ends.

This can negatively impact

```
bioinformatics analyses." / --text-align=fill --center --image=$ICON --size=fit --center --button="
Previous ":4 --button=" Run ":3 --button=" Select samples ":6 --button="Trim low quality
reads":5 --button=" Sickle Manual " --buttons-layout=center
```

```

mode="$?"
case $mode in
  3)~/GGOSS/Scripts/SicklePrepRun.sh ;;
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)SettingsSelected=1 ;;
  6)FileSelection=1 ;;
esac

if [ $SettingsSelected = 1 ];then

if [ -f ~/GGOSS/tmp/Sickle_SettingsChange.txt ]; then
rm ~/GGOSS/tmp/Sickle_SettingsChange.txt
fi

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/SickleMenu.sh ;;
esac | \

yad --title="          GGOSS - Sickle -- Low quality read trimming          Created
by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="          Quality type:":CB \
--field="          End type:":CB \
--field="          Length Threshold (optional):" \
--field="          Quality Threshold (optional):" \
--field="          Input read type:":CB \
--field="If interleaved selected: fastq file with any discarded read written to output file as a single
N:":CB \
"illumina!sanger!solexa!" "Single End!Paired End!" '- ' '- "Separate!Interleaved!" "No!Yes!" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --editable
--filename=~/GGOSS/tmp/Sickle_SettingsChange.txt > ~/GGOSS/tmp/Sickle_SettingsChange.txt

~/GGOSS/Buttons/SickleMenu.sh

fi

if [ $FileSelection = 1 ];then

echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

#this means in the case of script exit do 1 of the following
mode="$?"
case $mode in
  3)~/GGOSS/Buttons/SickleMenu.sh ;;
esac | \

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="          GENOME SEQUENCING PROGRAM
- Sickle --file selection          Created by Giles Holt" --list --column="Select files for the trimming
of low quality read" --multiple --width 800 --height 600 --center --align=center --on-top --
button="Save selection:0" --button="Previous":3 --separator=" --
filename=~/GGOSS/tmp/Sickle_SelectedFiles.txt > ~/GGOSS/tmp/Sickle_SelectedFiles.txt &

~/GGOSS/Buttons/SickleMenu.sh

fi

```

10.9.2.3.7 FastQC Main menu

```
#!/bin/bash
```

```
FileSelection=2
```

```
ICON=~/.GGOSS/Pictures/FastqcMenu.jpg
```

```
yad --title="          GENOME SEQUENCING PROGRAM -- FastQC    Created by Giles  
Holt" --text="          FastQC
```

```
Quality control check on
```

```
raw sequence data. A
```

```
modular set of analyses
```

```
providing a quick
```

```
impression of data problems
```

```
important before further
```

```
analysis." / --center --image=$ICON --size=fit --center --button="Previous":4 --button="Select  
samples":6 --button="Run FastQC":5 --button="FastQC Manual" --buttons-layout=center
```

```
mode="$?"
```

```
case $mode in
```

```
4)~/GGOSS/GenomicsProgram.sh ;;
```

```
5)~/GGOSS/Scripts/FastqcScriptPrep.sh ;;
```

```
6)FileSelection=1 ;;
```

```
esac
```

```
if [ $FileSelection = 1 ];then
```

```
echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt
```

```
cat ~/GGOSS/tmp/SelectFile.txt | yad --title="          GENOME SEQUENCING PROGRAM  
- Fastqc --file selection    Created by Giles Holt" --list --column="Select files for fastqc run" --  
multiple --width 800 --height 600 --center --align=center --on-top --button="Save selection:0" --  
button="Previous" --separator=" " --filename=~/.GGOSS/tmp/Fastqc_SelectedFiles.txt >  
~/GGOSS/tmp/Fastqc_SelectedFiles.txt
```

```
~/GGOSS/Buttons/FastqcMenu.sh
```

```
fi
```

10.9.2.4 Mapping tools

10.9.2.4.1 PRICE main menu, settings, file selection

```
#!/bin/bash

#####-----#####
##### PRICE MENU #####
#####-----#####

if [ -f ~/GGOSS/tmp/PRICE_OpenMenu.txt ];then
rm ~/GGOSS/tmp/PRICE_OpenMenu.txt
fi

if [ -f ~/GGOSS/tmp/PRICE_FilesSaved.txt ];then
rm ~/GGOSS/tmp/PRICE_FilesSaved.txt
fi
#ensures all variable have a number to prevent errors
PRICE_SettingsSelected_Read=2
PRICE_SettingsSelected_Filter=2
PRICE_SettingsSelected_FilterOther=2
PRICE_SettingsSelected_Parameter=2

ReOpenScripts=2

ICON=~/.GGOSS/Pictures/PRICETI1.jpg

if [ ! -f ~/GGOSS/tmp/PRICE_FileSelectionMenu.txt ] && [ ! -f
~/GGOSS/tmp/PRICE_SettingsMenu.txt ];then

yad --title="                                GENOME SEQUENCING PROGRAM -
PRICETI                                created by Giles Holt" --text="                                PRICE

(Paired-Read Iterative Contig Extension)

a de novo genome assembler. PRICE uses
paired-read information to iteratively
increase the size of existing contigs.

Initially, those contigs can be individual
reads from a subset of the paired-read
dataset, non-paired reads from sequencing
technologies that provide non-paired data,
or contigs that were output from a
```

```
prior run of PRICE or any other assembler." / --width=1000 --height=500 --center --image=$ICON --
size=fit --center --button=" Previous ":3 --button=" Run PRICE ":7 --button=" File
Selection ":5 --button=" PRICE Settings Menu ":4 --button=" PRICE Manual " --buttons-
layout=center
```

```
mode="$?"
case $mode in
  7)~/GGOSS/Scripts/price.sh ;;
  5)echo "1" > ~/GGOSS/tmp/PRICE_FileSelectionMenu.txt ;;
  3)~/GGOSS/GenomicsProgram.sh ;;
  4)echo "1" > ~/GGOSS/tmp/PRICE_SettingsMenu.txt ;;
esac
```

```
fi
```

```
#####---- PRICE SETTINGS MENU ----#####
```

```
if [ -f ~/GGOSS/tmp/PRICE_SettingsMenu.txt ];then
yad --title=" GENOME SEQUENCING PROGRAM -
PRICETI created by Giles Holt" --width=800 --height=500 --center --image=$ICON
--size=fit --center --button="Previous":7 --button="PRICE Input/Output Settings":1 --button="PRICE
Parameter Settings":6 --button="PRICE Filter Read Settings":4 --button="PRICE Filter Contig
Settings":5 --buttons-layout=center
```

```
mode="$?"
case $mode in
  7)echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt ;;
  1)PRICE_SettingsSelected_Read=1 ;;
  4)PRICE_SettingsSelected_Filter=1 ;;
  5)PRICE_SettingsSelected_FilterOther=1 ;;
  6)PRICE_SettingsSelected_Parameter=1 ;;
esac
```

```
#going back to main menu without having selected anything so removing any files that could detract
from opening the main menu
```

```
if [ -f ~/GGOSS/tmp/PRICE_OpenMenu.txt ];then
  if [ -f ~/GGOSS/tmp/PRICE_SettingsMenu.txt ];then
    rm ~/GGOSS/tmp/PRICE_SettingsMenu.txt
  fi
fi
```

```
#####---- PRICE SETTINGS ----#####
```

```
#### Read
```

```
if [ $PRICE_SettingsSelected_Read = 1 ];then
```

```
mode="$?"
case $mode in
  0)echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt ;;
  1)echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt ;;
esac | \
```



```

yad --title="
                                GGOSS - GENOMIC ANALYSIS PROGRAM --
PRICETI Settings      Created by Giles Holt" --center --size=fit --center --form \
--field="Filtering Reads: rqf ":"CB \
--field="rqf: % of nucleotides in read that must be high quality:" \
--field="rqf: Min allowed probability of a nucleotide being correct:" \
--field="rqf: Cycles passed:" \
--field="rqf: Number of cycles to run for:" \
--field="Filtering Reads: rnf ":"CB \
--field="rnf: % of nucleotides in a read that must be called:" \
--field="rnf: Cycles passed:" \
--field="rnf: Number of cycles to run for:" \
--field="maxHP: Filtering Reads: filter read pair if either nucleotide length has homo-polymer track
>:" \
--field="maxDi: Filtering Reads: filter read pair if either nucleotide length has repeating di-
nucleotide track >:" \
--field="Filtering reads: badf":CB \
--field="badf: File path and name:" \
--field="badf: Min ungapped % identity match to the above file before being prevented from being
mapped to contigs:" \
--field="Filtering reads: repmask":CB \
--field="repmask: Cycle number at which repeats will be detected:" \
--field="repmask: Repeats sought at the start or End of the cycle":CB \
--field="repmask: Min num. of variance units > median that will be counted as high-coverage:" \
--field="repmask: Min fold increase in coverage > median that will be counted as high-coverage:" \
--field="repmask: Min size in nt for a detected repeat:" \
--field="repmask: Min %identity match to a repeat for read to not be mapped to contigs:" \
--field="repmask: Output file to which the detected repeats will be written":CB \
--field="Filtering reads: reset":CB \
--field="reset: List the cycles to be reset (e.g. 1,3,5,7):" \
"No!Yes!" ' ' ' ' ' ' "No!Yes!" ' ' ' ' ' ' "No!Yes!" ' ' ' "No!Yes!" ' "Start!End!" ' ' ' ' ' ' "Not
applicable!fasta!priceq!" "No!Yes!" ' ' \
--text-info --show-uri --width=800 --height=500 --center --wrap \
--button="Previous":1 --button="gtk-save:0" --editable --
filename=~/.GGOSS/tmp/PRICETI_FilterSettingsChange.txt >
~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt
fi

```

```
##### Filter other
```

```
if [ $PRICE_SettingsSelected_FilterOther = 1 ];then
```

```

mode="$?"
case $mode in
  0)echo "1" > ~/.GGOSS/tmp/PRICE_OpenMenu.txt ;;
  1)echo "1" > ~/.GGOSS/tmp/PRICE_OpenMenu.txt ;;
esac | \

```

```

yad --title="
                                GGOSS - GENOMIC ANALYSIS PROGRAM --
PRICETI Settings      Created by Giles Holt" --center --size=fit --center --form --columns=2 \
--field="Filtering/Processing Assembled Contigs: lenf":CB \
--field="lenf: Filter contigs at the end of every cycle shorter than (nt's):" \
--field="lenf: Number of cycles to skip before filtering contigs:" \
--field="Filtering/Processing Assembled Contigs: trim":CB \

```

```

--field="trim: Run on the end of cycle number:" \
--field="trim: Min coverage level for trimming edges of contigs:" \
--field="trim: After trimming delete contigs shorter than:" \
--field="Filtering/Processing Assembled Contigs: trimB":CB \
--field="trimB: Number of cycles to skip:" \
--field="trimB: Min coverage level for trimming edges of contigs at end of every cycle:" \
--field="trimB: After trimming delete contigs shorter than:" \
--field="Filtering/Processing Assembled Contigs: trimI":CB \
--field="trimI: Minimal coverage level to trim to:" \
--field="trimI: After trimming delete contigs shorter than:" \
--field="Filtering/Processing Assembled Contigs: target":CB \
--field="target: % identity to an input initial contig to count as a match (ungapped):" \
--field="target: Num. cycles to skip before applying this filter:" \
--field="target: While target filtering, filtered/-unfiltered cycles will alternate
(No.Filtered,No.Unfiltered):" \
--field="Filtering/Processing Assembled Contigs: targetF":CB \
--field="targetF: % identity to an input initial contig to count as a match (ungapped):" \
--field="targetF: Num. cycles to skip before applying this filter:" \
--field="targetF: While target filtering, filtered/-unfiltered cycles will alternate
(No.Filtered,No.Unfiltered):" \
--field="Filtering Initial Contigs:icbf":CB \
--field="icbf: Sequence file and path:" \
--field="icbf: Min % identity to sequence in file:" \
--field="icbf: Don't apply filter to sequences >:" \
--field="Filtering Initial Contigs:icmHp":CB \
--field="icmHp: Filter out initial contig if its homo-polymer track nucleotide length is >:" \
--field="icmHp: Don't apply this filter to sequences >:" \
--field="Filtering Initial Contigs:icmDi":CB \
--field="icmDi: filter out initial contig if its repeating di-nucleotide track nucleotide length is >:" \
--field="icmDi: Don't apply this filter to sequences >:" \
--field="Filtering Initial Contigs:icqf":CB \
--field="icqf: % of nucleotides in a read that must be high quality:" \
--field="icqf: Min allowed probability of a nucleotide being correct:" \
--field="icqf: Don't apply this filter to sequences >:" \
--field="Filtering Initial Contigs:icnf":CB \
--field="icnf: % nucleotides in a read that must be called:" \
--field="icnf: Don't apply this filter to sequences >:" \
"No!Yes!" ' ' ' "No!Yes!" ' ' '
' ' "No!Yes!" ' ' ' "No!Yes!" ' ' ' "No!Yes!" ' ' ' "No!Yes!" ' ' ' "No!Yes!" ' ' ' "No!Yes!" ' ' '
--text-info --show-uri --width=800 --height=500 --center --wrap \
--button="Previous":1 --button="gtk-save:0" --editable --
filename=~/.GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt >
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt

```

fi

```
##### Parameter
```

```
if [ $PRICE_SettingsSelected_Parameter = 1 ];then
```

```
mode="$?"
```

```
case $mode in
```

```
0)echo "1" > ~/.GGOSS/tmp/PRICE_OpenMenu.txt ;;
```

```
1)echo "1" > ~/.GGOSS/tmp/PRICE_OpenMenu.txt ;;
```



```

0)echo "PRICETISelectedFiles.txt|PRICETISelectedFilesP4.txt|Path2selectedFile.txt|1" >
~/GGOSS/tmp/SelectedFilesForWhichInput.txt ;;
1)echo
"PRICETIAdditionalInput1SelectedFiles.txt|PRICETIAdditionalInput1SelectedFilesP4.txt|Additional
Input1Path2selectedFile.txt|1" > ~/GGOSS/tmp/SelectedFilesForWhichInput.txt ;;
2)echo
"PRICETIAdditionalInput2SelectedFiles.txt|PRICETIAdditionalInput2SelectedFilesP4.txt|Additional
Input2Path2selectedFile.txt|1" > ~/GGOSS/tmp/SelectedFilesForWhichInput.txt ;;
3)echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt ;;
esac

#gets rid of file that skips to file selection menu, so when script re-runs it opens PRICE main menu
if [ -f ~/GGOSS/tmp/PRICE_OpenMenu.txt ];then
rm ~/GGOSS/tmp/PRICE_FileSelectionMenu.txt
fi

fi

if [ -f ~/GGOSS/tmp/SelectedFilesForWhichInput.txt ];then
#changes path and file variable names according to type selected
PRICESelectedFiles=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/SelectedFilesForWhichInput.txt)

PRICESelectedFilesP4=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/SelectedFilesForWhichInput.txt)

Path2selectedFile=$(awk -F '|' '{print $3}' ~/GGOSS/tmp/SelectedFilesForWhichInput.txt)

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="          GENOME SEQUENCING PROGRAM - PRICETI
file Input selection          Created by Giles Holt" --list --column="Select files you wish to run" --
multiple --width 800 --height 600 --center --align=center --button="Open" --button="Previous":3 --
separator=" > ~/GGOSS/tmp/${PRICESelectedFiles}

mode="$?"
case $mode in
3)echo "1" > ~/GGOSS/tmp/PRICE_FileSelectionPrevious.txt && echo "1" >
~/GGOSS/tmp/PRICE_OpenMenu.txt ;;
esac

if [ -f ~/GGOSS/tmp/PRICE_FileSelectionPrevious.txt ];then
rm ~/GGOSS/tmp/SelectedFilesForWhichInput.txt
rm ~/GGOSS/tmp/PRICE_FileSelectionPrevious.txt
fi

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/${PRICESelectedFiles})
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/${Path2selectedFile}
fi

if [ -f ~/GGOSS/tmp/SelectedFilesForWhichInput.txt ];then
SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")
echo "$SelectFile2" | yad --title="          GENOME SEQUENCING
PROGRAM - PRICETI file Input selection          Created by Giles Holt" --list --column="Select
files you wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/${PRICESelectedFilesP4}

```

```

mode="$?"
case $mode in
    2)echo "1" > ~/GGOSS/tmp/PRICE_FilesSaved.txt ;;
    esac

if [ ! -f ~/GGOSS/tmp/PRICE_FilesSaved.txt ];then
rm ~/GGOSS/tmp/PRICE_FilesSaved.txt
rm ~/GGOSS/tmp/SelectedFilesForWhichInput.txt
rm ~/GGOSS/tmp/PRICE_FileSelectionMenu.txt
rm ~/GGOSS/tmp/PRICE_SettingsMenu.txt
echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt
else
rm ~/GGOSS/tmp/PRICE_FilesSaved.txt
rm ~/GGOSS/tmp/PRICE_SettingsMenu.txt
echo "1" > ~/GGOSS/tmp/PRICE_OpenMenu.txt
fi

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/${PRICESelectedFilesP4})
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/${Path2selectedFile}

fi

```

#####---- PRICE return to/Open menu after selections ----#####

```

if [ -f ~/GGOSS/tmp/PRICE_OpenMenu.txt ];then
ReOpenScripts=1
rm ~/GGOSS/tmp/PRICE_OpenMenu.txt
fi

if [ $ReOpenScripts = 1 ];then
~/GGOSS/Buttons/Button_PRICETI.sh
fi

```

10.9.2.4.2 File selection menu for additional file selection options

```
#!/bin/bash
```

```

yad --title="
PRICETI
GENOME SEQUENCING PROGRAM -
created by Giles Holt" --width=800 --height=150 --center --size=fit --
center --button="Previous":3 --button="Input File Selection":0 --button="Additional Input File
Selection (1)":1 --button="Additional Input File Selection (2)":2 --buttons-layout=center --text="If
using an additional file list for running alongside the input file list, ensure that the files you want run
together carry the same starting name. As this

```

program will alphabetically order file lists separately and run them in order with each other. For example; Input file in list position 1 will be run with

the additional input file that's in position 1 in its own list"

```

mode="$?"
case $mode in

```

```

0)~/GGOSS/Scripts/FileSelection/PRICETI_FileSelection.sh ;;
1)~/GGOSS/Scripts/FileSelection/PRICETI_AdditionalInput1FileSelection.sh ;;
2)~/GGOSS/Scripts/FileSelection/PRICETI_AdditionalInput2FileSelection.sh ;;
3)~/GGOSS/Buttons/Button_PRICETI.sh ;;
esac

```

10.9.2.4.3 PRICE additional file selection option 1

```

#!/bin/sh

if [ -f ~/GGOSS/tmp/PRICETISelectedFiles.txt ]
then
rm ~/GGOSS/tmp/PRICETISelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/PRICETISelectedFilesP4.txt ]
then
rm ~/GGOSS/tmp/PRICETISelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="                GENOME SEQUENCING PROGRAM - PRICETI
file Input selection                Created by Giles Holt" --list --column="Select files you wish to run" --
multiple --width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --
separator=" > ~/GGOSS/tmp/PRICETISelectedFiles.txt

mode="$?"
case $mode in
1)~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh ;;
esac

if [ -s ~/GGOSS/tmp/PRICETISelectedFiles.txt ]
then

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETISelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="                GENOME SEQUENCING
PROGRAM - PRICETI file Input selection                Created by Giles Holt" --list --column="Select
files you wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/PRICETISelectedFilesP4.txt

mode="$?"
case $mode in
2)~/GGOSS/Scripts/FileSelection/PRICETI_FileSelection.sh ;;
esac

```

```

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/PRICETISelectedFilesP4.txt ]
then

~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh
fi

```

10.9.2.4.4 PRICE additional file selection option 2

```

#!/bin/sh

if [ -f ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFiles.txt ]
then
rm ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt ]
then
rm ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="
GENOME SEQUENCING PROGRAM - PRICETI
additional file Input (1) selection Created by Giles Holt" --list --column="Select files you wish
to run" --multiple --width 800 --height 600 --center --align=center --button="Open" --
button="Previous":1 --separator=" > ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFiles.txt

mode="$?"
case $mode in
1)~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh ;;
esac

if [ -s ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFiles.txt ]
then

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" >
~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

```

```

echo "$SelectFile2" | yad --title="
PROGRAM - PRICETI additional file Input (1) selection          GENOME SEQUENCING
                                                                    Created by Giles Holt" --list --
column="Select files you wish to run" --multiple --width 800 --height 600 --center --align=center --
button="OK" --button="Previous":2 --separator=">" >
~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt

```

```

mode="$?"
case $mode in
  2)~/GGOSS/Scripts/FileSelection/PRICETI_AdditionalInput1FileSelection.sh ;;
esac

```

```

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt

```

```
fi
```

```

if [ -s ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt ]
then

```

```

~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh
fi

```

10.9.2.4.5 PRICE additional file selection option 3

```
#!/bin/sh
```

```

if [ -f ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFiles.txt ]
then
rm ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFiles.txt
fi

```

```

if [ -f ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt ]
then
rm ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt
fi

```

```
SelectFile=$(ls ~/GGOSS_InputOutput/)
```

```

echo "$SelectFile" | yad --title="
additional file Input (2) selection          GENOME SEQUENCING PROGRAM - PRICETI
                                                                    Created by Giles Holt" --list --column="Select files you wish
to run" --multiple --width 800 --height 600 --center --align=center --button="Open" --
button="Previous":1 --separator=">" > ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFiles.txt

```

```

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh ;;
esac

```

```

    if [ -s ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFiles.txt ]
then
FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" >
~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="
                                GENOME SEQUENCING
PROGRAM - PRICETI additional file Input (2) selection      Created by Giles Holt" --list --
column="Select files you wish to run" --multiple --width 800 --height 600 --center --align=center --
button="OK" --button="Previous":2 --separator=" " >
~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt

mode="$?"
    case $mode in
        2)~/GGOSS/Scripts/FileSelection/PRICETI_AdditionalInput2FileSelection.sh ;;
    esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt ]
then

~/GGOSS/Buttons/PriceTiInputTypeFileSelect.sh
fi

```

10.9.2.4.6 PRICE additional settings main menu

```

#!/bin/bash

ICON=~/GGOSS/Pictures/PRICETI.jpg

yad --title="
PRICETI                                GENOME SEQUENCING PROGRAM -
                                created by Giles Holt" --width=800 --height=500 --center --image=$ICON
--size=fit --center --button="Previous":7 --button="PRICE Input/Output Settings":1 --button="PRICE
Parameter Settings":6 --button="PRICE Filter Read Settings":4 --button="PRICE Filter Contig
Settings":5 --buttons-layout=center

mode="$?"
    case $mode in
        7)~/GGOSS/Buttons/Button_PRICETI.sh ;;
        1)~/GGOSS/Buttons/PRICETI_Readsettings.sh ;;
        4)~/GGOSS/Buttons/PRICETI_FilterSettings.sh ;;
        5)~/GGOSS/Buttons/PRICETI_FilterOtherSettings.sh ;;
        6)~/GGOSS/Buttons/PRICETI_ParameterSettings.sh ;;
    esac

```

10.9.2.4.7 BWA main menu, settings and file selection

```
#!/bin/bash

#####-----#####
##### BWA MENU #####
#####-----#####
CustomSettingsMenu=2
FalseSelectionOfSaveAndFurtherSettings=3
DontUseAPresavedSetting=2
PreSavedSelectableChosen=2
CustomWasSaved=2
if [ -f ~/GGOSS/tmp/BWA_FileSelectionPrevious.txt ];then
BWA_FileSelectionPrevious=$(head -n1 ~/GGOSS/tmp/BWA_FileSelectionPrevious.txt)
fi

BWA_FileSelectionPrevious=$(( $BWA_FileSelectionPrevious + 1 ))

## the ol file selection method was actually a problem anyway and preventing closing with repeat re-
openings, fix file selection with method used in QUAST
BWAfileselect=2
BWA_SettingsSelected=2
BWA_PreSavedSettingsSelected=2
ICON=~/.GGOSS/Pictures/BWA.jpeg

if [ $BWA_FileSelectionPrevious != 2 ];then
BWA_FileSelectionPrevious=3
  if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/BWA_OpenMenu.txt
  fi
yad --title="GENOME SEQUENCING PROGRAM -- BWA" "GGOSS - Created
by Giles Holt" --center --image=$ICON --size=fit --center --fixed --button=" Previous ":4 --
button=" Run BWA ":6 --button=" Select samples ":5 --button=" Settings
":7 --button="Use Pre-saved Settings":8 --button=" BWA Manual " --buttons-layout=start --
text=" BWA

Maps DNA sequences against a large
reference genome, such as the human
genome. It consists of three algorithms:
BWA-backtrack, BWA-SW and BWA-MEM"

mode="$?"
case $mode in
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)BWAfileselect=1 ;;
  6)~/GGOSS/Scripts/BWA_script.sh ;;
  7)BWA_SettingsSelected=1 ;;
  8)BWA_PreSavedSettingsSelected=1 ;;
esac

#####---- BWA Pre-saved SETTINGS ----#####
if [ $BWA_PreSavedSettingsSelected = 1 ];then
  if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
```

```
rm ~/GGOSS/tmp/BWA_OpenMenu.txt
fi
```

```
ls ~/GGOSS/tmp/PreSavedSettings/BWA | yad --title="GENOME SEQUENCING PROGRAM --
BWA -- Pre-saved setting options          GGOSS - Created by Giles Holt" --list --
column="Select pre-saved setting to use" --width 800 --height 600 --center --align=center --button="
Previous      ":10 --button="      Select setting type      ":1 --buttons-layout=center --text="Select
" --separator=" > ~/GGOSS/tmp/BWApresavedSettingSelected.txt
```

```
mode="$?"
case $mode in
  10)echo "1" > ~/GGOSS/tmp/BWA_OpenMenu.txt && DontUseAPresavedSetting=1 ;;
  1)echo "1" > ~/GGOSS/tmp/BWA_OpenMenu.txt ;;
esac
```

```
if [ "$DontUseAPresavedSetting" != "1" ];then
#take the saved settings file name
PreSavedFileName=$(awk 'NR==1 {print}' ~/GGOSS/tmp/BWApresavedSettingSelected.txt)
```

```
#copy it and rename copy to the settings file used by the run script
cp ~/GGOSS/tmp/PreSavedSettings/BWA/${PreSavedFileName}
~/GGOSS/tmp/BWA_SettingsChange.txt
fi
fi
```

```
#####---- BWA SETTINGS ----#####
```

```
if [ $BWA_SettingsSelected = 1 ];then
  if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/BWA_OpenMenu.txt
  fi
fi
```

```
yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM -- BWA Settings          GGOSS -
Created by Giles Holt" --on-top --center --size=fit --center --form \
--field="
Preset default run types":CB \
--field="
Create SAM file output":CB \
--field="                                Directory path to, and name
of, reference genome (example given):" \
--field="                                Primary BWA
tool to use (Custom run type only)":CB \
--field="                                Secondary BWA tool to use
(Custom run type only, optional for custom run)":CB \
--field="                                Read type: Single or
paired end (Custom run type only)":CB \
--field="Tool: samse; (-n) Max no. of alignments to output in the XA tag for reads paired properly. If
a read has > INT hits, the XA tag won't be written (Custom run type only):" \
--field="                                Tool: samse; (-r) Specify the read group in a format
like '@RG\tID:foo\tSM:bar' (Custom run type only):" \
```



```

chosen to be saved as a selectable pre-saved option in the future please provide a name for the setting
choices and select the 'Make a pre-saved selectable' button. Otherwise just select ignore" --form \
--field="Save setting selection as:" \
'\ \
--text-info --show-uri --center --wrap \
--button="Make a pre-saved selectable":0 --button="Ignore":2 --editable --
filename=~/.GGOSS/tmp/BWA_SettingsPreSavetmp.txt >
~/GGOSS/tmp/BWA_SettingsPreSavetmp.txt

```

```

mode="$?"
case $mode in
  0)PreSavedSelectableChosen=1 ;;
  2)PreSavedSelectableChosen=2 ;;
esac

```

```

PreSavedSettingName=$(awk -F '|' 'NR==1 {print $1}' ~/GGOSS/tmp/BWA_SettingsPreSavetmp.txt)
echo "PreSavedSettingName: $PreSavedSettingName"
if [ "$PreSavedSelectableChosen" = "1" ];then
  cp ~/GGOSS/tmp/BWA_SettingsChange.txt
~/GGOSS/tmp/PreSavedSettings/BWA/${PreSavedSettingName}
fi
fi

```

```

#Was mem selected
type1=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
type2=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/BWA_SettingsChange.txt)

```

```

if [ "$type1" = "mem" ] || [ "$type2" = "mem" ];then
#add 'mem' to the start of the last line in the settings file
echo -n "mem|" >> ~/GGOSS/tmp/BWA_SettingsChange.txt

```

```

yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM -- BWA Settings          GGOSS -
Created by Giles Holt" --on-top --size=fit --center --form \
--field="                                          Tool: mem; (-t) Number of
threads (Custom run type only):" \
--field="                                          Tool: mem; (-k) Min seed
length (Custom run type only):" \
--field="                                          Tool: mem; (-w) Band
width (Custom run type only):" \
--field="                                          Tool: mem; (-d) Off-diagonal
X-dropoff (Custom run type only):" \
--field="                                          Tool: mem; (-r) Trigger re-seeding for a MEM longer
than minSeedLen times? (Custom run type only):" \
--field="                                          Tool: mem; (-c) Discard a MEM if it has more than
INT occurrence in the genome (Custom run type only):" \
--field="          Tool: mem; (-P) paired-end mode, using SW: rescue missing hits only but don't
try to find hits fitting a proper pair (Custom run type only):":CB \
--field="                                          Tool: mem; (-A)
Matching score (Custom run type only):" \
--field="          Tool: mem; (-B) Mismatch penalty. The sequence error rate is
approximately: {.75 * exp[-log(4) * B/A]} (Custom run type only):" \
--field="          Tool: mem; (-O) Gap open penalty (for
opening a zero-length gap) (Custom run type only):" \

```



```

else
FalseSelectionOfSaveAndFurtherSettings=1
fi

#Was aln selected
if [ "$type1" = "aln" ] || [ "$type2" = "aln" ];then
#add 'mem|' to the start of the last line in the settings file
echo -n "aln|" >> ~/GGOSS/tmp/BWA_SettingsChange.txt

yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM -- BWA Settings          GGOSS -
Created by Giles Holt" --on-top --size=fit --center --form \
--field="Tool: aln; (-n) Maximum edit distance if the value is INT, or the fraction of missing
alignments given 2% uniform base error rate if FLOAT (Custom run type only):" \
--field="                                          Tool: aln; (-o) Maximum number
of gap opens (Custom run type only):" \
--field="                                          Tool: aln; (-e) Maximum number of gap extensions, -
1 for k-difference mode (Custom run type only):" \
--field="                                          Tool: aln; (-d) Disallow a long deletion within INT
bp towards the 3'-end (Custom run type only):" \
--field="                                          Tool: aln; (-i) Disallow an indel within INT bp
towards the ends (Custom run type only):" \
--field="                                          Tool: aln; (-l) Take the first INT
subsequence as seed (Custom run type only):" \
--field="                                          Tool: aln; (-k) Maximum edit
distance in the seed (Custom run type only):" \
--field="                                          Tool: aln; (-t) Number of threads (multi-
threading mode) (Custom run type only):" \
--field="                                          Tool: aln; (-M) Mismatch
penalty (Custom run type only):" \
--field="                                          Tool: aln; (-O) Gap open
penalty (Custom run type only):" \
--field="                                          Tool: aln; (-E) Gap extension
penalty (Custom run type only):" \
--field="                                          Tool: aln; (-R) Proceed with suboptimal alignments if there are no
more than INT equally best hits (Custom run type only):" \
--field="                                          Tool: aln; (-c) Reverse query but not complement it, which is required
for alignment in the color space (Custom run type only):":CB \
--field="                                          Tool: aln; (-N) Disable iterative search. All hits with no more than
maxDiff differences will be found (Custom run type only):":CB \
--field="                                          Tool: aln; (-q) Parameter for read
trimming (Custom run type only):" \
--field="                                          Tool: aln; (-I) The input is in the Illumina
1.3+ read format (Custom run type only):":CB \
--field="                                          Tool: aln; (-B) Length of barcode starting
from the 5'-end (Custom run type only):" \
--field="                                          Tool: aln; (-b) Specify the input read sequence file
is the BAM format (Custom run type only):":CB \
--field="                                          Tool: aln; (-0) When -b is specified, only use single-
end reads in mapping (Custom run type only):":CB \
--field="                                          Tool: aln; (-1) When -b is specified, only use the first read in
a read pair in mapping (Custom run type only):":CB \
--field="                                          Tool: aln; (-2) When -b is specified, only use the second read
in a read pair in mapping (Custom run type only):":CB \
--text-info --show-uri --center --wrap \

```



```

if [ $BWAfileselect = 1 ];then
  if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/BWA_OpenMenu.txt
  fi

  if [ -f ~/GGOSS/tmp/BWASelectedFiles.txt ]
  then
    rm ~/GGOSS/tmp/BWASelectedFiles.txt
  fi

  if [ -f ~/GGOSS/tmp/BWASelectedFilesP4.txt ]
  then
    rm ~/GGOSS/tmp/BWASelectedFilesP4.txt
  fi

  SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="          GENOME SEQUENCING PROGRAM - BWA file
selection          Created by Giles Holt" --list --column="Select files you wish to run" --multiple --
width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --separator=" >
~/GGOSS/tmp/BWASelectedFiles.txt

mode="$?"
  case $mode in
    1)echo "3" > ~/GGOSS/tmp/BWA_FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/BWA_OpenMenu.txt ;;
    esac

  if [ -s ~/GGOSS/tmp/BWASelectedFiles.txt ];then
    if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
      rm ~/GGOSS/tmp/BWA_OpenMenu.txt
    fi
    FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/BWASelectedFiles.txt)
    echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

    SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="          GENOME SEQUENCING
PROGRAM - BWA file selection          Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/BWASelectedFilesP4.txt

mode="$?"
  case $mode in
    2)echo "1" > ~/GGOSS/tmp/BWA_FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/BWA_OpenMenu.txt ;;
    esac

  FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/BWASelectedFilesP4.txt)
  echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt
fi

```

```

if [ -s ~/GGOSS/tmp/BWASelectedFilesP4.txt ]
then
  if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/BWA_OpenMenu.txt
  fi
echo "1" > ~/GGOSS/tmp/BWA_OpenMenu.txt
fi

fi

```

#####---- BWA return to/Open menu after selections ----#####

```

if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
BWA_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/BWA_OpenMenu.txt)
fi
BWA_OpenMenu=$(( $BWA_OpenMenuPrep + 1 ))

if [ $BWA_OpenMenu = 2 ];then
~/GGOSS/Buttons/Button_BWA.sh
fi

if [ -f ~/GGOSS/tmp/BWA_OpenMenu.txt ];then
rm ~/GGOSS/tmp/BWA_OpenMenu.txt
fi

```

10.9.2.4.8 BLAST Main menu, settings, file selection

```

#!/bin/bash

if [ -f ~/GGOSS/tmp/BlastFileSelection.txt ] || [ -f ~/GGOSS/tmp/CheckpointBlast_SelectedFiles.txt
];then
  if [ -f ~/GGOSS/tmp/BlastFileSelection.txt ];then
    BlastFileSelection=$(head -n1 ~/GGOSS/tmp/BlastFileSelection.txt)
    rm ~/GGOSS/tmp/BlastFileSelection.txt
  else
    BlastFileSelection=2
  fi

  if [ "$BlastFileSelection" = 1 ];then

    if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then
      rm ~/GGOSS/tmp/Blast_OpenMenu.txt
    fi

    if [ -f ~/GGOSS/tmp/Blast_SelectedFiles.txt ];then
      rm ~/GGOSS/tmp/Blast_SelectedFiles.txt
    fi

    if [ -f ~/GGOSS/tmp/BlastSelectedFilesP4.txt ];then

```

```

rm ~/GGOSS/tmp/BlastSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="                GENOME SEQUENCING PROGRAM - Blast -
-file selection          Created by Giles Holt" --list --column="Select files to re-assemble against ref
genome" --multiple --width 800 --height 600 --center --align=center --on-top --button="OK" --
button="Previous":1 --separator=" --filename=~/GGOSS/tmp/Blast_SelectedFiles.txt >
~/GGOSS/tmp/Blast_SelectedFiles.txt

#this means in the case of script exit do 1 of the following
mode="$?"
case $mode in
    1)echo "1" > ~/GGOSS/tmp/Blast_OpenMenu.txt ;;
esac

if [ -s ~/GGOSS/tmp/Blast_SelectedFiles.txt ];then
echo "1" > ~/GGOSS/tmp/CheckpointBlast_SelectedFiles.txt
fi

if [ -s ~/GGOSS/tmp/CheckpointBlast_SelectedFiles.txt ];then

if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Blast_OpenMenu.txt
fi
if [ -f ~/GGOSS/tmp/BlastFileSelection.txt ];then
rm ~/GGOSS/tmp/BlastFileSelection.txt
fi

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/Blast_SelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="                GENOME
SEQUENCING PROGRAM - Blast file selection          Created by Giles Holt" --list --
column="Select files to re-assemble against ref genome" --multiple --width 800 --height 600 --center -
-align=center --button="OK" --button="Previous":2 --separator=" >
~/GGOSS/tmp/BlastSelectedFilesP4.txt

mode="$?"
case $mode in
    2)rm ~/GGOSS/tmp/CheckpointBlast_SelectedFiles.txt && echo "1" >
~/GGOSS/tmp/BlastFileSelection.txt ;;
esac

if [ -s ~/GGOSS/tmp/BlastSelectedFilesP4.txt ];then
rm ~/GGOSS/tmp/BlastFileSelection.txt
rm ~/GGOSS/tmp/CheckpointBlast_SelectedFiles.txt
fi

```

```

echo "1" > ~/GGOSS/tmp/Blast_OpenMenu.txt

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/BlastSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

fi

fi

#####----- File Selection End -----#####

#####----- Blast menu -----#####

if [ -f ~/GGOSS/tmp/BlastFileSelection.txt ];then
echo ""
else

if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Blast_OpenMenu.txt
fi

ICON=~/GGOSS/Pictures/DNA2_1000.jpg

yad --title="
BLAST                               GENOME SEQUENCING PROGRAM --
Created by Giles Holt" --text="    BLAST

Can be used to align
Genomic data to reference
databases, identifying
genes, proteins,
organisms, etc" / --center --image=$ICON image-on-top --size=fit --center --button="Previous":4 --
button="Run":0 --button="Select samples":1 --button="BLAST settings
":2 --button="Make Blast database":5 --button="Make RPS-Blast database":6 --button="Pre-made
Settings":7 --button="BLAST Manual":3 --buttons-layout=center

mode="$?"
case $mode in
0)Choice=1 ;;
1)echo "1" > ~/GGOSS/tmp/BlastFileSelection.txt && echo "1" >
~/GGOSS/tmp/Blast_OpenMenu.txt ;;
2)Choice=3 ;;
3)Choice=4 ;;
4)~/GGOSS/GenomicsProgram.sh ;;
5)Choice=5 ;;
6)Choice=6 ;;
6)Choice=7 ;;
esac

#####----- Settings Start-----#####

```

```
##### SORT this still
if [ "$Choice" = 3 ];then
```

```
yad --title="
BLAST                               GENOME SEQUENCING PROGRAM --
Created by Giles Holt" --text="      BLAST
```

```
Can be used to align
Genomic data to reference
databases, identifying
genes, proteins,
organisms, etc" / --center --image=$ICON image-on-top --size=fit --center --button="Previous":8 --
button="      Base Blast settings      ":1 --button="      Blastn      ":2 --button="
Blastp      ":3 --button="      Blastx      ":4 --button="      tBlastn      ":5 --
button="      tBlastx      ":6 --button="      rpsBlast      ":7 --buttons-layout=center
```

```
mode="$?"
case $mode in
  1)BLAST="blast" ;;
  2)BLAST="blastn" ;;
  3)BLAST="blastp" ;;
  4)BLAST="blastx" ;;
  5)BLAST="tblastn" ;;
  6)BLAST="tblastx" ;;
  7)BLAST="rpsblast" ;;
  8)~/GGOSS/GenomicsProgram.sh
esac
```

```
fi
```

```
if [[ $BLAST = "blast" ]];then
yad --title="
BLAST Settings          Created by Giles Holt" --center --size=fit --center --form \
--field="
                                BLAST database to use:" :CB \
--field="
                                Location on the query sequence (query-loc,
optional):" :CB \
--field="
                                Expect value (E) for saving hits (evaluate, optional):" \
--field="
                                File with subject sequence(s) to search (subject, optional):"
\
--field="
                                Location on the subject sequence (Format: start-stop) (subject-loc,
optional):" \
--field="
                                Show NCBI GIs in report (show-gis,
optional):" :CB \
--field="
                                Number of db sequences to show one-line descriptions for (num-
descriptions, optional):" \
--field="
                                Number of database sequences to show alignments for (num-
alignments, optional):" \
--field="
                                Number of aligned sequences to keep. Not compatible with num-des or num-align (max-
target-seqs, optional):" \
--field="
                                Max No. of HSPs (alignments) to keep for any single query-subject pair
(max_hsp, optional):" \
--field="
                                Produce HTML output (html, optional):" :CB
\
--field="
                                Restrict search of database to GI's listed in this file. Local only (gilist,
optional):" :CB \
```

```

--field="Restrict search of database to everything except GI's listed in this file. Local only. negative-
gelist, optional):":CB \
--field="          Restrict search with the given Entrez query. Remote searches only (entrez-
query, optional):":CB \
--field="          Delete a hit that is enveloped by at least this many higher-scoring hits (culling-limit,
optional):" \
--field="          Best Hit algorithm overhang value (recommended value: 0.1) (best-hit-
overhang, optional):" \
--field="          Best Hit algorithm score edge value (recommended value: 0.1) (best-hit-score-
edge, optional):" \
--field="          Effective size of the database (dbsize, optional):" \
--field="          Effective length of the search space (searchsp, optional):"
\
--field="          Search strategy file to read (import-search-strategy,
optional):" \
--field="          Record search strategy to this file (export-search-strategy,
optional):" \
--field="          Parse query and subject bar delimited sequence identifiers (parse-deflines,
optional):":CB \
--field="          Execute search on NCBI servers? (remote,
optional):":CB \
--field="          Alignment view options (outfmt, optional):" \
"ViralBlast!BacteriaBlast!FungalBlast!Custom Database" "No!Yes!" '-''-'' "No!Yes!" '-''-''-''
"No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" '-''-''-''-''-''-'' "No!Yes!" "No!Yes!" '-'\
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/BLASTSettingsChange.txt > ~/.GGOSS/tmp/BLASTSettingsChange.txt

mode="$?"
case $mode in
  0)echo "1" > ~/.GGOSS/tmp/Blast_OpenMenu.txt ;;
  1)echo "1" > ~/.GGOSS/tmp/Blast_OpenMenu.txt ;;
esac

fi

if [[ $BLAST = "blastn" ]];then
yad --title="          GENOME SEQUENCING PROGRAM --
BLASTn Settings      Created by Giles Holt" --center --size=fit --center --form \
--field="          word_size:Length of initial exact match:" \
--field="          gapopen: Cost to open a gap:" \
--field="          gapextend: Cost to extend a gap:" \
--field="          reward: Reward for a nucleotide match:" \
--field="          penalty: Penalty for a nucleotide mismatch:" \
--field="          strand: Query strand(s) to search against database/subject":CB
\
--field="          dust: Filter query sequence with dust:" \
--field="          filtering_db: Mask query using the sequences in this database
(optional):" \
--field="          window_masker_taxid: Enable WindowMasker filtering using a Taxonomic
ID (optional):" \
--field="          window_masker_db: Enable WindowMasker filtering using this file
(optional):" \

```



```

--field="                                db_gen_code: Genetic code to translate subject
sequences:" \
--field="max_intron_length: Length of largest intron allowed in translated nucleotide sequence when
linking multiple distinct alignments:" \
'-''-' "No!Yes!" "false!true!" "No!Yes!" '-''-' "both!minus!plus!" '-''-' \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/BLASTSettingsChange.txt >
~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt

```

```

mode="$?"
case $mode in
0)echo "1" > ~/.GGOSS/tmp/Blast_OpenMenu.txt ;;
1)echo "1" > ~/.GGOSS/tmp/Blast_OpenMenu.txt ;;
esac

```

```
fi
```

```
fi
```

```
#####----- Settings End -----#####
```

```

# to be put into each setting menu to allow the save
2)echo "1" > ~/.GGOSS/tmp/Blast_OpenMenu.txt & PreSavedSettingMenu=1 ;;
esac
if [ $PreSavedSettingMenu = 1 ];then
cp ~/.GGOSS/tmp/BLASTSettingsChange.txt
~/GGOSS/tmp/PreSavedSettings/$NameToSaveFile.txt
fi

```

```
#####----- Saved settings Start -----#####
```

```
if [ "$Choice" = 7 ];then
```

```

yad --title="                                GENOME SEQUENCING PROGRAM --
BLAST Pre-Saved Settings      Created by Giles Holt" --text "Select pre-saved settings to use:" --
center --size=fit --center --form \
--field="BLAST base setting:":CB \
--field="BLASTn setting:":CB \
--field="BLASTp setting:":CB \
--field="tBLASTn setting:":CB \
--field="tBLASTx setting:":CB \
--field="rpsBLAST setting:":CB \

```

```

"N/A!$BaseSetting1!$BaseSetting2!$BaseSetting3!$BaseSetting4!$BaseSetting5!$BaseSetting6!$Base
Setting7!$BaseSetting8!$BaseSetting9!$BaseSetting10!$BaseSetting11!$BaseSetting12!$BaseSetti
ng13!$BaseSetting14!$BaseSetting15!$BaseSetting16!$BaseSetting17!$BaseSetting18!$BaseSetting
19!$BaseSetting20!"

```

```

"N/A!$BLASTnSetting1!$BLASTnSetting2!$BLASTnSetting3!$BLASTnSetting4!$BLASTnSetting
5!$BLASTnSetting6!$BLASTnSetting7!$BLASTnSetting8!$BLASTnSetting9!$BLASTnSetting10!$
BLASTnSetting11!$BLASTnSetting12!$BLASTnSetting13!$BLASTnSetting14!$BLASTnSetting15
!$BLASTnSetting16!$BLASTnSetting17!$BLASTnSetting18!$BLASTnSetting19!$BLASTnSetting
20!"

```

```

"N/A!$BLASTpSetting1!$BLASTpSetting2!$BLASTpSetting3!$BLASTpSetting4!$BLASTpSetting

```

```
5!$BLASTpSetting6!$BLASTpSetting7!$BLASTpSetting8!$BLASTpSetting9!$BLASTpSetting10!$
BLASTpSetting11!$BLASTpSetting12!$BLASTpSetting13!$BLASTpSetting14!$BLASTpSetting15
!$BLASTpSetting16!$BLASTpSetting17!$BLASTpSetting18!$BLASTpSetting19!$BLASTpSetting
20!"
```

```
"N/A!$tBLASTnSetting1!$tBLASTnSetting2!$tBLASTnSetting3!$tBLASTnSetting4!$tBLASTnSetti
ng5!$tBLASTnSetting6!$tBLASTnSetting7!$tBLASTnSetting8!$tBLASTnSetting9!$tBLASTnSetti
ng10!$tBLASTnSetting11!$tBLASTnSetting12!$tBLASTnSetting13!$tBLASTnSetting14!$tBLAST
nSetting15!$tBLASTnSetting16!$tBLASTnSetting17!$tBLASTnSetting18!$tBLASTnSetting19!$tB
LASTnSetting20!"
```

```
"N/A!$tBLASTxSetting1!$tBLASTxSetting2!$tBLASTxSetting3!$tBLASTxSetting4!$tBLASTxSett
ing5!$tBLASTxSetting6!$tBLASTxSetting7!$tBLASTxSetting8!$tBLASTxSetting9!$tBLASTxSetti
ng10!$tBLASTxSetting11!$tBLASTxSetting12!$tBLASTxSetting13!$tBLASTxSetting14!$tBLAST
xSetting15!$tBLASTxSetting16!$tBLASTxSetting17!$tBLASTxSetting18!$tBLASTxSetting19!$tB
LASTxSetting20!"
```

```
"N/A!$rpsBLASTSetting1!$rpsBLASTSetting2!$rpsBLASTSetting3!$rpsBLASTSetting4!$rpsBLAS
TSetting5!$rpsBLASTSetting6!$rpsBLASTSetting7!$rpsBLASTSetting8!$rpsBLASTSetting9!$rpsB
LASTSetting10!$rpsBLASTSetting11!$rpsBLASTSetting12!$rpsBLASTSetting13!$rpsBLASTSetti
ng14!$rpsBLASTSetting15!$rpsBLASTSetting16!$rpsBLASTSetting17!$rpsBLASTSetting18!$rpsB
LASTSetting19!$rpsBLASTSetting20!" \
```

```
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="OK":0 --button="":2 --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/BLASTPreSavedSettingsSelect.txt >
~/GGOSS/tmp/BLASTPreSavedSettingsSelect.txt
```

```
#The below needs to be specific to each blast type
SelectedBaseSetting=$(~/GGOSS/tmp/BLASTPreSavedSettingsSelect.txt)
cp ~/GGOSS/tmp/PreSavedSettings/$SelectedBaseSetting
~/GGOSS/tmp/PreSavedSettings/BLASTSettingsChange.txt
fi
```

```
#####----- Saved settings end -----#####
```

```
#####----- Run START -----#####
```

```
if [ "$Choice" = 1 ];then
```

```
yad --title="                                GENOME SEQUENCING PROGRAM -- BLAST
Run type      Created by Giles Holt" --center --size=fit --center --form \
--field="      BLAST run type:" :CB \
--field="Single or paired end:" :CB \
--field="      File format:" :CB \
"blastn!blastp!blastx!tblastn!tblastx!rpsblast!" "Single!Paired!" "FASTA!FASTQ!" \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/BLASTSettingsRunType.txt > ~/GGOSS/tmp/BLASTSettingsRunType.txt
```

```
mode="$?"
```

```
case $mode in
  0)AdaptRunScript=1 && echo "1" > ~/GGOSS/tmp/Blast_OpenMenu.txt ;;
  1)echo "1" > ~/GGOSS/tmp/Blast_OpenMenu.txt ;;
esac
```

```
if [ "$AdaptRunScript" = 1 ];then
```

```

echo "Applying base BLAST settings..."

cp ~/GGOSS/Scripts/BLAST.sh ~/GGOSS/Scripts/RunBLAST.sh

#edit base BLAST settings, these need to be edited regardless of blast type
#ref database
RefDataBase=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
sed -i -e "s/DATABASE\+/$RefDataBase/g" ~/GGOSS/Scripts/RunBLAST.sh

PathToDataBase=$(awk -F '|' '{print $12}' ~/GGOSS/tmp/PathToTools_SettingsChange.txt)
if [ "$PathToDataBase" = "N/A" ];then
PathToDataBase="$HOME/"
fi
sed -i -e "s|PathToDataBase|${PathToDataBase}|g" ~/GGOSS/Scripts/RunBLAST.sh

#query location
QueryLoc=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$QueryLoc" = "No" ]];then
sed -i -e "s/queryloc\+/-query_loc $QueryLoc/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/queryloc \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#evaluate
evaluate=$(awk -F '|' '{print $3}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$evaluate" = "-" ]];then
sed -i -e "s/evaluate\+/-evaluate $evaluate/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/evaluate \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#subject
subject=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$subject" = "-" ]];then
sed -i -e "s/subject\+/-subject $subject/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/subject \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#subjectloc
subjectloc=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$subjectloc" = "-" ]];then
sed -i -e "s/subjectloc\+/-subject_loc $subjectloc/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/subjectloc \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#showgis
showgis=$(awk -F '|' '{print $6}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$showgis" = "No" ]];then
sed -i -e "s/showgis\+/-show_gis/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/showgis \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

```

```

#numdescriptions
numdescriptions=$(awk -F '|' '{print $7}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$numdescriptions" = "-" ]];then
    sed -i -e "s/numdescriptions\+/-num_descriptions $numdescriptions/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/numdescriptions \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#numalignments
numalignments=$(awk -F '|' '{print $8}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$numdescriptions" = "-" ]];then
    sed -i -e "s/numalignments\+/-num_alignments $numalignments/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/numalignments \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#maxtargetseqs
maxtargetseqs=$(awk -F '|' '{print $9}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$maxtargetseqs" = "-" ]];then
    sed -i -e "s/maxtargetseqs\+/-max_target_seqs $maxtargetseqs/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/maxtargetseqs \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#maxhsp
maxhsp=$(awk -F '|' '{print $10}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$maxhsp" = "-" ]];then
    sed -i -e "s/maxhsp\+/-max_hsp $maxhsp/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/maxhsp \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#html
html=$(awk -F '|' '{print $11}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$html" = "No" ]];then
    sed -i -e "s/html\+/-html/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/html \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#gilist
gilist=$(awk -F '|' '{print $12}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$gilist" = "No" ]];then
    sed -i -e "s/gi_list\+/-gilist/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/gi_list \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#negativegilist
negativegilist=$(awk -F '|' '{print $13}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$negativegilist" = "No" ]];then

```

```

    sed -i -e "s/negativegelist\+/-negative_gelist/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/negativegelist \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#entrezquery
entrezquery=$(awk -F '|' '{print $14}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$entrezquery" = "No" ]];then
    sed -i -e "s/entrezquery\+/-entrez_query/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/entrezquery \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#cullinglimit
cullinglimit=$(awk -F '|' '{print $15}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$cullinglimit" = "-" ]];then
    sed -i -e "s/cullinglimit\+/-culling_limit $cullinglimit/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/cullinglimit \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#besthitoverhang
besthitoverhang=$(awk -F '|' '{print $16}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$besthitoverhang" = "-" ]];then
    sed -i -e "s/besthitoverhang\+/-best_hit_overhang $besthitoverhang/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/besthitoverhang \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#besthitscoreedge
besthitscoreedge=$(awk -F '|' '{print $17}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$besthitscoreedge" = "-" ]];then
    sed -i -e "s/besthitscoreedge\+/-best_hit_score_edge $besthitscoreedge/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/besthitscoreedge \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#dbsize
dbsize=$(awk -F '|' '{print $18}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$dbsize" = "-" ]];then
    sed -i -e "s/dbsize\+/-dbsize $dbsize/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/dbsize \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#searchsp
searchsp=$(awk -F '|' '{print $19}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$searchsp" = "-" ]];then
    sed -i -e "s/searchsp\+/-searchsp $searchsp/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/searchsp \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

```

```

#importsearchstrategy
importsearchstrategy=$(awk -F '|' '{print $20}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$importsearchstrategy" = "-" ]];then
    sed -i -e "s/importsearchstrategy\+/-import_search_strategy $importsearchstrategy/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/importsearchstrategy \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#exportsearchstrategy
exportsearchstrategy=$(awk -F '|' '{print $21}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$exportsearchstrategy" = "-" ]];then
    sed -i -e "s/exportsearchstrategy\+/-export_search_strategy $exportsearchstrategy/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/exportsearchstrategy \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#parsedeflines
parsedeflines=$(awk -F '|' '{print $22}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$parsedeflines" = "No" ]];then
    sed -i -e "s/parsedeflines\+/-parse_deflines/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/parsedeflines \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#remote
remote=$(awk -F '|' '{print $23}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$remote" = "No" ]];then
    sed -i -e "s/remote\+/-remote/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/remote \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

#outfmt
outfmt=$(awk -F '|' '{print $24}' ~/GGOSS/tmp/BLASTSettingsChange.txt)
if [[ ! "$outfmt" = "-" ]];then
    # sed -i -e "s/outfmt\+/-outfmt/g" ~/GGOSS/Scripts/RunBLAST.sh
    echo "unsure how to implement outfmt"
else
    sed -i -e "s/outfmt \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    echo "unsure how to implement outfmt"
fi

echo "Completed applying of base BLAST settings..."

# Change settings specific to BLAST type
BLASTtype=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/BLASTSettingsRunType.txt)

if [[ "$BLASTtype" = "blastn" ]];then
    echo "Applying base BLASTn settings..."
    #sed -i -e "s/FileInput1FileName\+/$filenameR1/g" ~/GGOSS/Scripts/RunBLAST.sh

```

```

echo "Completed applying of base BLASTn settings..."
fi

if [[ "$BLASTtype" = "blastn" ]];then
echo "Applying base BLASTn settings..."

word_size=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$word_size" = "-" ]];then
sed -i -e "s/word_size\+/-word_size $word_size/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/word_size \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

gapopen=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$gapopen" = "-" ]];then
sed -i -e "s/gapopen\+/-gapopen $gapopen/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/gapopen \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

gapextend=$(awk -F '|' '{print $3}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$gapextend" = "-" ]];then
sed -i -e "s/gapextend\+/-gapextend $gapextend/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/gapextend \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

reward=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$reward" = "-" ]];then
sed -i -e "s/reward\+/-reward $reward/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/reward \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

penalty=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$penalty" = "-" ]];then
sed -i -e "s/penalty\+/-penalty $penalty/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/penalty \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

strand=$(awk -F '|' '{print $6}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$strand" = "both" ]] && [[ ! "$strand" = "minus" ]];then
sed -i -e "s/strand\+/-strand plus/g" ~/GGOSS/Scripts/RunBLAST.sh
else
if [[ ! "$strand" = "both" ]];then
sed -i -e "s/strand\+/-strand minus/g" ~/GGOSS/Scripts/RunBLAST.sh
else
sed -i -e "s/strand \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi
fi

```

```

dust=$(awk -F '|' '{print $7}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$dust" = "-" ]];then
    sed -i -e "s/dust\+/-dust $dust/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/dust \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

filtering_db=$(awk -F '|' '{print $8}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$filtering_db" = "-" ]];then
    sed -i -e "s/filtering_db\+/-filtering_db $filtering_db/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/filtering_db \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

window_masker_taxid=$(awk -F '|' '{print $9}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$window_masker_taxid" = "-" ]];then
    sed -i -e "s/window_masker_taxid\+/-window_masker_taxid $window_masker_taxid/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/window_masker_taxid \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

window_masker_db=$(awk -F '|' '{print $10}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$window_masker_db" = "-" ]];then
    sed -i -e "s/window_masker_db\+/-window_masker_db $window_masker_db/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/window_masker_db \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

soft_masking=$(awk -F '|' '{print $11}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$soft_masking" = "false" ]];then
    sed -i -e "s/soft_masking\+/-soft_masking true/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/soft_masking \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

lcase_masking=$(awk -F '|' '{print $12}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$lcase_masking" = "No" ]];then
    sed -i -e "s/lcase_masking\+/-lcase_masking/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/lcase_masking \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

db_soft_mask=$(awk -F '|' '{print $13}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$db_soft_mask" = "-" ]];then
    sed -i -e "s/db_soft_mask\+/-db_soft_mask $db_soft_mask/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/db_soft_mask \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

db_hard_mask=$(awk -F '|' '{print $14}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
if [[ ! "$db_soft_mask" = "-" ]];then

```

```

        sed -i -e "s/db_hard_mask\+/-db_hard_mask $db_hard_mask/g"
~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/db_hard_mask \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    perc_identity=$(awk -F '|' '{print $15}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$perc_identity" = "-" ]];then
        sed -i -e "s/perc_identity\+/-perc_identity $perc_identity/g" ~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/perc_identity \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    xdrop_ungap=$(awk -F '|' '{print $16}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$xdrop_ungap" = "-" ]];then
        sed -i -e "s/xdrop_ungap\+/-xdrop_ungap $xdrop_ungap/g" ~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/xdrop_ungap \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    xdrop_gap=$(awk -F '|' '{print $17}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$xdrop_gap" = "-" ]];then
        sed -i -e "s/xdrop_gap\+/-xdrop_gap $xdrop_gap/g" ~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/xdrop_gap \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    xdrop_gap_final=$(awk -F '|' '{print $18}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$xdrop_gap_final" = "-" ]];then
        sed -i -e "s/xdrop_gap_final\+/-xdrop_gap_final $xdrop_gap_final/g"
~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/xdrop_gap_final \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    min_raw_gapped_score=$(awk -F '|' '{print $19}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$min_raw_gapped_score" = "-" ]];then
        sed -i -e "s/min_raw_gapped_score\+/-min_raw_gapped_score $min_raw_gapped_score/g"
~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/min_raw_gapped_score \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    ungapped=$(awk -F '|' '{print $20}' ~/GGOSS/tmp/BLAST_n_SettingsChange.txt)
    if [[ ! "$ungapped" = "No" ]];then
        sed -i -e "s/ungapped\+/-ungapped/g" ~/GGOSS/Scripts/RunBLAST.sh
    else
        sed -i -e "s/ungapped \+//g" ~/GGOSS/Scripts/RunBLAST.sh
    fi

    echo "Completed applying of base BLASTn settings..."
    fi

    if [[ "$BLASTtype" = "tblastx" ]];then
        echo "Applying base tBLASTx settings..."

```

```

word_size=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$word_size" = "-" ]];then
    sed -i -e "s/word_size\+/-word_size $word_size/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/word_size \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

matrix=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$matrix" = "-" ]];then
    sed -i -e "s/matrix\+/-matrix $matrix/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/matrix \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

threshold=$(awk -F '|' '{print $3}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$threshold" = "-" ]];then
    sed -i -e "s/threshold\+/-threshold $threshold/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/threshold \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

seg=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$seg" = "No" ]];then
    sed -i -e "s/seg\+/-seg/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/seg \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

soft_masking=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$soft_masking" = "false" ]];then
    sed -i -e "s/soft_masking\+/-soft_masking true/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/soft_masking \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

lcase_masking=$(awk -F '|' '{print $6}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$lcase_masking" = "No" ]];then
    sed -i -e "s/lcase_masking\+/-lcase_masking/g" ~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/lcase_masking \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

db_soft_mask=$(awk -F '|' '{print $7}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$db_soft_mask" = "-" ]];then
    sed -i -e "s/db_soft_mask\+/-db_soft_mask $db_soft_mask/g"
~/GGOSS/Scripts/RunBLAST.sh
else
    sed -i -e "s/db_soft_mask \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

db_hard_mask=$(awk -F '|' '{print $8}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$db_hard_mask" = "-" ]];then
    sed -i -e "s/db_hard_mask\+/-db_hard_mask $db_hard_mask/g"
~/GGOSS/Scripts/RunBLAST.sh

```

```

else
  sed -i -e "s/db_hard_mask \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

strand=$(awk -F '|' '{print $9}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$strand" = "both" ]] && [[ ! "$strand" = "minus" ]];then
  sed -i -e "s/strand\+/-strand plus/g" ~/GGOSS/Scripts/RunBLAST.sh
else
  if [[ ! "$strand" = "both" ]];then
    sed -i -e "s/strand\+/-strand minus/g" ~/GGOSS/Scripts/RunBLAST.sh
  else
    sed -i -e "s/strand \+//g" ~/GGOSS/Scripts/RunBLAST.sh
  fi
fi

query_genetic_code=$(awk -F '|' '{print $10}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$query_genetic_code" = "-" ]];then
  sed -i -e "s/query_genetic_code\+/-query_genetic_code $query_genetic_code/g"
~/GGOSS/Scripts/RunBLAST.sh
else
  sed -i -e "s/query_genetic_code \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

db_gen_code=$(awk -F '|' '{print $11}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$db_gen_code" = "-" ]];then
  sed -i -e "s/db_gen_code\+/-db_gen_code $db_gen_code/g" ~/GGOSS/Scripts/RunBLAST.sh
else
  sed -i -e "s/db_gen_code \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

max_intron_length=$(awk -F '|' '{print $12}' ~/GGOSS/tmp/t_BLAST_x_SettingsChange.txt)
if [[ ! "$max_intron_length" = "-" ]];then
  sed -i -e "s/max_intron_length\+/-max_intron_length $max_intron_length/g"
~/GGOSS/Scripts/RunBLAST.sh
else
  sed -i -e "s/max_intron_length \+//g" ~/GGOSS/Scripts/RunBLAST.sh
fi

echo "Completed applying of base tBLASTx settings..."
fi

echo "BLAST settings configured to user specifications"
echo "Running BLAST on user specified files using user specified settings"
~/GGOSS/Scripts/RunBLAST.sh
pkill yad
fi

fi

```

```
#####----- Run End -----#####
```

```
#####---- Blast re-open this script after selections if requested ----#####
```

```
if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then  
Blast_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/Blast_OpenMenu.txt)  
fi  
Blast_OpenMenu=$(( $Blast_OpenMenuPrep + 1 ))
```

```
if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then  
rm ~/GGOSS/tmp/Blast_OpenMenu.txt  
fi
```

```
if [ $Blast_OpenMenu = 2 ];then  
~/GGOSS/Buttons/ButtonBlast.sh  
fi
```

```
if [ -f ~/GGOSS/tmp/Blast_OpenMenu.txt ];then  
rm ~/GGOSS/tmp/Blast_OpenMenu.txt  
fi
```

```
#####----- Settings End -----#####
```

10.9.2.4.9 Ragout Main menu, settings, file selection

```
#!/bin/bash
```

```
#####----- File Selection start -----#####
```

```
if [ -f ~/GGOSS/tmp/RagoutFileSelection.txt ] || [ -f  
~/GGOSS/tmp/CheckpointRagout_SelectedFiles.txt ];then  
if [ -f ~/GGOSS/tmp/RagoutFileSelection.txt ];then  
RagoutFileSelection=$(head -n1 ~/GGOSS/tmp/RagoutFileSelection.txt)  
rm ~/GGOSS/tmp/RagoutFileSelection.txt  
else  
RagoutFileSelection=2  
fi
```

```
if [ "$RagoutFileSelection" = 1 ];then
```

```
if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then  
rm ~/GGOSS/tmp/Ragout_OpenMenu.txt  
fi
```

```
if [ -f ~/GGOSS/tmp/Ragout_SelectedFiles.txt ];then  
rm ~/GGOSS/tmp/Ragout_SelectedFiles.txt  
fi
```

```
if [ -f ~/GGOSS/tmp/RagoutSelectedFilesP4.txt ];then  
rm ~/GGOSS/tmp/RagoutSelectedFilesP4.txt  
fi
```

```
SelectFile=$(ls ~/GGOSS_InputOutput/)
```

```

    echo "$SelectFile" | yad --title="
--file selection          GENOME SEQUENCING PROGRAM - Ragout
          Created by Giles Holt" --list --column="Select files to re-assemble against ref
genome" --multiple --width 800 --height 600 --center --align=center --on-top --button="OK" --
button="Previous":1 --separator=" --filename=~//GGOSS/tmp/Ragout_SelectedFiles.txt >
~/GGOSS/tmp/Ragout_SelectedFiles.txt

```

```

#this means in the case of script exit do 1 of the following

```

```

mode="$?"
case $mode in
    1)echo "1" > ~/GGOSS/tmp/Ragout_OpenMenu.txt ;;
esac

```

```

if [ -s ~/GGOSS/tmp/Ragout_SelectedFiles.txt ];then
echo "1" > ~/GGOSS/tmp/CheckpointRagout_SelectedFiles.txt
fi

```

```

if [ -s ~/GGOSS/tmp/CheckpointRagout_SelectedFiles.txt ];then

```

```

    if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/Ragout_OpenMenu.txt
    fi
    if [ -f ~/GGOSS/tmp/RagoutFileSelection.txt ];then
    rm ~/GGOSS/tmp/RagoutFileSelection.txt
    fi

```

```

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/Ragout_SelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

```

```

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

```

```

    echo "$SelectFile2" | yad --title="
SEQUENCING PROGRAM - Ragout file selection          GENOME
          Created by Giles Holt" --list --
column="Select files to re-assemble against ref genome" --multiple --width 800 --height 600 --center -
-align=center --button="OK" --button="Previous":2 --separator=" >
~/GGOSS/tmp/RagoutSelectedFilesP4.txt

```

```

mode="$?"
case $mode in
    2)rm ~/GGOSS/tmp/CheckpointRagout_SelectedFiles.txt && echo "1" >
~/GGOSS/tmp/RagoutFileSelection.txt ;;
esac

```

```

if [ -s ~/GGOSS/tmp/RagoutSelectedFilesP4.txt ];then
rm ~/GGOSS/tmp/RagoutFileSelection.txt
rm ~/GGOSS/tmp/CheckpointRagout_SelectedFiles.txt
fi

```

```

cat ~/GGOSS/tmp/RagoutFileSelection.txt

```

```

echo "1" > ~/GGOSS/tmp/Ragout_OpenMenu.txt

```

```

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/RagoutSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

fi

fi

#####----- File Selection End -----#####

#####----- Ragout menu -----#####

if [ -f ~/GGOSS/tmp/RagoutFileSelection.txt ];then
echo ""
else

if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Ragout_OpenMenu.txt
fi

ICON=~/GGOSS/Pictures/DNA2_1000.jpg

yad --title="
Created by Giles Holt" --text="
                                GENOME SEQUENCING PROGRAM -- Ragout
                                Ragout

Can be used to re-assemble
to reference genomes" / --center --image=$ICON image-on-top --size=fit --center --
button="Previous":4 --button="
                                Run
                                ":0 --button="
                                Select samples
":1 --button="
                                Ragout settings
                                ":2 --button="
                                Ragout Manual
                                ":3 --
buttons-layout=center

mode="$?"
case $mode in
0)Choice=1 ;;
1)echo "1" > ~/GGOSS/tmp/RagoutFileSelection.txt && echo "1" >
~/GGOSS/tmp/Ragout_OpenMenu.txt ;;
2)Choice=3 ;;
3)Choice=4 ;;
4)~/GGOSS/GenomicsProgram.sh
esac

if [ "$Choice" = 1 ];then
echo "not built 1 yet"
fi

if [ "$Choice" = 2 ];then
echo "not built 2 yet"
fi

#####----- Ragout menu End -----#####

```

```
#####----- Settings Start-----#####
```

```
if [ "$Choice" = 3 ];then
```

```
yad --title="                                GENOME SEQUENCING PROGRAM -- Ragout
Settings      Created by Giles Holt" --center --size=fit --center --form \
--field="                                           Ragout Type:":CB \
--field="                                           Align to:":CB \
--field="                                Optimize for queries shorter (optional):":CB \
--field="                                evaluate - Expect value (E) for saving hits (optional):" \
--field="                                Number of threads (optional):" \
--field="                                Location on the subject sequence (Format: start-stop) (subject_loc,
optional):" \
--field="                                Show NCBI GIs in report (show_gis,
optional):":CB \
--field="                                Number of db sequences to show one-line descriptions for
(num_descriptions, optional):" \
--field="                                Number of database sequences to show alignments for
(num_alignments, optional):" \
--field="                                Number of aligned sequences to keep. Not compatible with num_des or
num_align (optional):" \
--field="                                Max No. of HSPs (alignments) to keep for any single query-subject pair
(max_hsps, optional):" \
--field="                                Produce HTML output (html, optional):":CB
\
--field="                                Restrict search of database to GI's listed in this file. Local only (gilist,
optional):":CB \
--field="Restrict search of database to everything except GI's listed in this file. Local only.
negative_gilist, optional):":CB \
--field="                                Restrict search with the given Entrez query. Remote searches only
(entrez_query, optional):":CB \
--field="                                Delete a hit that is enveloped by at least this many higher-scoring hits
(culling_limit, optional):" \
--field="                                Best Hit algorithm overhang value (recommended value: 0.1)
(best_hit_overhang, optional):" \
--field="                                Best Hit algorithm score edge value (recommended value: 0.1)
(best_hit_score_edge, optional):" \
--field="                                Effective size of the database (dbsize, optional):" \
--field="                                Effective length of the search space (searchsp, optional):"
\
--field="                                Search strategy file to read (import_search_strategy,
optional):" \
--field="                                Record search strategy to this file (export_search_strategy,
optional):" \
--field="                                Parse query and subject bar delimited sequence identifiers (parse_deflines,
optional):":CB \
--field="                                Execute search on NCBI servers? (remote,
optional):":CB \
--field="                                Alignment view options (outfmt, optional):" \
"blastn!blastp!tblastn!tblastp!blastx!tblastx!rpsblast!megablast!dc-megablast!" "Viral
Database!Bacterial Database!Fungal Database!Custom Database" "No!Yes!" '-' '-' "No!Yes!" '-' '-'
-' "No!Yes!" "No!Yes!" "No!Yes!" "No!Yes!" '-' '-' '-' "No!Yes!" "No!Yes!" '-' \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/RagoutSettingsChange.txt > ~/.GGOSS/tmp/RagoutSettingsChange.txt
```

```

mode="$?"
  case $mode in
    0)echo "1" > ~/GGOSS/tmp/Ragout_OpenMenu.txt ;;
    1)echo "1" > ~/GGOSS/tmp/Ragout_OpenMenu.txt ;;
  esac

fi

fi

#####----- Settings End -----#####

#####---- Ragout re-open this script after selections if requested ----#####

if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then
Ragout_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/Ragout_OpenMenu.txt)
fi
Ragout_OpenMenu=$(( $Ragout_OpenMenuPrep + 1 ))

if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Ragout_OpenMenu.txt
fi

if [ $Ragout_OpenMenu = 2 ];then
~/GGOSS/Buttons/ButtonRagout.sh
fi

if [ -f ~/GGOSS/tmp/Ragout_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Ragout_OpenMenu.txt
fi

```

10.9.2.5 Annotation

10.9.2.5.1 Annotation main menu

```
#!/bin/bash

ICON=~/.GGOSS/Pictures/DNA4_900.jpg

yad --title="
Annotation Created by Giles Holt" --width=800 --height=500 --center --image=$ICON --
image-on-top --size=fit --center --button="
Prokka ":0 --
button="Artemis":1 --button="Previous":3 --buttons-layout=center

mode="$?"
case $mode in
  0)~/GGOSS/Buttons/ButtonProkka.sh ;;
  1)~/GGOSS/Scripts/Artemis.sh ;;
  3)~/GGOSS/GenomicsProgram.sh ;;
esac
```

10.9.2.5.2 Prokka Main menu

```
#!/bin/bash

ICON=~/.GGOSS/Pictures/DNA4_900.jpg

yad --title="
Created by Giles Holt" --width=800 --height=500 --center --image=$ICON --image-on-top --size=fit -
-center --button="
Run ":3 --button="
File Selection ":4 --button="
Settings ":5 --button="
Previous ":6 --buttons-layout=center

mode="$?"
case $mode in
  3)~/GGOSS/Scripts/Prokka.sh ;;
  4)~/GGOSS/Buttons/ProkkaSelectFile.sh ;;
  5)~/GGOSS/Buttons/Prokka_settingsmenu.sh ;;
  6)~/GGOSS/Buttons/ButtonAnnotation.sh ;;
esac
```

10.9.2.5.3 Prokka file selection

```
#!/bin/sh

if [ -f ~/.GGOSS/tmp/ProkkaSelectedFiles.txt ]
then
rm ~/.GGOSS/tmp/ProkkaSelectedFiles.txt
fi

if [ -f ~/.GGOSS/tmp/ProkkaSelectedFilesP4.txt ]
```

```

then
rm ~/GGOSS/tmp/ProkkaSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="
                                GENOME SEQUENCING
PROGRAM                        Created by Giles Holt" --list --column="Select file containing input" --
multiple --width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --
separator=" > ~/GGOSS/tmp/ProkkaSelectedFiles.txt

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/ButtonProkka.sh ;;
  esac

  if [ -s ~/GGOSS/tmp/ProkkaSelectedFiles.txt ]
then

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/ProkkaSelectedFiles.txt)
echo "$HOME/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="
                                GENOME SEQUENCING
PROGRAM                        Created by Giles Holt" --list --column="Select input files" --multiple --
width 800 --height 600 --center --align=center --button="OK" --button="Previous":2 --separator=" >
~/GGOSS/tmp/ProkkaSelectedFilesP4.txt

mode="$?"
case $mode in
  2)~/GGOSS/Buttons/ProkkaSelectFile.sh ;;
  esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/ProkkaSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/ProkkaSelectedFilesP4.txt ];then
~/GGOSS/Buttons/ButtonProkka.sh
fi

```

10.9.2.5.4 Prokka Settings

```

#!/bin/bash

reuse=1
recycledSettings=1
date=$(date +"%F")
ICON=~/.GGOSS/Pictures/DNA4_900.jpg

mode="$?"

```

```

case $mode in
  1)~/GGOSS/Buttons/ButtonProkka.sh ;;
  2)~/GGOSS/Buttons/Prokka_settingsmenu.sh ;;
  3)reuse=2 ;;
esac |\

yad --title="                                GENOME SEQUENCING PROGRAM -- Prokka
Settings      Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --
form \
--field="Input file type":CB \
--field="Specify data from file type for annotation:":CB \
--field="Top how many nodes(Only change if node seperation selected and you don't want all nodes
run):": \
--field="Minimum node size (Only change if node seperation selected and you don't want all nodes
run):": \
--field="Minimum node coverage (Only change if node seperation selected and you don't want all
nodes run):": \
'contigs.fasta!scaffolds.fasta!' "Run file type as a whole!Run nodes of each sample seperately by
criteria given!" 'Not applicable' 'Not applicable' 'Not applicable' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="Re-use":3 --button="Reset":2 --
button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/ProkkaSettingsChange.txt >
~/GGOSS/tmp/"$date"_ProkkaSettingsChange.txt

if (( "$reuse" == 2 ));then

SelectFile=$(ls ~/GGOSS/tmp/ | grep "ProkkaSettingsChange.txt")

echo "$SelectFile" | yad --title="                                GENOME SEQUENCING PROGRAM - Recycle
previous settings      Created by Giles Holt" --list --column="Select a previous settings file you
wish to re-use" --multiple --width 800 --height 600 --center --align=center --button="Select":2 --
button="Previous":1 --separator=" > ~/GGOSS/tmp/reuse_ProkkaSetting.txt

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/Prokka_settingsmenu.sh ;;
  2)recycledSettings=2 ;;
esac

#need to edit so that this includes the selected previous settings
if (( "$recycledSettings" == 2 ));then

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/ButtonProkka.sh ;;
  2)~/GGOSS/Buttons/Prokka_settingsmenu.sh ;;
  3)reuse=2 ;;
esac |\

yad --title="                                GENOME SEQUENCING PROGRAM -- Prokka
Settings      Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --
form \

```

```

--field="Input file type":CB \
--field="Specify data from file type for annotation:":CB \
--field="Top how many nodes(Only change if node seperation selected and you don't want all nodes
run):": \
--field="Minimum node size (Only change if node seperation selected and you don't want all nodes
run):": \
--field="Minimum node coverage (Only change if node seperation selected and you don't want all
nodes run):": \
'contigs.fasta!scaffolds.fasta!' "Run file type as a whole!Run nodes of each sample seperately by
criteria given!" 'Not applicable' 'Not applicable' 'Not applicable' \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="Re-use":3 --button="Reset":2 --
button="gtk-close:1" --buttons-layout=center --editable --
filename=~/.GGOSS/tmp/ProkkaSettingsChange.txt >
~/GGOSS/tmp/"$date"_ProkkaSettingsChange.txt

```

```
fi
```

```
~/GGOSS/Buttons/ButtonProkka.sh
```

10.9.2.5.5 Artemis script connection through GGOSS

```
#!/bin/bash
```

```
~/artemis/art
```

10.9.2.6 Community analysis

10.9.2.6.1 Bacterial community analysis main menu

```
#!/bin/bash
```

```
ICON=~/.GGOSS/Pictures/bacteria_taxonomy_900.jpg
```

```

yad --title="
GENOME SEQUENCING PROGRAM
Created by Giles Holt" --width=800 --height=500 --center --image=$ICON --image-on-top --size=fit -
-center --button="Mothur":0 --button="Qiime":3 --button="Post Mothur analysis":1 --button="Post
Qiime analysis":2 --button="Previous":4 --buttons-layout=center

```

```
mode="$?"
```

```

case $mode in
0)~/GGOSS/Buttons/ButtonMothur.sh ;;
1)~/GGOSS/Buttons/ButtonPostMothurAnalysis.sh ;;
2)~/GGOSS/Buttons/ButtonPostQiimeAnalysis.sh ;;
3)~/GGOSS/Buttons/ButtonMothur.sh ;;
4)~/GGOSS/GenomicsProgram.sh ;;

```

```
Esac
```

10.9.2.6.2 Mothur

```
#!/bin/bash

MothurChecklist=2
SettingsSelect=2
SelectedAll=2
ApplySettingsThenRun=2

ICON=~/.GGOSS/Pictures/bacteria_taxonomy_900.jpg

if [ ! -e ~/.GGOSS/tmp/Mothur_ChecklistBlank.txt ];then

yad --title="GENOME SEQUENCING PROGRAM -- Mothur          Created by Giles Holt" --
width=800 --height=500 --center --image=$ICON --image-on-top --size=fit --center --
button="Previous":2 --button="Select samples":4 --button="Run":0 --button="Select mothur steps":1
--button="Settings Menu":3 --buttons-layout=center

mode="$?"
  case $mode in
    0)ApplySettingsThenRun=1 ;;
    1)MothurChecklist=1 ;;
    2)~/.GGOSS/Buttons/Button16S_Analysis.sh ;;
    3)SettingsSelect=1 ;;
    4)~/.GGOSS/Scripts/FileSelection/Mothurfileselect.sh ;;
  esac

fi

#####---- Mothur Settings ----#####

if [ "$SettingsSelect" = 1 ];then

  if [ -f ~/.GGOSS/tmp/Mothur_ChecklistBlank.txt ];then
    rm ~/.GGOSS/tmp/Mothur_ChecklistBlank.txt
  fi
  path=~/.

yad --title="
                                GENOME SEQUENCING PROGRAM -- Mothur
Settings          Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --on-
top --form \
--field="No. of processors to use" \
--field="Amount of Ram to commit" \
--field="Reference Database":CB \
--field="Custom reference database (e.g. ${path}referenceDB.txt)" \
'-' ' ' "SILVA!Other one!Fungal" '-' --text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.GGOSS/tmp/MothurSettingsChange.txt >
~/.GGOSS/tmp/MothurSettingsChange.txt

mode="$?"
  case $mode in
    0)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
    1)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
```

```

    esac

fi

#####---- Mothur Checklist ----#####

if [ "$MothurChecklist" = 1 ] || [ -f ~/GGOSS/tmp/Mothur_ChecklistBlank.txt ];then
    if [ -f ~/GGOSS/tmp/Mothur_ChecklistBlank.txt ];then
        rm ~/GGOSS/tmp/Mothur_ChecklistBlank.txt
    fi

    if [ -f ~/GGOSS/tmp/Mothur_OpenMenu.txt ];then
        rm ~/GGOSS/tmp/Mothur_OpenMenu.txt
    fi

yad --title="                                GENOME SEQUENCING PROGRAM
Created by Giles Holt" --center --height="600" --width="1000" --on-top --list --print-all \
--column=A:chk --column="Reducing sequencing and PCR errors:" \
--column=B:chk --column="Processing improved sequences:" \
--column=C:chk --column="Assessing error rates:" \
--column=D:chk --column="Preparing for analysis:" \
--column=E:chk --column="OTUs:" \
--column=F:chk --column="Phylotypes:" \
--column=G:chk --column="Phylogenetic:" \
--column=H:chk --column="Analysis:" \
--column=I:chk --column="OTU-based analysis:" \
false "A 1.Make Contigs" false "B 1.Unique seqs" false "C 1.Get groups" \
false "D 1.Remove groups" false "E 1.Dist seqs" false "F 1.Phylotype" false "G 1.Dist seqs" false "H
1.System" false "I 1.Unique seqs" false "A 2.Summary seqs" false "B 2.Count seqs" false "C 2.Seq
error" false "D 2.N/A" false "E 2.Make shared" false "F 2.Clearcut" false "G 2.System" false "H
2.System" false "I 2.Count seqs" false "A 3.Screen seqs" false "B 3.Summary seqs" false "C 3.Dist
seqs" false "D 3.N/A" false "E 3.Classify OTU" false "F 3.Count groups" false "G 3.N/A" false "H
3.Count groups" false "I 3.Summary seqs" false "A 4.Get current" false "B 4.PCR seqs" false "C
4.Cluster" false "D 4.N/A" false "E 4.Sub sample" false "F 4.N/A" false "G 4.N/A" false "H 4.Sub
sample" false "I 4.PCR seqs" false "A 5.Final section summary seqs" false "B 5.System" false "C
5.Make shared" false "D 5.N/A" false "E 5.System" false "F N/A" false "G 5.N/A" false "H 5.N/A"
false "I 5.System" false "A 6.N/A" false "B 6.Summary seqs" false "C 6.Rarefaction single" false "D
6.N/A" false "E 6.N/A" false "F 6.N/A" false "G 6.N/A" false "H 6.N/A" false "I 6.Summary seqs"
false "A 7.N/A" false "B 7.Align seqs" false "C 7.N/A" false "D 7.N/A" false "E 7.N/A" false "F
7.N/A" false "G 7.N/A" false "H 7.N/A" false "I 7.Align seqs" false "A 8.N/A" false "B 8.Summary
seqs" false "C 8.N/A" false "D 8.N/A" false "E 8.N/A" false "F 8.N/A" false "G 8.N/A" false "H
8.N/A" false "I 8.Summary seqs" false "A 9.N/A" false "B 9.Screen seqs" false "C 9.N/A" false "D
9.N/A" false "E 9.N/A" false "F 9.N/A" false "G 9.N/A" false "H 9.N/A" false "I 9.Screen seqs" false
"A 10.N/A" false "B 10.Summary seqs" false "C 10.N/A" false "D 10.N/A" false "E 10.N/A" false "F
10.N/A" false "G 10.N/A" false "H 10.N/A" false "I 10.Summary seqs" false "A 11.N/A" false "B
11.Filter seqs" false "C 11.N/A" false "D 11.N/A" false "E 11.N/A" false "F 11.N/A" false "G
11.N/A" false "H 11.N/A" false "I 11.Filter seqs" false "A 12.N/A" false "B 12.Unique seqs" false "C
12.N/A" false "D 12.N/A" false "E 12.N/A" false "F 12.N/A" false "G 12.N/A" false "H 12.N/A" false
"I 12.Unique seqs" false "A 13.N/A" false "B 13.Pre cluster" false "C 13.N/A" false "D 13.N/A" false
"E 13.N/A" false "F 13.N/A" false "G 13.N/A" false "H 13.N/A" false "I 13.Pre cluster" false "A
14.N/A" false "B 14.Chimera uchime" false "C 14.N/A" false "D 14.N/A" false "E 14.N/A" false "F
14.N/A" false "G 14.N/A" false "H 14.N/A" false "I 14.Chimera uchime" false "A 15.N/A" false "B
15.Remove seqs" false "C 15.N/A" false "D 15.N/A" false "E 15.N/A" false "F 15.N/A" false "G

```

```

15.N/A" false "H 15.N/A" false "I 15.Remove seqs" false "A 16.N/A" false "B 16.Summary seqs"
false "C 16.N/A" false "D 16.N/A" false "E 16.N/A" false "F 16.N/A" false "G 16.N/A" false "H
16.N/A" false "I 16.Summary seqs" false "A 17.N/A" false "B 17.Classify seqs" false "C 17.N/A"
false "D 17.N/A" false "E 17.N/A" false "F 17.N/A" false "G 17.N/A" false "H 17.N/A" false "I
17.Classify seqs" false "A 18.N/A" false "B 18.Remove lineage" false "C 18.N/A" false "D 18.N/A"
false "E 18.N/A" false "F 18.N/A" false "G 18.N/A" false "H 18.N/A" false "I 18.Remove lineage" \
--button="Select all":2 --button="gtk-save:0" --button="gtk-close:1" --editable --
filename=~/.GGOSS/tmp/MothurCheckListChange.txt > ~/.GGOSS/tmp/MothurCheckListChange.txt

```

```

mode="$?"
case $mode in
0)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
1)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
2)SelectedAll=1 ;;
esac

```

```

if [ "$SelectedAll" = 1 ];then
echo "checklist Fully selected"
if [ -f ~/.GGOSS/tmp/Mothur_OpenMenu.txt ];then
rm ~/.GGOSS/tmp/Mothur_OpenMenu.txt
fi

```

```

yad --title="
GENOME SEQUENCING PROGRAM
Created by Giles Holt" --center --height="600" --width="1000" --on-top --list --print-all \
--column=A:chk --column="Reducing sequencing and PCR errors:" \
--column=B:chk --column="Processing improved sequences:" \
--column=C:chk --column="Assessing error rates:" \
--column=D:chk --column="Preparing for analysis:" \
--column=E:chk --column="OTUs:" \
--column=F:chk --column="Phylotypes:" \
--column=G:chk --column="Phylogenetic:" \
--column=H:chk --column="Analysis:" \
--column=I:chk --column="OTU-based analysis:" \
true "A 1.Make Contigs" true "B 1.Unique seqs" true "C 1.Get groups" \
true "D 1.Remove groups" true "E 1.Dist seqs" true "F 1.Phylotype" true "G 1.Dist seqs" true "H
1.System" true "I 1.Unique seqs" true "A 2.Summary seqs" true "B 2.Count seqs" true "C 2.Seq error"
false "D 2.N/A" true "E 2.Make shared" true "F 2.Clearcut" true "G 2.System" true "H 2.System" true
"I 2.Count seqs" true "A 3.Screen seqs" true "B 3.Summary seqs" true "C 3.Dist seqs" false "D
3.N/A" true "E 3.Classify OTU" true "F 3.Count groups" false "G 3.N/A" true "H 3.Count groups"
true "I 3.Summary seqs" true "A 4.Get current" true "B 4.PCR seqs" true "C 4.Cluster" false "D
4.N/A" true "E 4.Sub sample" false "F 4.N/A" false "G 4.N/A" true "H 4.Sub sample" true "I 4.PCR
seqs" true "A 5.Final section summary seqs" true "B 5.System" true "C 5.Make shared" false "D
5.N/A" true "E 5.System" false "F N/A" false "G 5.N/A" false "H 5.N/A" true "I 5.System" false "A
6.N/A" true "B 6.Summary seqs" true "C 6.Rarefaction single" false "D 6.N/A" false "E 6.N/A" false
"F 6.N/A" false "G 6.N/A" false "H 6.N/A" true "I 6.Summary seqs" false "A 7.N/A" true "B 7.Align
seqs" false "C 7.N/A" false "D 7.N/A" false "E 7.N/A" false "F 7.N/A" false "G 7.N/A" false "H
7.N/A" true "I 7.Align seqs" false "A 8.N/A" true "B 8.Summary seqs" false "C 8.N/A" false "D
8.N/A" false "E 8.N/A" false "F 8.N/A" false "G 8.N/A" false "H 8.N/A" true "I 8.Summary seqs"
false "A 9.N/A" true "B 9.Screen seqs" false "C 9.N/A" false "D 9.N/A" false "E 9.N/A" false "F
9.N/A" false "G 9.N/A" false "H 9.N/A" true "I 9.Screen seqs" false "A 10.N/A" true "B 10.Summary
seqs" false "C 10.N/A" false "D 10.N/A" false "E 10.N/A" false "F 10.N/A" false "G 10.N/A" false
"H 10.N/A" true "I 10.Summary seqs" false "A 11.N/A" true "B 11.Filter seqs" false "C 11.N/A" false
"D 11.N/A" false "E 11.N/A" false "F 11.N/A" false "G 11.N/A" false "H 11.N/A" true "I 11.Filter

```

```

seqs" false "A 12.N/A" true "B 12.Unique seqs" false "C 12.N/A" false "D 12.N/A" false "E 12.N/A"
false "F 12.N/A" false "G 12.N/A" false "H 12.N/A" true "I 12.Unique seqs" false "A 13.N/A" true "B
13.Pre cluster" false "C 13.N/A" false "D 13.N/A" false "E 13.N/A" false "F 13.N/A" false "G
13.N/A" false "H 13.N/A" true "I 13.Pre cluster" false "A 14.N/A" true "B 14.Chimera uchime" false
"C 14.N/A" false "D 14.N/A" false "E 14.N/A" false "F 14.N/A" false "G 14.N/A" false "H 14.N/A"
true "I 14.Chimera uchime" false "A 15.N/A" true "B 15.Remove seqs" false "C 15.N/A" false "D
15.N/A" false "E 15.N/A" false "F 15.N/A" false "G 15.N/A" false "H 15.N/A" true "I 15.Remove
seqs" false "A 16.N/A" true "B 16.Summary seqs" false "C 16.N/A" false "D 16.N/A" false "E
16.N/A" false "F 16.N/A" false "G 16.N/A" false "H 16.N/A" true "I 16.Summary seqs" false "A
17.N/A" true "B 17.Classify seqs" false "C 17.N/A" false "D 17.N/A" false "E 17.N/A" false "F
17.N/A" false "G 17.N/A" false "H 17.N/A" true "I 17.Classify seqs" false "A 18.N/A" true "B
18.Remove lineage" false "C 18.N/A" false "D 18.N/A" false "E 18.N/A" false "F 18.N/A" false "G
18.N/A" false "H 18.N/A" true "I 18.Remove lineage" \
    --button="Deselect all":2 --button="gtk-save:0" --button="gtk-close:1" --editable --
filename=~/.GGOSS/tmp/MothurCheckListChange.txt > ~/.GGOSS/tmp/MothurCheckListChange.txt

mode="$?"
case $mode in
    0)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
    1)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt ;;
    2)echo "1" > ~/.GGOSS/tmp/Mothur_OpenMenu.txt && echo "1" >
~/GGOSS/tmp/Mothur_ChecklistBlank.txt ;;
    esac

fi

fi

#####---- Mothur File Select ----#####

#####---- Mothur Apply Settings and run ----#####

if [ "$ApplySettingsThenRun" = 1 ];then
    if [ -f ~/.GGOSS/tmp/Mothur_ChecklistBlank.txt ];then
        rm ~/.GGOSS/tmp/Mothur_ChecklistBlank.txt
    fi
    if [ -f ~/.GGOSS/tmp/Mothur_OpenMenu.txt ];then
        rm ~/.GGOSS/tmp/Mothur_OpenMenu.txt
    fi
fi

cp ~/.GGOSS/Scripts/Mothur.sh ~/.GGOSS/Scripts/MothurRun.sh

Processors=$(nproc)
A1MakeContigs=$(awk -F '|' 'NR==1 {print $1}' ~/.GGOSS/tmp/MothurCheckListChange.txt)

if [ "$A1MakeContigs" == "TRUE" ];then
sed -e "s/makecontigs\+\/make.contigs(file=stability.files, processors=$Processors)/g"
~/GGOSS/Scripts/MothurRun.sh > ~/.GGOSS/tmp/tmp.txt
mv ~/.GGOSS/tmp/tmp.txt ~/.GGOSS/Scripts/MothurRun.sh
fi

```

```
A2SummarySeqs=$(awk -F '|' 'NR==2 {print $1}' ~/GGOSS/tmp/MothurCheckListChange.txt)
```

```
if [ "$A2SummarySeqs" == "TRUE" ];then  
sed -e "s/summaryseqs\+/summary.seqs(fasta=stability.trim.contigs.fasta)/g"  
~/GGOSS/Scripts/MothurRun.sh > ~/GGOSS/tmp/tmp.txt  
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/MothurRun.sh  
fi
```

```
A3ScreenSeqs=$(awk -F '|' 'NR==3 {print $1}' ~/GGOSS/tmp/MothurCheckListChange.txt)  
NumberOfColumnsSoFar=$(awk ';' 'NF {print}')  
TakeLastListOfFiles=$(awk -F ';' -v x=$NumberOfColumnsSoFar '{print $x}'  
~/GGOSS/Scripts/MothurRun.sh | awk -F '(' '{print $2}' | awk -F ')' '{print $1}')
```

```
if [ "$A3ScreenSeqs" == "TRUE" ];then  
sed -e "s/screenseqs\+/screen.seqs($TakeLastListOfFiles, group=stability.contigs.groups,  
maxambig=0, maxlength=275)/g" ~/GGOSS/Scripts/MothurRun.sh > ~/GGOSS/tmp/tmp.txt  
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/MothurRun.sh  
fi
```

```
A4SummarySeqs=$(awk -F '|' 'NR==4 {print $1}' ~/GGOSS/tmp/MothurCheckListChange.txt)
```

```
if [ "$A4SummarySeqs" == "TRUE" ];then  
sed -e "s/summaryseqs\+/summary.seqs()/g" ~/GGOSS/Scripts/MothurRun.sh >  
~/GGOSS/tmp/tmp.txt  
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/MothurRun.sh  
fi
```

```
####check this as says get current in settings list
```

```
A5UniqueSeqs=$(awk -F '|' 'NR==4 {print $1}' ~/GGOSS/tmp/MothurCheckListChange.txt)
```

```
if [ "$A5UniqueSeqs" == "TRUE" ];then  
sed -e "s/uniqueseqs\+/unique.seqs(fasta=stability.trim.contigs.good.fasta)/g"  
~/GGOSS/Scripts/MothurRun.sh > ~/GGOSS/tmp/tmp.txt  
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/MothurRun.sh  
fi
```

```
#Run Script  
~/GGOSS/Scripts/Mothur.sh
```

```
if [ -f ~/GGOSS/Scripts/MothurRun.sh ];then  
rm ~/GGOSS/Scripts/MothurRun.sh  
fi
```

```
fi
```

```
#####---- Mothur return to/Open menu after selections ----#####
```

```
if [ -f ~/GGOSS/tmp/Mothur_OpenMenu.txt ];then  
Mothur_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/Mothur_OpenMenu.txt)  
fi  
Mothur_OpenMenu=$(( $Mothur_OpenMenuPrep + 1 ))
```

```
if [ -f ~/GGOSS/tmp/Mothur_OpenMenu.txt ];then
```

```

rm ~/GGOSS/tmp/Mothur_OpenMenu.txt
fi

if [ $Mothur_OpenMenu = 2 ];then
~/GGOSS/Buttons/ButtonMothur.sh
fi

if [ -f ~/GGOSS/tmp/Mothur_OpenMenu.txt ];then
rm ~/GGOSS/tmp/Mothur_OpenMenu.txt
fi

```

10.9.2.6.3 MEGAN

```

#!/bin/bash
ICON=~/.GGOSS/Pictures/MEGAN1_800.jpg

yad --title="
                GENOME SEQUENCING PROGRAM -- MEGAN
Created by Giles Holt" --text-align=center --window-icon=Giles_Holt GENOME SEQUENCING
PROGRAM --center --image=$ICON image-on-top --size=fit --center --button="MEGAN Viral
Taxonomy":0 --button="Previous":2 --buttons-layout=center

mode="$?"
case $mode in
0)~/GGOSS/Scripts/FileSelection/MEGANselectfile.sh ;;
2)~/GGOSS/GenomicsProgram.sh ;;
*)
esac

```

10.9.2.6.4 PIPITS Main menu, settings, file selection

```

#!/bin/bash

#####-----#####
##### PIPITS MENU #####
#####-----#####

if [ -f ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt ];then
PIPITS_FileSelectionPrevious=$(head -n1 ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt)
fi

PIPITS_FileSelectionPrevious=$(( $PIPITS_FileSelectionPrevious + 1 ))

PIPITSfileselect=2
PIPITS_SettingsSelected=2
ICON=~/.GGOSS/Pictures/PIPITS_image1.jpg

if [ $PIPITS_FileSelectionPrevious != 2 ];then
PIPITS_FileSelectionPrevious=3
if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
fi

```

```
yad --title="GENOME SEQUENCING PROGRAM -- PIPITS                               GGOSS - Created
by Giles Holt" --center --image=$ICON --size=fit --center --fixed --button=" Previous ":4 --
button=" Run PIPITS ":6 --button=" Select samples ":5 --button=" Settings ":7 --button="
PIPITS Manual " --buttons-layout=center --text="                               PIPITS
```

PIPITS is an automated pipeline
for analyses of fungal internal
transcribed spacer (ITS) sequences
from the Illumina sequencing platform."

```
mode="$?"
case $mode in
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)PIPITSfileselect=1 ;;
  6)~/GGOSS/Scripts/PIPITS_script.sh ;;
  7)PIPITS_SettingsSelected=1 ;;
esac
```

```
#####---- PIPITS SETTINGS ----#####
```

```
if [ $PIPITS_SettingsSelected = 1 ];then
  if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
  fi
```

```
mode="$?"
case $mode in
  0)echo "1" > ~/GGOSS/tmp/PIPITS_OpenMenu.txt ;;
  1)echo "1" > ~/GGOSS/tmp/PIPITS_OpenMenu.txt ;;
esac | \
```

```
yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM -- PIPITS Settings      GGOSS -
Created by Giles Holt" --image=$ICON --image-on-top --on-top --center --size=fit --center --form \
--field="Read type:":CB \
--field="ITS type:":CB \
--field="Maximum memory:" \
--field="Output table for FUNGuild analysis:":CB \
--field="Output file name:" \
"Paired read!Single read!" "ITS1!ITS2!" "4" "No!Yes!" "Experiment Name" \
--text-info --show-uri --center --wrap \
--button="Previous":1 --button="gtk-save:0" --editable --
filename=~/GGOSS/tmp/PIPITS_SettingsChange.txt > ~/GGOSS/tmp/PIPITS_SettingsChange.txt
```

```
fi
```

```
else
if [ -f ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt ];then
rm ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt
fi
```

```
if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
fi
```

```
PIPITSfileselect=1
```

fi

```
#####---- PIPITS File Select ----#####
```

```
if [ $PIPITSfileselect = 1 ];then
  if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
  fi
```

```
if [ -f ~/GGOSS/tmp/PIPITSSelectedFiles.txt ]
then
  rm ~/GGOSS/tmp/PIPITSSelectedFiles.txt
fi
```

```
if [ -f ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt ]
then
  rm ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt
fi
```

```
SelectFile=$(ls ~/GGOSS_InputOutput/)
```

```
echo "$SelectFile" | yad --title="          GENOME SEQUENCING PROGRAM - PIPITS file
selection          Created by Giles Holt" --list --column="Select files you wish to run" --multiple --
width 800 --height 600 --center --align=center --button="Open" --button="Previous":1 --separator=" >
~/GGOSS/tmp/PIPITSSelectedFiles.txt
```

```
mode="$?"
  case $mode in
    1)echo "3" > ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/PIPITS_OpenMenu.txt ;;
    esac
```

```
if [ -s ~/GGOSS/tmp/PIPITSSelectedFiles.txt ];then
  if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
  fi
  FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PIPITSSelectedFiles.txt)
  echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt
```

```
SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")
```

```
echo "$SelectFile2" | yad --title="          GENOME SEQUENCING
PROGRAM - PIPITS file selection          Created by Giles Holt" --list --column="Select files you
wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt
```

```

mode="$?"
  case $mode in
    2)echo "1" > ~/GGOSS/tmp/PIPITS_FileSelectionPrevious.txt & echo "1" >
~/GGOSS/tmp/PIPITS_OpenMenu.txt ;;
    esac

  FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt)
  echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt ]
then
  if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
    rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
  fi
  echo "1" > ~/GGOSS/tmp/PIPITS_OpenMenu.txt
fi

fi

#####---- PIPITS return to/Open menu after selections ----#####

if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
PIPITS_OpenMenuPrep=$(head -n1 ~/GGOSS/tmp/PIPITS_OpenMenu.txt)
fi
PIPITS_OpenMenu=$(( $PIPITS_OpenMenuPrep + 1 ))

if [ $PIPITS_OpenMenu = 2 ];then
~/GGOSS/Buttons/PIPITS_Menu.sh
fi

if [ -f ~/GGOSS/tmp/PIPITS_OpenMenu.txt ];then
rm ~/GGOSS/tmp/PIPITS_OpenMenu.txt
fi

```

10.9.2.7 Unique GGOSS built tools

10.9.2.7.1 Conserved gene finder, Main menu, settings, file selection

```
#!/bin/bash
SettingsSelected=2
FileSelection=2

ICON=~/.GGOSS/Pictures/GGOSS_ConervedGeneFinder.jpg

yad --title="          GENOME SEQUENCING PROGRAM -- Conserved Sequence
Finder          Created by Giles Holt" --text="    Conserved Sequence Finder

A Tool for indentifying
conserved DNA sequences
across a any number of
genomes and/or gene
regions " / --text-align=fill --center --image=$ICON --size=fit --center --button=" Previous " :4 --
button=" Run " :3 --button="Select genome/gene files":6 --button="Settings":5 --
button="Conserved_Sequence_Finder Manual" --buttons-layout=center

mode="$?"
case $mode in
    3)~/GGOSS/Scripts/ConservedDNAFinder.sh ;;
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)SettingsSelected=1 ;;
    6)FileSelection=1 ;;
esac

if [ $SettingsSelected = 1 ];then

    if [ -f ~/GGOSS/tmp/GeneFinder_SettingsChange.txt ]
    then
    rm ~/GGOSS/tmp/GeneFinder_SettingsChange.txt
    fi

    mode="$?"
    case $mode in
        1)~/GGOSS/Buttons/Button_ConervedGeneFinder.sh ;;
    esac | \

yad --title="    GGOSS - Conserved Sequence Finder -- Settings          Created by Giles
Holt" --center --image=$ICON --image-on-top --size=fit --center --form \
--field="Minimum sequence length:" \
--field="Minimum % of files for a given sequence to be found in:" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --editable
--filename=~/.GGOSS/tmp/GeneFinder_SettingsChange.txt >
~/GGOSS/tmp/GeneFinder_SettingsChange.txt

~/GGOSS/Buttons/Button_ConervedGeneFinder.sh
```

```

fi

if [ $FileSelection = 1 ];then

echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

#this means in the case of script exit do 1 of the following
mode="$?"
  case $mode in
    3)~/GGOSS/Buttons/Button_ConservedGeneFinder.sh ;;
    esac | \

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="      GENOME SEQUENCING PROGRAM -
Conserved Sequence Finder --file selection      Created by Giles Holt" --list --column="Select
files in which to identify conserved sequences" --multiple --width 800 --height 600 --center --
align=center --on-top --button="Save selection:0" --button="Previous":3 --separator=" --
filename=~/GGOSS/tmp/ConservedGeneFinder_SelectedFiles.txt >
~/GGOSS/tmp/ConservedGeneFinder_SelectedFiles.txt &

~/GGOSS/Buttons/Button_ConservedGeneFinder.sh
fi

```

10.9.2.7.2 Viral taxa finder, Main menu, settings, file selection

```

#!/bin/bash

Settings=1
ViralLineageMainMenu=1
RunLineageFinder=1

ICON=~/GGOSS/Pictures/taxonomy2_1000.jpg

yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM          created
by Giles Holt" --center --image=$ICON --image-on-top --size=fit --center --button=" Run ":0 --
button="Select File":3 --button="Settings":4 --button="Previous":2 --buttons-layout=center

mode="$?"
  case $mode in
    0)~/GGOSS/Scripts/ViralLineageFinder.sh ;;
    2)~/GGOSS/GenomicsProgram.sh ;;
    3)~/GGOSS/Scripts/FileSelection/Viral_InfoFinderfileselect.sh ;;
    4)Settings=2 ;;
    esac

if [[ "${Settings}" = 2 ]];then

yad --title="          GGOSS - GENOMIC ANALYSIS PROGRAM --
Viral_InfoFinder Settings      Created by Giles Holt" --center --image=$ICON --image-on-top --
size=fit --center --form \
--field="Taxonomy":CB \
--field="Sum/Merge data based given taxa classification":CB \
"Phylum!Class!Order!Family!Genus!Species!Host" "No!Yes" \

```

```

--text-info --show-uri --width=600 --height=500 --center --wrap \
  --button="Previous":1 --button="gtk-save:0" --editable --
filename=~/.GGOSS/tmp/Viral_InfoFinderSettingsChange.txt >
~/GGOSS/tmp/Viral_InfoFinderSettingsChange.txt

mode="$?"
case $mode in
  0)ViralLineageMainMenu=2 ;;
  1)ViralLineageMainMenu=2 ;;
esac

fi

if [[ "${ViralLineageMainMenu}" = 2 ]];then
~/GGOSS/Buttons/ButtonGGOSSViralTaxaFinder.sh
fi

#!/bin/sh

if [ -f ~/GGOSS/tmp/ViralTaxaFinderSelectedFiles.txt ]
then
rm ~/GGOSS/tmp/ViralTaxaFinderSelectedFiles.txt
fi

if [ -f ~/GGOSS/tmp/ViralTaxaFinderSelectedFilesP4.txt ]
then
rm ~/GGOSS/tmp/ViralTaxaFinderSelectedFilesP4.txt
fi

SelectFile=$(ls ~/GGOSS_InputOutput/)

echo "$SelectFile" | yad --title="
ViralTaxaFinder file selection          GENOME SEQUENCING PROGRAM -
Created by Giles Holt" --list --column="Select files you wish to
run" --multiple --width 800 --height 600 --center --align=center --button="Open" --
button="Previous":1 --separator=" > ~/GGOSS/tmp/ViralTaxaFinderSelectedFiles.txt

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/ButtonGGOSSViralTaxaFinder.sh ;;
esac

if [ -s ~/GGOSS/tmp/ViralTaxaFinderSelectedFiles.txt ]
then

FolderSelected=$(awk 'NR=1 {print}' ~/GGOSS/tmp/ViralTaxaFinderSelectedFiles.txt)
echo "~/GGOSS_InputOutput/$FolderSelected" > ~/GGOSS/tmp/Path2selectedFile.txt

SelectFile2=$(ls ~/GGOSS_InputOutput/"$FolderSelected")

echo "$SelectFile2" | yad --title="
PROGRAM - ViralTaxaFinder file selection          GENOME SEQUENCING
Created by Giles Holt" --list --column="Select
files you wish to run" --multiple --width 800 --height 600 --center --align=center --button="OK" --
button="Previous":2 --separator=" > ~/GGOSS/tmp/ViralTaxaFinderSelectedFilesP4.txt

```

```

mode="$?"
  case $mode in
    2)~/GGOSS/Scripts/FileSelection/Viral_InfoFinderfileselect.sh ;;
  esac

FolderSelected2=$(awk 'NR=1 {print}' ~/GGOSS/tmp/ViralTaxaFinderSelectedFilesP4.txt)
echo -n "$FolderSelected2" >> ~/GGOSS/tmp/Path2selectedFile.txt

fi

if [ -s ~/GGOSS/tmp/ViralTaxaFinderSelectedFilesP4.txt ];then
~/GGOSS/Buttons/ButtonGGOSSViralTaxaFinder.sh
fi

```

10.9.2.7.3 DNA and RNA converter, Main menu, settings, file selection

```

#!/bin/bash
SettingsSelected=2
FileSelection=2

ICON=~/GGOSS/Pictures/DNARNA_Conversion.jpg

yad --title="                GENOME SEQUENCING PROGRAM -- DNA RNA converter
Created by Giles Holt" --text="    DNA RNA converter

A Tool for converting
DNA and RNA sequences.
Input sequences must be
in FASTA format but can
be in either 3' or 5'
direction. Note that any
RNA output is given in
the 5'-3' direction
." / --text-align=fill --center --image=$ICON --size=fit --center --button=" Previous ":4 --button="
Run ":3 --button="Select genome/gene files":6 --button="Settings":5 --button="DNA RNA
converter Manual" --buttons-layout=center

mode="$?"
  case $mode in
    3)~/GGOSS/Scripts/DNA_RNA_Converter.sh ;;
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)SettingsSelected=1 ;;
    6)FileSelection=1 ;;
  esac

if [ $SettingsSelected = 1 ];then

  if [ -f ~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt ]
  then

```

```

rm ~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt
fi

mode="$?"
case $mode in
  1)~/GGOSS/Buttons/Button_DNA_RNA_Converter.sh ;;
esac | \

yad --title="  GGOSS - DNA / RNA converter -- Settings                               Created by Giles Holt"
--center --image-on-top --size=fit --center --form \
--field="DNA/RNA Converter:":CB \
--field="Input strand sense:":CB \
--field="Strand conversion:":CB \
--field="Input nucleic acid type:":CB \
--field="Input file type:":CB \
"N/A!DNA to RNA!RNA to DNA" "5-3!3-5" "N/A!Complement!Reverse!Reverse complement!"
"DNA!RNA!" "FASTA format!File containing just nucleotide bases" \
--text-info --show-uri --width=600 --height=500 --center --wrap \
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --editable
--filename=~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt >
~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt

~/GGOSS/Buttons/Button_DNA_RNA_Converter.sh

fi

if [ $FileSelection = 1 ];then

echo | ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

#this means in the case of script exit do 1 of the following
mode="$?"
case $mode in
  3)~/GGOSS/Buttons/Button_Button_DNA_RNA_Converter.sh ;;
esac | \

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="  GENOME SEQUENCING PROGRAM -
Conserved Sequence Finder --file selection          Created by Giles Holt" --list --column="Select
files in which to identify conserved sequences" --multiple --width 800 --height 600 --center --
align=center --on-top --button="Save selection:0" --button="Previous":3 --separator=" --
filename=~/GGOSS/tmp/DNA_RNA_Converter_SelectedFiles.txt >
~/GGOSS/tmp/DNA_RNA_Converter_SelectedFiles.txt &

~/GGOSS/Buttons/Button_DNA_RNA_Converter.sh

fi

```

10.9.2.7.4 File changing tool for editing of file on mass scale

```
#!/bin/bash

SelectFiles=2
Settings=2

yad --title="                GGOSS  --  GENOME SEQUENCING PROGRAM -- Mass File
manipulation      Created by Giles Holt" --center --size=fit \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
  --button="Previous":1 --button="Select files":2 --button="Settings":3 --buttons-layout=center
mode="$?"
  case $mode in
    1)~/GGOSS/GenomicsProgram.sh ;;
    2)SelectFiles=1 ;;
    3)Settings=1 ;;
  esac

if [ $SelectFiles = 1 ];then

echo |ls ~/GGOSS_InputOutput/FastqFiles/ > ~/GGOSS/tmp/SelectFile.txt

#this means in the case of script exit do 1 of the following
mode="$?"
  case $mode in
    3)~/GGOSS/Buttons/Button_MassFileChange.sh ;;
  esac | \

cat ~/GGOSS/tmp/SelectFile.txt | yad --title="      GENOME SEQUENCING PROGRAM -
Conserved Sequence Finder --file selection      Created by Giles Holt" --list --column="Select
files in which to identify conserved sequences" --multiple --width 800 --height 600 --center --
align=center --on-top --button="Save selection:0" --button="Previous":3 --separator=" --
filename=~ /GGOSS/tmp/DNA_RNA_Converter_SelectedFiles.txt >
~/GGOSS/tmp/DNA_RNA_Converter_SelectedFiles.txt &

fi

if [ $Settings = 1 ];then

yad --title="                GGOSS  --  GENOME SEQUENCING PROGRAM -- Settings
Created by Giles Holt" --center --size=fit --center --form \
--field="Delimiter for column identification within file names (e.g. _ or |):" \
--field="Trim names to include columns (e.g. 1,3,7):" \
--field="Merge files":CB \
--field="Merge files if names contain the string:": \
--field="Split files":CB \
--field="Split files into how many separate files":CB \
--field="Convert file to":CB \
'N/A' 'N/A' "All files! Files with same name!" 'N/A' "No!Yes" "N/A!1!2!3!4!5!6!7!8!9!10"
"Fasta!Fastq!png!jpeg!pdf" \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
```

```
--button="gtk-save:0" --button="Default reset":2 --button="gtk-close:1" --buttons-layout=center
--editable --filename=~/.GGOSS/tmp/MassFilesmanipulation_Settings.txt >
~/GGOSS/tmp/MassFilesmanipulation_Settings.txt
```

```
mode="$?"
case $mode in
  0)~/GGOSS/Buttons/Button_MassFileChange.sh ;;
  1)~/GGOSS/Buttons/Button_MassFileChange.sh ;;
  2)Reset=1 ;;
esac

fi
```

10.9.2.7.5 Taxonomic abundance table filtering

```
#!/bin/bash
```

```
ICON=~/.GGOSS/Pictures/bacteria_taxonomy_900.jpg
```

```
ls ~/GGOSS_InputOutput/Mothur/OTUtable > ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt
```

```
Choice1=$(sed -n 1p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice2=$(sed -n 2p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice3=$(sed -n 3p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice4=$(sed -n 4p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice5=$(sed -n 5p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice6=$(sed -n 6p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice7=$(sed -n 7p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice8=$(sed -n 8p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice9=$(sed -n 9p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice10=$(sed -n 10p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice11=$(sed -n 11p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice12=$(sed -n 12p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice13=$(sed -n 13p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice14=$(sed -n 14p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice15=$(sed -n 15p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice16=$(sed -n 16p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice17=$(sed -n 17p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
Choice18=$(sed -n 18p ~/GGOSS/tmp/All_OTUtableInputDataFiles.txt)
```

```
yad --title="
GENOME SEQUENCING PROGRAM -- OTU
table Trim Settings Created by Giles Holt" --center --image=$ICON --image-on-top --size=fit --
center --form \
--field="Input file":CB \
--field="Trim OTU names":CB \
--field="Trim OTUs by Hit No." \
--field="Trim OTUs by %" \
"${Choice1}!${Choice2}!${Choice3}!${Choice4}!${Choice5}!${Choice6}!${Choice7}!${Choice8}
!${Choice18}!${Choice17}!${Choice16}!${Choice15}!${Choice14}!${Choice13}!${Choice12}!$
{Choice11}!${Choice10}!${Choice9}!" "Yes!No!" '- '-'\
--text-info --show-uri --width=600 --height=500 --center --wrap \
```

```
--button="gtk-save:0" --button="Previous":1 --button="gtk-close:1" --buttons-layout=center --
editable --filename=~/.GGOSS/tmp/OTUtableSettingsChange.txt >
~/GGOSS/tmp/OTUtableSettingsChange.txt
```

```
mode="$?"
case $mode in
  1)~/GGOSS/Buttons/ButtonPostMothurAnalysis.sh ;;
esac
```

```
~/GGOSS/Buttons/ButtonRunTableTrim.sh
```

```
#!/bin/bash
```

```
#adapt the file to it reads like a proper settings menu
#change settings tmp text into individual lines per setting
sed 's/\\n/g' ~/GGOSS/tmp/OTUtableSettingsChange.txt >
~/GGOSS/tmp/OTUtableSettingsChangeList.txt
```

```
#Add specific text to start of each line
sed -i '1 s/^Input file selected: /' ~/GGOSS/tmp/OTUtableSettingsChangeList.txt
sed -i '2 s/^Trim OTU names:/' ~/GGOSS/tmp/OTUtableSettingsChangeList.txt
sed -i '4 s/^Trim OTUs by %:/' ~/GGOSS/tmp/OTUtableSettingsChangeList.txt
```

```
Settings=$(cat ~/GGOSS/tmp/OTUtableSettingsChangeList.txt)
```

```
yad --title="GGOSS - Table Trim Created by Giles Holt" --list --multiple --width 400 --height 100 -
-center --align=center --button="Run":0 --button="Cancel":1 --separator="" --text="$Settings"
```

```
mode="$?"
case $mode in
  0)~/GGOSS/Scripts/MothurTablePercentTrim.sh ;;
  1)~/GGOSS/Buttons/ButtonPostMothurAnalysis.sh ;;
esac
```

10.9.2.7.6 Automated taxonomic abundance plotting

```
#!/bin/bash
```

```
ICON=~/.GGOSS/Pictures/bacteria_taxonomy_900.jpg
```

```
yad --title="
Created by Giles Holt" --width=800 --height=500 --center --image=$ICON --image-on-top --size=fit -
-center --button="Rarefaction":0 --button="MDS":1 --button="Volcano Plot":2 --button="OTU
Histogram":3 --button="PCA":5 --button="OTU table clean-up":6 --button="Previous":4 --buttons-
layout=center
```

```
mode="$?"
```



```

Cutadapt="N/A"
FastQC="N/A"
Khmer="$HOME/khmerEnv/bin/"
Sickle="N/A"
MUMmer="NotSorted"
PRICE="$HOME/PriceSource140408/"
SPAdes="$HOME/SPAdes-3.6.1-Linux/bin/"
Velvet="NotSorted"
IDBA="NotSorted"
PROKKA="$HOME/prokka-1.11/bin/"
Artemis="N/A"
BLAST="$HOME/ncbi-blast-2.3.0+/"
QUAST="N/A"
Ragout="NotSorted"
BWA="N/A"
SAMtools="NotSorted"
Gepard="NotSorted"
Mauve="NotSorted"
Mothur="NotSorted"
PIPITS="N/A"
MetaPhlAn="NotSorted"

```

fi

```

yad --title="
                                GGOSS -- GENOME SEQUENCING PROGRAM --
Tool path settings      Created by Giles Holt" --center --size=fit --center --form \
--field="Cutadapt:" \
--field="FastQC:" \
--field="Khmer:" \
--field="Sickle:" \
--field="MUMmer:" \
--field="PRICE:" \
--field="SPAdes:" \
--field="Velvet:" \
--field="IDBA:" \
--field="PROKKA:" \
--field="Artemis:" \
--field="BLAST:" \
--field="QUAST:" \
--field="Ragout:" \
--field="BWA:" \
--field="SAMtools:" \
--field="Gepard:" \
--field="Mauve:" \
--field="Mothur:" \
--field="PIPITS:" \
--field="MetaPhlAn:" \
"$Cutadapt" "$FastQC" "$Khmer" "$Sickle" "$MUMmer" "$PRICE" "$SPAdes" "$Velvet" "$IDBA"
"$PROKKA" "$Artemis" "$BLAST" "$QUAST" "$Ragout" "$BWA" "$SAMtools" "$Gepard"
"$Mauve" "$Mothur" "$PIPITS" "$MetaPhlAn" \
--text-info --show-uri --width=600 --height=500 --align=right --wrap \
--button="gtk-save:0" --button="Default reset":2 --button="gtk-close:1" --buttons-layout=center
--editable --filename=~/.GGOSS/tmp/PathToTools_SettingsChange.txt >
~/GGOSS/tmp/PathToTools_SettingsChange.txt

```

```

mode="$?"
  case $mode in
    0)echo "1" > ~/GGOSS/tmp/OriginalDefaultHasBeenAdapted.txt &
~/GGOSS/GenomicsProgram.sh ;;
    1)~/GGOSS/GenomicsProgram.sh ;;
    2)Reset=1 ;;
  esac

  if [ $Reset = 1 ];then
  rm ~/GGOSS/tmp/OriginalDefaultHasBeenAdapted.txt
~/GGOSS/Buttons/ButtonToolPaths.sh
  fi

```

10.9.3 Scripts

The following sub-sections contain all the scripts written in order to carry out all the functions/tools selected in the GGOSS GUI

10.9.3.1 File clean-up

10.9.3.1.1 All GGOSS Cutadapt scripts

10.9.3.1.1.1 Adapter change

```
#!/bin/sh
```

```
#this removes the adapter currently in place but doesn't recognise the variable so puts nothing in its place
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScript3.sh
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScript5.sh
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScript_53.sh
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScriptAnchored3.sh
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScriptAnchored5.sh
```

```
sed -i -e "s/AACCGGTT\+/${(head ~/GGOSS/tmp/AdapterInput.txt)/g"
~/GGOSS/Scripts/CutAdapterScript_LinkedAdapter.sh
```

10.9.3.1.1.2 Custom adapter list run

```
#!/bin/sh
rm ~/GGOSS/tmp/CustomAdapterList.txt
#Custom Adapter list trim

#this does'nt work if they use tilde to indicate home diretory
AdapterList=$(awk -F "|" '{print $4}' ~/GGOSS/tmp/Cutadapt_SettingsChange.txt)
echo "AdapterList: $AdapterList"
ReverseCompliment=$(awk -F "|" '{print $5}' ~/GGOSS/tmp/Cutadapt_SettingsChange.txt)
echo "ReverseCompliment: $ReverseCompliment"
NumberOfLines=$(cat $AdapterList | grep -v -c "ThisIsMyAntiMatch")

##### create Reverse compliment of the adapter list #####

if [ "$ReverseCompliment" = "Yes" ];then
echo "Generating Reverse Compliment Adapters..."
#loop to number of lines
line=1
  for i in $(seq $NumberOfLines);do
    #grab first line and remove any white space
##Cat the variable/file is the problem here and above
    AdapterToRC=$(cat $AdapterList | awk -v x=$line 'NR==x {print}' | tr -d '[:space:]')
    #number of bases in given line
    NumberOfCharactersInLine=$(echo "$AdapterToRC" | awk '{ print length }')

    #loop to number of characters of chosen line
    CharacterNumber=1
    for i in $(seq $NumberOfCharactersInLine);do
      #last character (then 2nd to last then 3rd to last etc)
      Character=$(echo "$AdapterToRC" | rev | awk -v x=$CharacterNumber '{print substr ($0, x,
1)}')
      #Change the character
      if [ "$Character" = "A" ];then
        NewCharacter=T
      fi
      if [ "$Character" = "T" ];then
        NewCharacter=A
      fi
      if [ "$Character" = "G" ];then
        NewCharacter=C
      fi
      if [ "$Character" = "C" ];then
        NewCharacter=G
      fi

      if [ $CharacterNumber = $NumberOfCharactersInLine ];then
        echo "$NewCharacter" >> ~/GGOSS/tmp/CustomAdapterList.txt
      else
        echo -n "$NewCharacter" >> ~/GGOSS/tmp/CustomAdapterList.txt
      fi

      CharacterNumber=$(( $CharacterNumber + 1 ))
    done
  done
```

```

done
line=$(( $line + 1 ))
done

fi

#####

# -- combine the 2 lists RevComp and norm, whilst ensuring there are no spaces left between bases

cat "$AdapterList" | tr -d " " >> ~/GGOSS/tmp/CustomAdapterList.txt
TotalNumberOfAdapters=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/CustomAdapterList.txt)

##Run cutadapt on the adapters in the list

{
echo "          GGOSS Genomics - Created by Giles Holt"
"
echo "          CutAdapter run from GGOSS Genomics program"
"
echo | (date +"          CutAdapter Start Date: %d-%m-%y Time: %T

")

echo 0
echo "#Running Cutadapter...          0% complete"

Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt)

#checks more thoroughly
#variable of all files sorted alphabetically
AlphabetSortedFiles=$(sort ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt)

line=1
for i in $(seq 1 "$NumberOfFiles");do

FileToTrim=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')
filename=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}' | awk -F '_' '{NF-=4;
OFS="_"; print}')
FileToTrimRead1=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')

if [ -f ~/GGOSS_InputOutput/FastqFiles/$FileToTrimRead1 ]

```

```

then
  #Remove the run file if already exists
  if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ];then
  rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  fi

  #this loop runs through all the adapter possibilities for the file
  AdapChange=1
  for i in $(seq $TotalNumberOfAdapters);do
  cp ~/GGOSS/Scripts/CutAdapter_NexteraXTtemplate.sh
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  echo "-----

----- Beggining of loop for adapter: $AdapChange

-----"

  if [ "$AdapChange" != "$TotalNumberOfAdapters" ];then
  echo "Doesnt = $TotalNumberOfAdapters"
  sed -e "s/FileOutputNameRead1\+/tmp${AdapChange}Adapcut_{FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
  mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  else
  sed -e
"s|FastqFiles/FileOutputNameRead1|AdapterTrimmedFiles/Adapcut_{FileToTrimRead1}|g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
  mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  fi

  #Change input file name read 1, where it re-inputs the previous adapter removal file
  echo "AdapChange: $AdapChange"
  if [ "$AdapChange" != "1" ];then
  echo "AdapChange is > 1"
  AdapChangeForInput=$(( $AdapChange - 1 ))
  sed -e
"s/FileInputNameRead1\+/tmp${AdapChangeForInput}Adapcut_{FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
  mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  else
  sed -e "s/FileInputNameRead1\+/${FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
  mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
  fi

  #change adapter
  Adapter=$(awk -v x=$AdapChange 'NR==x {print}' ~/GGOSS/tmp/CustomAdapterList.txt | tr -
d [:space:])

  sed -e "s/TheAdapter\+/${Adapter}/g" ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh >
~/GGOSS/tmp/tmp.txt
  mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh

  Time2min=$(date +%M)
  Time2hourtmp=$(date +%H")

```

```

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$NumberOfFiles - $line" | bc )
EstimatedtimetoFinishMins=$( echo "scale=2; ($TimeTaken / $line) * $NumberFilesLeftToDo" |
bc )
AverageTimeTakenPerSample=$( echo "scale=2; ($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "scale=2; $EstimatedtimetoFinishMins / 60" | bc )

PercentWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfFiles" | bc )
PercentCompleteBySample=$( echo "scale=2; $PercentWorthOfEachFile * $line" | bc )
PercentWorthOfAdapterRun=$( echo "scale=2; ($PercentWorthOfEachFile /
$TotalNumberOfAdapters) * $AdapChange" | bc )
PercentComplete=$( echo "scale=2; (($PercentWorthOfEachFile * $line) -
$PercentWorthOfEachFile) + $PercentWorthOfAdapterRun" | bc )

```

```

Info=$(echo "Start time:${Time1hourtmp}:${Time1min}
Running sample: $FileToTrim ($line of $NumberOfFiles)
Time elapsed: ${TimeTaken} minutes
Estimated time remaining: $EstimatedtimetoFinishMins minutes (${EstimatedtimetoFinishHours}
hours)
Average time taken per sample: $AverageTimeTakenPerSample

```

Please check the Cutadapt Logfile to confirm the success of the run on your samples. You can also find other information there, including the version of Cutadapt that was used")

```

#Run edited script
#have to chmod it as renaming it removed permissions
echo "#Running Cutadapter... file: $filename AdapterType: $AdapChange
${PercentComplete}% complete"
echo ${PercentComplete}
chmod 755 ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh

#remove no longer needed created files
if [ "$AdapChange" > "1" ];then
#change below back to -1 for real run function, just doing -2 so can check the last run through
manually
AdapChangeRem=$(( $AdapChange - 1 ))
if [ -f
~/GGOSS_InputOutput/FastqFiles/tmp${AdapChangeRem}Adapcut_${FileToTrimRead1} ];then
rm -f
~/GGOSS_InputOutput/FastqFiles/tmp${AdapChangeRem}Adapcut_${FileToTrimRead1}
fi

```

```

fi

echo "-----

----- End of loop for adapter: $AdapChange

-----"
AdapChange=$(( $AdapChange + 1 ))
echo "AdapChange at end of loop: $AdapChange"
done

else

echo "Failed to find the selected file: $FileToTrim"

fi

#delete script in prep to remake for next file

if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ];then
rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

echo "End Of File: $line"

line=$(( $line + 1 ))

done
pskill yad

} | tee ~/GGOSS/LogFiles/CutAdapter_LOGFILE.txt | yad --progress --auto-kill --center --width=700
--image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- Cutadapt
GGOSS created by Giles Holt" --text="Running Cutadapt
$Info"

if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ];then
rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

#go back to main menu
yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Cutadapt" --
button="Open file location":5 --button="Return to menu":4 --text="

Cutadapt Complete

"

mode="$?"
case $mode in
4)~/GGOSS/GenomicsProgram.sh ;;
5)nautilus ~/GGOSS_InputOutput/AdapterTrimmedFiles & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.1.1.3 Cutadapt paired-end Nextera XT

```
#!/bin/sh
#paired end nextera xt

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                CutAdapter run from GGOSS Genomics program"
"
echo |(date +"                CutAdapter Start Date: %d-%m-%y Time: %T

")

echo 0
echo "#Running Cutadapter...                0% complete"

Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt)

#checks more thoroughly
#variable of all files sorted alphabetically
AlphabetSortedFiles=$(sort ~/GGOSS/tmp/Cutadapt_SelectedFiles.txt)

line=1
for i in $(seq 1 "$NumberOfFiles");do

FileToTrim=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')
filename=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}' | awk -F '_' '{NF-=4;
OFS="_"; print}')
FileToTrimRead1=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')

if [ -f ~/GGOSS_InputOutput/FastqFiles/$FileToTrimRead1 ]
then
#Remove the run file if already exists
if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ]
then
rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

#this loop runs through all the adapter possibilities for the file
```

```

AdapChange=1
for i in $(seq 50);do
cp ~/GGOSS/Scripts/CutAdapter_NexteraXTtemplate.sh
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
echo "-----"

----- Beggining of loop for adapter: $AdapChange
-----"

if [ "$AdapChange" != "50" ];then
echo "Doesnt = 50"
sed -e "s/FileOutputNameRead1\+/tmp${AdapChange}Adapcut_{FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
else
sed -e
"s|FastqFiles/FileOutputNameRead1|AdapterTrimmedFiles/Adapcut_{FileToTrimRead1}|g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

#Change input file name read 1, where it re-inputs the previous adapter removal file
echo "AdapChange: $AdapChange"
if [ "$AdapChange" != "1" ];then
echo "AdapChange is > 1"
AdapChangeForInput=$(( $AdapChange - 1 ))
sed -e
"s/FileInputNameRead1\+/tmp${AdapChangeForInput}Adapcut_{FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
else
sed -e "s/FileInputNameRead1\+/${FileToTrimRead1}/g"
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh > ~/GGOSS/tmp/tmp.txt
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

#change adapter
Adapter=$(awk -v x=$AdapChange 'NR==x {print}' ~/GGOSS/NexteraAdapterList.txt | tr -d
[:space:])

sed -e "s/TheAdapter\+/${Adapter}/g" ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh >
~/GGOSS/tmp/tmp.txt
mv ~/GGOSS/tmp/tmp.txt ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh

Time2min=$(date +%M")
Time2hourtmp=$(date +%H")

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

```

```

NumberFilesLeftToDo=$( echo "$NumberOfFiles - $line" | bc )
EstimatedtimetoFinishMins=$( echo "scale=2; ($TimeTaken / $line) * $NumberOfFilesLeftToDo" |
bc )
AverageTimeTakenPerSample=$( echo "scale=2; ($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "scale=2; $EstimatedtimetoFinishMins / 60" | bc )

PercentWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfFiles" | bc )
PercentCompleteBySample=$( echo "scale=2; $PercentWorthOfEachFile * $line" | bc )
PercentWorthOfAdapterRun=$( echo "scale=2; ($PercentWorthOfEachFile / 50) *
$AdapChange" | bc )
PercentComplete=$( echo "scale=2; (($PercentWorthOfEachFile * $line) -
$PercentWorthOfEachFile) + $PercentWorthOfAdapterRun" | bc )

Info=$(echo "Start time:${Time1hourtmp}:${Time1min}
Running sample: $FileToTrim ($line of $NumberOfFiles)
Time elapsed: ${TimeTaken} minutes
Estimated time remaining: $EstimatedtimetoFinishMins minutes (${EstimatedtimetoFinishHours}
hours)
Average time taken per sample: $AverageTimeTakenPerSample

```

Please check the Cutadapt Logfile to confirm the success of the run on your samples. You can also find other information there, including the version of Cutadapt that was used")

```

#Run edited script
#have to chmod it as renaming it removed permissions
echo "#Running Cutadapter... file: $filename AdapterType: $AdapChange
${PercentComplete}% complete"
echo ${PercentComplete}
chmod 755 ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh

#remove no longer needed created files
if [ "$AdapChange" > "1" ];then
#change below back to -1 for real run function, just doing -2 so can check the last run through
manually
AdapChangeRem=$(( $AdapChange - 1 ))
if [ -f
~/GGOSS_InputOutput/FastqFiles/tmp${AdapChangeRem}Adapcut_${FileToTrimRead1} ];then
rm -f
~/GGOSS_InputOutput/FastqFiles/tmp${AdapChangeRem}Adapcut_${FileToTrimRead1}
fi
fi

echo "-----

----- End of loop for adapter: $AdapChange

```

```

    -----"
    AdapChange=$(( $AdapChange + 1 ))
    echo "AdapChange at end of loop: $AdapChange"
done

else

echo "Failed to find the selected file: $FileToTrim"

fi

#delete script in prep to remake for next file

if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ]
then
rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

echo "End Of File: $line"

line=$(( $line + 1 ))

done

pkill yad

} | tee ~/GGOSS/LogFiles/CutAdapter_LOGFILE.txt | yad --progress --auto-kill --center --width=700
--image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- Cutadapt
GGOSS created by Giles Holt" --text="Running Cutadapt
$Info"

if [ -f ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh ]
then
rm ~/GGOSS/Scripts/CutAdapter_NexteraXTrun.sh
fi

#go back to main menu
yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Cutadapt" --
button="Open file location":5 --button="Return to menu":4 --text="

Cutadapt Complete

"

mode="$?"
case $mode in
4)~/GGOSS/GenomicsProgram.sh ;;
5)nautilus ~/GGOSS_InputOutput/AdapterTrimmedFiles & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.1.1.4 Cutadapt template

```
#!/bin/sh
cutadapt -a TheAdapter -o ~/GGOSS_InputOutput/FastqFiles/FileOutputNameRead1
~/GGOSS_InputOutput/FastqFiles/FileInputNameRead1
```

10.9.3.1.1.5 Cutadapt paired end template

```
#!/bin/sh

#change output path

cutadapt -a ForwardAdapter -A ReverseAdapter -o
~/GGOSS_InputOutput/AdapterTrimmedFiles/FileOutputNameRead1 -p
~/GGOSS_InputOutput/AdapterTrimmedFiles/FileOutputNameRead2
~/GGOSS_InputOutput/FastqFiles/FileInputNameRead1
~/GGOSS_InputOutput/FastqFiles/FileInputNameRead2
```

10.9.3.1.2 Khmer

10.9.3.1.2.1 Khmer template

```
#!/bin/sh

PathToKhmer=$(awk -F "|" '{print $3}' ~/GGOSS/tmp/PathToTools_SettingsChange.txt)
if [ "$PathToKhmer" = "N/A" ];then
PathToKhmer="$HOME/"
fi
LoadIntoCount=$(awk -F "|" '{print $3}' ~/GGOSS/tmp/KhmerSettingsChange.txt)
if [ "$LoadIntoCount" = "Yes" ]; then
"${PathToKhmer}"load-into-counting.py Processor_Number RAM_Limit
~/GGOSS_InputOutput/Khmer/LIC_FileName.kh FilePath/File_NameFull
fi

AbundanceDistrib=$(awk -F "|" '{print $4}' ~/GGOSS/tmp/KhmerSettingsChange.txt)
if [ "$AbundanceDistrib" = "Yes" ]; then
"${PathToKhmer}"abundance-dist.py ~/GGOSS_InputOutput/FastqFiles/FileKhName
FilePath/File_NameFull ~/GGOSS_InputOutput/Khmer/AbDist_FileName.hist
fi
```

10.9.3.1.2.2 GGOSS script for Khmer

```
#!/bin/sh

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                Khmer run from GGOSS Genomics program"
"
}
```

```

echo | (date +"                Khmer Start Date: %d-%m-%y Time: %T

")

yad --title="GENOME SEQUENCING PROGRAM -- Khmer                Created by Giles Holt" --
timeout=10 --no-buttons --no-escape --width=300 --length=300 --text="Please check the Khmer run
Logfile to ascertain and

confirm the success of khmer run on your samples. You can

also find other information there, including the version of Khmer that was used

" &

#sorts them so they are in order, as R1 and R2 names are identical except for the 1 or 2, they will
be sure to be grouped together, allowing to select them by the line and the line +1
sort ~/GGOSS/tmp/KhmerSelectedFiles.txt

Settings=$(cat ~/GGOSS/tmp/KhmerSettingsChangeList.txt)
AbundanceDistrib=$(awk -F "|" '{print $4}' ~/GGOSS/tmp/KhmerSettingsChange.txt)

NumberOfSamplesToRun=$(grep                -c                -v                "ThisIsMyAntiMatch"
~/GGOSS/tmp/KhmerSelectedFilesP4.txt)

Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

line=1
for i in $(seq 1 ${NumberOfSamplesToRun});do

Time2min=$(date +"%M")
Time2hourtmp=$(date +"%H")

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$NumberOfSamplesToRun - $line" | bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberOfFilesLeftToDo" | bc )

AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

```

```

filenameR1=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/KhmerSelectedFilesP4.txt)

#things left to sort - below variable and the button links to this script once name changed

#this will only set the filename correctly if there is an underscore after the filename
filename="$filenameR1"

khFilename=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/KhmerkHSelectedFilesP4.txt)

if [ -f ~/GGOSS_InputOutput/Khmer/Khmer_$filename ]
then
    echo "Assembly output file for $filename already exists, if you wish to re-run this samples
please delete or move its pre-existing Khmer output. Running the next sample
"
    echo | pkill yad
    echo | notify-send "Assembly output file for $filename already exists, if you wish to re-run
this samples please delete or move its pre-existing Khmer output. Running the next sample
"
fi

FilePath=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt | awk -F '/' '{ print substr($0,
index($0,$2)) }')
echo "File Path and Name: ${FilePath}/${filenameR1}"
if [ -f "$HOME/${FilePath}/${filenameR1}" ];then
    notify-send "Running Khmer for file: $filename"

    echo "Running Khmer for file: $filename
"

#change the Khmer assembler script for the filename at hand
#change file 1
sed -i -e "s/File_NameFull\+/${filenameR1}/g" ~/GGOSS/Scripts/Khmer.sh

#change output file name
sed -i -e "s/FileName\+/${filename}/g" ~/GGOSS/Scripts/Khmer.sh

#for abundance dist

if [ "$AbundanceDistrib" = "Yes" ]; then
sed -i -e "s/FileKhName\+/${khFilename}/g" ~/GGOSS/Scripts/Khmer.sh
fi

yad --title="GGOSS -- Khmer" --width=700 --center --sticky --on-top --no-buttons --no-escape --
text-align=center --text="    Running Khmer

Total number of samples for Khmer to run: $NumberOfSamplesToRun

Currently running Khmer on file: ${filename}. Started at:${Time2hourtmp}:${Time2min}

Time taken thus far: ${TimeTaken} minutes
Average time taken per sample:$AverageTimeTakenPerSample

```

Estimated time left until completion:\$EstimatedtimetoFinishMins minutes
(\${EstimatedtimetoFinishHours} hours)

Settings:
\$Settings
" &

\$HOME/GGOSS/Scripts/Khmer.sh

pkill yad

#change back ready for the next file type

#change file 1

sed -i -e "s/\$filenameR1\+/File_NameFull/g" ~/GGOSS/Scripts/Khmer.sh

#change output file

sed -i -e "s/\$filename\+/FileName/g" ~/GGOSS/Scripts/Khmer.sh

else

notify-send "Can't find the \$filename file selected for Khmer run, either a significant error has occurred or the file name or location has been tampered with since starting the run.

Trying next sample"

echo "Can't find the \$filename file selected for Khmer run, either a significant error has occurred or the file name or location has been tampered with since starting the run.

Trying next sample

"

fi

line=\$((\$line + 1))

done

} | tee ~/GGOSS/LogFiles/Khmer_LOGFILE.txt

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Khmer" --button="Open file location":5 --button="Return to menu":4 --text="

Khmer Complete

"

mode="\$?"

case \$mode in

4)~/GGOSS/GenomicsProgram.sh ;;

5)nautilus ~/GGOSS_InputOutput/Khmer & ~/GGOSS/GenomicsProgram.sh ;;

esac

10.9.3.1.3 GGOSS script for FastQC

```
#!/bin/bash

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                Fastqc run from GGOSS Genomics program"
"
echo |(date +"                Fastqc Start Date: %d-%m-%y Time: %T

")

cp ~/GGOSS/Scripts/FastqcTemplate.sh ~/GGOSS/Scripts/FastqcRun.sh

echo -n "fastqc " >> ~/GGOSS/Scripts/FastqcRun.sh

NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Fastqc_SelectedFiles.txt)

line=1
for i in $(seq 1 $NumberOfFiles)
do

File=$(awk -v y=$line 'NR==y {print}' ~/GGOSS/tmp/Fastqc_SelectedFiles.txt)

echo -n "~/GGOSS_InputOutput/FastqFiles/$File " >> ~/GGOSS/Scripts/FastqcRun.sh

line=$(( $line + 1 ))

done

echo -n "-o ~/GGOSS_InputOutput/Fastqc/" >> ~/GGOSS/Scripts/FastqcRun.sh

StartTime=$(date +"%H:%M")

yad --title="GGOSS -- Fastqc" --width=400 --center --sticky --on-top --no-buttons --no-escape --text-align=center --text="    Running Fastqc

Total number of files for Fastqc: $NumberOfFiles
Fastqc was started at: $StartTime
" &

~/GGOSS/Scripts/FastqcRun.sh

#delete script

if [ -f ~/GGOSS/Scripts/FastqcRun.sh ]
then
rm ~/GGOSS/Scripts/FastqcRun.sh
```

```

fi

#Closes window
pkill yad

} | tee ~/GGOSS/LogFiles/Fastqc_LOGFILE.txt

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- FastQC" --
button="Open file location":5 --button="Return to menu":4 --text="

FastQC Complete

"

mode="$?"
case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/Fastqc & ~/GGOSS/GenomicsProgram.sh ;;
    Esac

```

10.9.3.1.4 Sickle

10.9.3.1.4.1 Sickle template script

```

#!/bin/sh

RunType=$(awk -F "|" '{print $2}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
echo "Run Type: $RunType"
if [ "$RunType" = "Paired End" ]
then

Interleaved=$(awk -F "|" '{print $5}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)

    if [ "$Interleaved" = "Interleaved" ]
    then
#Paired end Interleaved reads
sickle pe -c ~/GGOSS_InputOutput/FastqFiles/FileInputName1 -t SeqMachineCompany -M
~/GGOSS_InputOutput/Sickle/SickleTrim_FileOutputName1 -s SinglesFile

    else
#Paired end Seperate reads
sickle pe -f ~/GGOSS_InputOutput/FastqFiles/FileInputName1 -r
~/GGOSS_InputOutput/FastqFiles/FileInputName2 -t SeqMachineCompany -o
~/GGOSS_InputOutput/Sickle/SickleTrim_FileOutputName1 -p
~/GGOSS_InputOutput/Sickle/SickleTrim_FileOutputName2 -s SinglesFile -q QualityThreshold -l
LengthThreshold
    fi
fi

```

10.9.3.1.4.2 GGOSS script for Sickle

```

#!/bin/bash

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                Sickle run from GGOSS Genomics program"
"
echo |(date +"                Sickle Start Date: %d-%m-%y Time: %T
")

echo 1
echo "#Sickle applying settings          1% Complete"
Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

RunType=$(awk -F "|" '{print $2}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)

if [ "$RunType" = "Paired End" ]
then
echo 2
echo "#Sickle applying settings          2% Complete"
NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Sickle_SelectedFiles.txt)

NumberOfPairedFiles=$(echo "$NumberOfFiles / 2" | bc)

NoPairPrep=$(echo "$NumberOfPairedFiles" | awk -F '.' '{print $2}')

InterleavedOrSeparate=$(awk -F "|" '{print $5}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)

if [ "$InterleavedOrSeparate" = "Interleaved" ];then
echo 3
echo "#Sickle applying settings          3% Complete"
NoPair=1
else
echo 3
echo "#Sickle applying settings          3% Complete"
NoPair=$(( $NoPairPrep + 1 ))
fi

#reset file error number (just in case variables still exist from previous attempt)
FilePairError=2

#Checks for even number of files and that every file has its counterpart.

if [ "$NoPair" = "1" ]
then
echo 4
echo "#Sickle applying settings          4% Complete"

```

```

echo "Expected number of files for paired end quality trimming found"

#checks more thoroughly
#variable of all files sorted alphabetically
AlphabetSortedFiles=$(sort ~/GGOSS/tmp/Sickle_SelectedFiles.txt)

FileLine=1
for i in $(seq 1 $NumberOfFiles)
do
#takes the name of the file
FileName=$(echo "$AlphabetSortedFiles" | awk -v y=$FileLine 'NR==y {print $1}' | awk -F
'_' '{NF-=4; OFS=" "; print}')

#counts how often that name comes up into the file list
FileNameFrequency=$(grep -c $FileName ~/GGOSS/tmp/Sickle_SelectedFiles.txt)

if [ "$FileNameFrequency" = 2 ]
then
echo "File pairing complete for file $FileName"
else
echo "
Error in file pairing! File: ${FileName}, has $FileNameFrequency match/es, there should only
be 2.
Please resolve this file issue and retry. Note that it may be your pairing file sample name is not
named the same for some reason, or that the sample name contains an '_', if this is the case, adjusting
this will resolve the problem
"
FilePairError=1

fi

FileLine=$(( $FileLine + 1 ))
done

echo 5
echo "#Sickle applying settings      5% Complete"

if [ "$FilePairError" = "1" ]
then
echo "Paired end quality trimming not run due to file pairing error"
echo 6
echo "#Sickle applying settings      6% Complete"
else
echo 6
echo "#Sickle applying settings      6% Complete"
echo "File pairing complete - Commencing paired end quality trimming"

#orders the files by name, then takes the beginning name (col 1 with . or _ as seperator),
flicksthrough list finding match, counts match, which should equal 2, if not it issues a warning, letting
them know there could be a problem with the run, and which files are the issues.

line=1

for i in $(seq 1 "$NumberOfPairedFiles")

```

```

do

FileToTrim=$(echo "$AlphabetSortedFiles" | awk -v x=$line ' NR==x {print}')
echo "FileToTrim: $FileToTrim"
FileToTrimRead1=$(echo "$AlphabetSortedFiles" | awk -v x=$line ' NR==x {print}')
echo "FileToTrimRead1: $FileToTrimRead1"
File2Line=$(( $line + 1 ))
echo "File2Line: $File2Line"

FileToTrimRead2=$(echo "$AlphabetSortedFiles" | awk -v x=$File2Line ' NR==x {print}')
echo "FileToTrimRead2: $FileToTrimRead2"

#is the file selected still there
if [ -f ~/GGOSS_InputOutput/FastqFiles/$FileToTrimRead1 ] && [ -f
~/GGOSS_InputOutput/FastqFiles/$FileToTrimRead2 ]
then

#Remove the run file if already exists
# if [ -f ~/GGOSS/Scripts/Sickle_Run.sh ];then
#rm ~/GGOSS/Scripts/Sickle_Run.sh
#fi

cp ~/GGOSS/Scripts/Sickle_Template.sh ~/GGOSS/Scripts/Sickle_Run.sh

#edit script: ~/GGOSS/Scripts/Sickle_Run.sh: ready for Running

#change output file name read 1
sed -i -e "s/FileOutputName1\+/${FileToTrimRead1}/g" ~/GGOSS/Scripts/Sickle_Run.sh

#change output file name read 2
sed -i -e "s/FileOutputName2\+/${FileToTrimRead2}/g" ~/GGOSS/Scripts/Sickle_Run.sh

#Change input file name read 1
sed -i -e "s/FileInputName1\+/${FileToTrimRead1}/g" ~/GGOSS/Scripts/Sickle_Run.sh

#Change input file name read 2
sed -i -e "s/FileInputName2\+/${FileToTrimRead2}/g" ~/GGOSS/Scripts/Sickle_Run.sh

#Change SeqMachineCompany
SickleCompanySelected=$(awk -F "|" '{print $1}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)

sed -i -e "s/SeqMachineCompany\+/${SickleCompanySelected}/g" ~/GGOSS/Scripts/Sickle_Run.sh

#length threshold change

LengthThreshold=$(awk -F "|" '{print $3}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)

if [ "$LengthThreshold" = "-" ];then
sed -i -e "s/-1 LengthThreshold\+//g" ~/GGOSS/Scripts/Sickle_Run.sh
else
sed -i -e "s/LengthThreshold\+/${LengthThreshold}/g" ~/GGOSS/Scripts/Sickle_Run.sh
fi

```

#quality threshold change

```
QualityThreshold=$(awk -F "|" '{print $4}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
if [ "$QualityThreshold" = "-" ];then
```

```
sed -i -e "s/-q QualityThreshold \+//g" ~/GGOSS/Scripts/Sickle_Run.sh
```

```
else
```

```
sed -i -e "s/QualityThreshold\+/{QualityThreshold}/g" ~/GGOSS/Scripts/Sickle_Run.sh
```

```
fi
```

#Seperate or interleaved

```
InterleavedOrSeparate=$(awk -F "|" '{print $5}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
InterleavedOutputType=$(awk -F "|" '{print $6}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
if [ "$InterleavedOrSeparate" = "Separate" ] || [ "$InterleavedOutputType" = "No" ];then
```

```
sed -i -e "s/SinglesFile\+/Singles_{FileName}/g" ~/GGOSS/Scripts/Sickle_Run.sh
```

```
else
```

```
sed -i -e "s/-s SinglesFile \+//g" ~/GGOSS/Scripts/Sickle_Run.sh
```

```
fi
```

```
InterleavedOutputType=$(awk -F "|" '{print $6}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
if [ "$InterleavedOutputType" = "No" ];then
```

```
sed -i -e "s/-M \+/-m /g" ~/GGOSS/Scripts/Sickle_Run.sh
```

```
fi
```

```
Interleaved=$(awk -F "|" '{print $5}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
RunType=$(awk -F "|" '{print $2}' ~/GGOSS/tmp/Sickle_SettingsChange.txt)
```

```
Time2min=$(date +%M)
```

```
Time2hourtmp=$(date +%H)
```

```
Time2hour=$( echo "$Time2hourtmp * 60" | bc )
```

```
EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )
```

```
TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )
```

```
NumberFilesLeftToDo=$( echo "$NumberOfFiles - $line" | bc )
```

```
EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberFilesLeftToDo" | bc )
```

```
AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )
```

```
EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )
```

```
PercentWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfPairedFiles" | bc )
```

```
echo "PercentWorthOfEachFile: $PercentWorthOfEachFile"
```

```
PercentComplete=$( echo "scale=2; $PercentWorthOfEachFile * $line" | bc )
```

```
echo "PercentComplete: $PercentComplete"
```

```

#Run edited script
echo "#Running Cutadapter... file: $FileToTrim Time left:${EstimatedtimetoFinishMins}mins
(${EstimatedtimetoFinishHours}hrs) ${PercentComplete}% complete"
echo $PercentComplete
~/GGOSS/Scripts/Sickle_Run.sh

#delete script

if [ -f ~/GGOSS/Scripts/Sickle_Run.sh ];then
rm ~/GGOSS/Scripts/Sickle_Run.sh
fi

#need to close window
pkill yad

echo "Completed paired end trimming of files $FileToTrimRead1 and $FileToTrimRead2"
notify-send "Completed paired end trimming of files $FileToTrimRead1 and $FileToTrimRead2"

else

echo "Failed to find the selected file: $FileToTrim"
notify-send "Failed to find the selected file: $FileToTrim"

fi

#+2 to account for pair
line=$(( $line + 2 ))

done

fi

else

echo "WARNING! PROBLEM:

Odd number of files! Paired end adapter trimming requires pairs

At least 1 file is missing and needs to be included.

-----

Program has not continued and no files have been trimmed. Please exclude the additional file or find
and include the missing file before trying again
"

#checks more thoroughly to find the problem
AlphabetSortedFiles=$(sort ~/GGOSS/tmp/Sickle_SelectedFiles.txt)

FileLine=1
for i in $(seq 1 $NumberOfFiles)
do
#takes the name of the file
FileName=$(echo "$AlphabetSortedFiles" | awk -v y=$FileLine 'NR==y {print $1}' | awk -F
'_' '{NF-=4; OFS="_"; print}')

```

```

#counts how often that name comes up into the file list
FileNameFrequency=$(grep -c $FileName ~/GGOSS/tmp/Sickle_SelectedFiles.txt)

if [ "$FileNameFrequency" = 2 ];then
echo "Searching for problem files... no pairing error found with: $FileName"
else
echo "Searching for problem files... an error in file pairing found! File: ${FileName}, has
$FileNameFrequency match/es, there should only be 2.
Please resolve this file issue and retry. Note that it may be your pairing file sample name is not
named the same for some reason, or that the sample name contains an '_', if this is the case, adjusting
this will resolve the problem"
FilePairError=1
fi

FileLine=$(( $FileLine + 1 ))
done

fi

if [ -f ~/GGOSS/Scripts/Sickle_Run.sh ];then
rm ~/GGOSS/Scripts/Sickle_Run.sh
fi

else

echo "Still need to build single end"

fi

} | tee ~/GGOSS/LogFiles/Sickle_LOGFILE.txt | yad --progress --auto-close --auto-kill --center --
width=700 --image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- Sickle
GGOSS created by Giles Holt" --text="Running Sickle

Settings:
Sequencer type = $SickleCompanySelected
End type = $RunType
Quality Threshold = $QualityThreshold
Length Threshold = $LengthThreshold
Interleaved = $Interleaved
"

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Sickle" --
button="Open file location":5 --button="Return to menu":4 --text="

Sickle Complete

"

mode="$?"
case $mode in
4)~/GGOSS/GenomicsProgram.sh ;;
5)nautilus ~/GGOSS_InputOutput/Sickle & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.2 Genome assembly scripts

10.9.3.2.1 GGOSS scripts for SPAdes assembly

10.9.3.2.1.1 SPAdes template

```
#!/bin/sh
#Replace the path to SPAdes with the below variable
PathToSPAdes=$(awk -F '|' '{print $7}' ~/GGOSS/tmp/PathToTools_SettingsChange.txt)
if [ "$PathToSPAdes" = "N/A" ];then
PathToSPAdes="$HOME/"
fi
echo "Path to SPAdes: $PathToSPAdes"

echo | ${PathToSPAdes}spades.py SingleCell ErrorCorrectionOnly AssemblyModuleOnly
MetaSPAdes KhmerValues SetCareful ThreadNumber MemoryAmount CoverageCutoff
PHREDQualityOffset MachineType LibraryType FileInputMethod -o
~/GGOSS_InputOutput/SPAdes/FileOutputFilename
```

10.9.3.2.1.2 SPAdes command creation from settings

```
#!/bin/bash

cp ~/GGOSS/Scripts/SPAdesAssemblerTemplate.sh ~/GGOSS/Scripts/SPAdesAssembler.sh

#CORE CHANGES

#Run Read error correction only
ErrorCorrectionOnly=$(awk -F '[' '{print $9}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ $ErrorCorrectionOnly = "No" ];then
sed -i -e "s/ErrorCorrectionOnly \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/ErrorCorrectionOnly\+/\-only\-error\-correction/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
fi

AssemblyModuleOnly=$(awk -F '[' '{print $10}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ $AssemblyModuleOnly = "No" ];then
sed -i -e "s/AssemblyModuleOnly \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/AssemblyModuleOnly\+/\-only\-assembler/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#change to metaSPAdes according to settings
SettingsMetaSPAdes=$(awk -F '[' '{print $4}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)
```

```

if [ $SettingsMetaSPAdes = "Yes" ];then
sed -i -e "s/MetaSPAdes\+/\-meta/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/MetaSPAdes \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#change Khmer according to settings
SettingsKvalue=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$SettingsKvalue" != "-" ];then
sed -i -e "s/KhmerValues\+/\-k $SettingsKvalue/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/KhmerValues \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#processor number
SettingsProcessors=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$SettingsProcessors" = - ];then
sed -i -e "s/ThreadNumber\+/\-t $(nproc)/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
#change the current script number to that
sed -i -e "s/ThreadNumber\+/\-t $SettingsProcessors/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#set memory limit according to settings
SettingsMemory=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$SettingsMemory" != "-" ];then
sed -i -e "s/MemoryAmount\+/\-m $SettingsMemory/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/MemoryAmount \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#careful setting
SetCareful=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ $SetCareful = "Yes" ];then
sed -i -e "s/SetCareful\+/\-careful/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/SetCareful \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#Data from machine type
MachineType=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$MachineType" != "Illumina" ];then
sed -i -e "s/MachineType\+/\-MachineType/g" ~/GGOSS/Scripts/SPAdesAssembler.sh

else
sed -i -e "s/MachineType \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

```

```

#read coverage cutoff value
CoverageCutoff=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$CoverageCutoff" = "off" ];then
sed -i -e "s/CoverageCutoff \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/CoverageCutoff\+/\-cov\-\cutoff $CoverageCutoff/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#PHRED quality offset for input reads
PHREDQualityOffset=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$PHREDQualityOffset" != "-" ];then
sed -i -e "s/PHREDQualityOffset\+/\-phred\-\offset $PHREDQualityOffset/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/PHREDQualityOffset \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh

fi

#single cell mode
SingleCell=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ $SingleCell = "No" ];then
sed -i -e "s/SingleCell \+//g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/SingleCell\+/\-sc/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

#Library numeracy

LibraryNumeracy=$(awk -F '[]' '{print $7}' ~/GGOSS/tmp/SPAdesSettingsChange.txt)

if [ "$LibraryNumeracy" = "One Library" ]
then
#Single numeracy section
LibraryType=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
if [ "$LibraryType" = "Interlaced forward and reverse paired-end reads" ]
then
sed -i -e "s/LibraryType FileInputMethod\+/\-12 FilePath\FastqFileInput/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
else
if [ "$LibraryType" = "Seperate sample file for forward and reverse reads" ]
then
sed -i -e "s/LibraryType FileInputMethod\+/\-1 FilePath\FileInput1FileName -2
FilePath\FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
else
sed -i -e "s/LibraryType FileInputMethod\+/\-s FilePath\FileInputFileName/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
fi
fi
else

```

```
#Multiple numeracy section
```

```
#find which multiple type was selected
SingleReadLibraries=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
PairedEndLibraries=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
MatePairLibraries=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
HighQualityMatePairLibraries=$(awk -F '[]' '{print $4}'
~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
LucigenNxSeqLongMatePairLibraries=$(awk -F '[]' '{print $5}'
~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
```

```
#Single
```

```
if [ "$SingleReadLibraries" = "Yes" ]
then
    sed -i -e "s/LibraryType FileInputMethod\+/\-s1 FilePath\FastqFileInput/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
```

```
fi
```

```
#Paired End
```

```
if [ "$PairedEndLibraries" != "Not applicable" ];then
    if [ "$PairedEndLibraries" = "Interlaced forward and reverse paired-end reads" ];then
        sed -i -e "s/LibraryType FileInputMethod\+/\-pe1\12 FilePath\FastqFileInput/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
    else
        if [ "$PairedEndLibraries" = "Seperate sample file for forward and reverse reads" ];then
            sed -i -e "s/LibraryType FileInputMethod\+/\-pe1\1 FilePath\FileInput1FileName --pe1-2
FilePath\FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
        else
            sed -i -e "s/LibraryType FileInputMethod\+/\-pe1\s FilePath\FileInputFileName/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
```

```
    fi
```

```
    fi
```

```
fi
```

```
#Mate-pair libraries
```

```
if [ "$MatePairLibraries" != "Not applicable" ]
then
    if [ "$MatePairLibraries" = "Interlaced forward and reverse paired-end reads" ]
    then
        sed -i -e "s/LibraryType FileInputMethod\+/\-mp1\12 FilePath\FastqFileInput/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
    else
        if [ "$MatePairLibraries" = "Seperate sample file for forward and reverse reads" ]
        then
            sed -i -e "s/LibraryType FileInputMethod\+/\-mp1\1 FilePath\FileInput1FileName \-mp1\2
FilePath\FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
```

```
        fi
```

```
        fi
```

```
fi
```

```
#High-quality mate-pair
```

```
if [ "$HighQualityMatePairLibraries" != "Not applicable" ]
```

```

then
  if [ "$HighQualityMatePairLibraries" = "Interlaced forward and reverse paired-end reads" ]
  then
    sed -i -e "s/LibraryType FileInputMethod\+/\-hpmp1\-12 FilePath\FastqFileInput/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
  else
    if [ "$HighQualityMatePairLibraries" = "Seperate sample file for forward and reverse reads" ]
    then
      sed -i -e "s/LibraryType FileInputMethod\+/\-hpmp1\-1 FilePath\FileInput1FileName \-hpmp1\-2 FilePath\FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
    else
      sed -i -e "s/LibraryType FileInputMethod\+/\-hpmp1\-s FilePath\FileInputFileName/g"
~/GGOSS/Scripts/SPAdesAssembler.sh
    fi
  fi
fi

#Lucigen NxSeq Long Mate Pair libraries
if [ "$LucigenNxSeqLongMatePairLibraries" != "Not applicable" ]
then
  sed -i -e "s/LibraryType FileInputMethod\+/\-nxmate1\-1 FilePath\FileInput1FileName \-nxmate1\-2 FilePath\FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh
fi

fi

```

```

#sets the file name containing those files
FilePath=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

sed -i -e "s|FilePath|${FilePath}|g" ~/GGOSS/Scripts/SPAdesAssembler.sh

```

10.9.3.2.1.3 SPAdes script for imaging selected settings whilst running

```

#!/bin/bash

#script to check for and (if necessary) make setting changes to the assembly script prior running
~/GGOSS/Scripts/SPAdesSettingsChangeScript.sh

#change settings tmp text into individual lines per setting
sed 's|/\n/g' ~/GGOSS/tmp/SPAdesSettingsChange.txt >
~/GGOSS/tmp/SPAdesSettingsChangeList.txt

#Add specific text to start of each line
sed -i '1 s/^Data from machine type: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt

sed -i '2 s/^Read coverage cutoff value: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt

sed -i '3 s/^PHRED quality offset for input reads: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt

sed -i '4 s/^Khmer values selected are: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt

```

```

sed -i '5 s/^Number of processors set at: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '6 s/^Memory limit set to: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '7 s/^Library numeracy: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '8 s/^Single cell data: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '9 s/^Run read error correction only: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '10 s/^Run assembly module only: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '11 s/^Reduce mismatch and short indel number, and run MismatchCorrector: /'
~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '12 s/^Trusted contigs: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '13 s/^Untrusted contigs: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '14 s/^Assemble with MetaSPAdes: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '15 s/^Assemble plasmid: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt
sed -i '16 s/^Assemble RNA-seq data: /' ~/GGOSS/tmp/SPAdesSettingsChangeList.txt

```

```

#An opening yad window explaining assembly is starting under the settings they have chosen
Settings=$(cat ~/GGOSS/tmp/SPAdesSettingsChangeList.txt)

```

```

yad --title="GILES -- Assembly - Running SPAdes under the following settings" --width=800 --
length=800 --center --sticky --on-top --no-buttons --no-escape --text="$Settings" --text-align=left --
timeout=2

```

```

sleep 1

```

```

~/GGOSS/Scripts/LoopTheSPAdesAssemblerScript.sh

```

10.9.3.2.1.4 GGOSS script for running SPAdes

```

#!/bin/sh
if [ -f ~/GGOSS/LogFiles/SPAdesAssembly_LOGFILE.txt ];then
rm ~/GGOSS/LogFiles/SPAdesAssembly_LOGFILE.txt
fi
{

PercentCompleteStart=1
echo $PercentCompleteStart
echo "#Checking files and applying settings      ${PercentCompleteStart}% complete"

```

```

#automatically open logfile upon completion, get the progress bar to work

#sorts them so they are in order, as R1 and R2 names are identical except for the 1 or 2, they will
be sure to be grouped together, allowing to select them by the line and the line +1
sort ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt

##### CHECKS IF USING FORWARD AND REVERSE OR NOT, AND ADAPTS
SingleReadLibraries=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
PairedEndLibraries=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
MatePairLibraries=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
HighQualityMatePairLibraries=$(awk -F '[]' '{print $4}'
~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)
LucigenNxSeqLongMatePairLibraries=$(awk -F '[]' '{print $5}'
~/GGOSS/tmp/SPAdesSettingsLibraryType.txt)

if [ "$SingleReadLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$PairedEndLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$MatePairLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$HighQualityMatePairLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$LucigenNxSeqLongMatePairLibraries" = "Seperate sample file for forward and reverse reads" ]
then
    NumberOfSamplesToRunPrep=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)
    NumberOfSamplesToRun=$(echo "$NumberOfSamplesToRunPrep / 2" | bc)

else
    NumberOfSamplesToRun=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)
fi

Settings=$(cat ~/GGOSS/tmp/SPAdesSettingsChangeList.txt)

UnpairedNumberOfSamplesToRun=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)

Time1min=$(date +%M)
Time1hourtmp=$(date +%H)

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

PercentComplete=1

line=1
for i in $(seq $NumberOfSamplesToRun);do

echo "" >> ~/SolveSpades.txt
echo "Start Of loop, Loop number: $line" >> ~/SolveSpades.txt
echo "" >> ~/SolveSpades.txt

PercentCompleteOnfinishingThisSample=$( echo "scale=2; ( $line /
$UnpairedNumberOfSamplesToRun ) * 100" | bc | awk -F '.' '{print $1}' )

Time2min=$(date +%M)

```

```

Time2hourtmp=$(date +%H")
Time2hour=$( echo "$Time2hourtmp * 60" | bc )
EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )
TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )
NumberFilesLeftToDo=$( echo "$UnpairedNumberOfSamplesToRun - $line" | bc )
EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberFilesLeftToDo" | bc )
AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )
EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

lineR2=$(( $line + 1 ))
filenameR1=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)
filenameR2=$(awk -v y=$lineR2 'NR==y {print}' ~/GGOSS/tmp/SPAdesSelectedFilesP4.txt)

#this will only set the filename correctly based on illumina underscoring atm, check other output file
types for ion torrent etc
filename=$(echo $filenameR1 | awk -F '_' '{NF-=4; OFS="_"; print}')

    if [ -f ~/GGOSS_InputOutput/SPAdes/SPAdes_$filename ]
    then
        echo "Assembly output file for $filename already exists, if you wish to re-run this samples
please delete or move its pre-existing SPAdes output. Running the next sample
"
        echo | notify-send "Assembly output file for $filename already exists, if you wish to re-run
this samples please delete or move its pre-existing SPAdes output. Running the next sample
"
    fi

FilePath=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

    PercentCompleteStart=$(echo "2 + $PercentComplete" | bc)
    echo $PercentCompleteStart
    echo "#Complete - checking files and applying settings      ${PercentCompleteStart}%
complete"
    echo "#Running SPAdes assembly for file: $filename      ${PercentCompleteStart}%
complete"

    #change th SPAdes assembler script for the filename at hand
    #change file 1
    sed -i -e "s/FileInput1FileName\+/$filenameR1/g" ~/GGOSS/Scripts/SPAdesAssembler.sh

    #change file 2
    sed -i -e "s/FileInput2FileName\+/$filenameR2/g" ~/GGOSS/Scripts/SPAdesAssembler.sh

    #change output file

```

```

        sed -i -e "s/FileOutputFilename\+/SPAdes_${filename}/g"
~/GGOSS/Scripts/SPAdesAssembler.sh

Prevent2ndRunthruIf=1

GapFiller=$(echo "Running SPAdes assembly for file:")

##### monitors progress through an assembly
touch ~/SolveSpades.txt

~/GGOSS/Scripts/SPAdesAssembler.sh | tee ~/GGOSS/LogFiles/SPAdesAssembly_LOGFILE.txt |
while read -r CheckForMarker
do

    echo "" >> ~/SolveSpades.txt

    variable=$(echo "$CheckForMarker")
    echo "#####-----variable: $variable" >>
~/SolveSpades.txt
    echo "#####-----NumberKvalueRuns:
$NumberKvalueRuns" >> ~/SolveSpades.txt

    if [ "$Prevent2ndRunthruIf" = "1" ];then
    echo "If:1  Should only happen until If:2" >> ~/SolveSpades.txt
    NumberKvalueRuns=$(grep '^ k: ' ~/GGOSS/LogFiles/SPAdesAssembly_LOGFILE.txt | awk -F ' '
'{'print NF; exit}')
    NumberKvalueRunsFromSettings=$(awk -F '[]' '{print $4}'
~/GGOSS/tmp/SPAdesSettingsChange.txt | awk -F ' ' '{print NF; exit}')
    fi

    if [ "$NumberKvalueRunsFromSettings" = "$NumberKvalueRuns" ] && [ "$Prevent2ndRunthruIf"
= "1" ];then
        Prevent2ndRunthruIf=2
        echo "If:2  Should only happen once" >> ~/SolveSpades.txt
        echo "#####-----NumberKvalueRuns:
$NumberKvalueRuns" >> ~/SolveSpades.txt
        PercentToBeSplitOutOverSample=$( echo "$PercentCompleteOnfinishingThisSample -
$PercentCompleteStart" | bc )
        echo "#####-----
PercentToBeSplitOutOverSample: $PercentToBeSplitOutOverSample" >> ~/SolveSpades.txt
        PercentToAddAtEachKvalue=$( echo "$PercentToBeSplitOutOverSample / $NumberKvalueRuns"
| bc )
        echo "#####-----
PercentToAddAtEachKvalue: $PercentToAddAtEachKvalue" >> ~/SolveSpades.txt
        PercentComplete=$( echo "$PercentCompleteStart + $PercentToAddAtEachKvalue" | bc )
        echo "#####-----PercentComplete:
$PercentComplete" >> ~/SolveSpades.txt
        fi

    if [ "$PassedPreviousPoint" != 1 ];then
        echo "If:3  Should happen continuesly until If:5" >> ~/SolveSpades.txt
        echo "#####-----
PassedPreviousPoint: $PassedPreviousPoint - doesn't = 1  " >> ~/SolveSpades.txt

```

```

variable3=$(echo "$variable" | awk -F ' ' '{print $1}')
echo "variable3: $variable3" >> ~/SolveSpades.txt
  if [ "$variable3" = "==== Read error correction started" ];then
    echo "If:4a Should only happen once"
    GapFiller=$(echo "Running Read error correction...")
    echo "#####-----GapFiller: $GapFiller"
>> ~/SolveSpades.txt
  echo $PercentComplete
  echo "#Running Read error correction... file: $filename  ${PercentComplete}% complete"
  else
    #this is where i think the program is getting to
    echo "#$GapFiller file: $filename  ${PercentComplete}% complete"
    echo "If:4b Should happen continuesly until If:5"
  fi
fi
#trim the variable
#else
variable2=$(echo "$variable" | awk -F ' ' '{print $1}')
  if [ "$variable2" = "== Running assembler" ];then
    echo "If:5 Should only happen on occasions where line says running spades assembler" >>
~/SolveSpades.txt
  echo "#####----- Running assembler
match " >> ~/SolveSpades.txt
  PassedPreviousPoint=1
  PercentComplete=$( echo "$PercentComplete + $PercentToAddAtEachKvalue" | bc )
  Kvalue=$(echo "$variable" | awk -F ' ' '{print $2}')
  GapFiller=$(echo "Running assembler: $Kvalue...")
  echo $PercentComplete
  echo "#Running assembler: $Kvalue... file: $filename  ${PercentComplete}% complete"
  else
    echo "#$GapFiller file: $filename  ${PercentComplete}% complete"
  fi

  echo "" >> ~/SolveSpades.txt
#####

done
echo "outside first loop" >> ~/SolveSpades.txt
  #pkill yad

  #change back ready for the next file type
  #change file 1
  sed -i -e "s/$filenameR1\+/FileInput1FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh

  #change file 2
  sed -i -e "s/$filenameR2\+/FileInput2FileName/g" ~/GGOSS/Scripts/SPAdesAssembler.sh

  #change output file
  sed -i -e "s/SPAdes_$filename\+/FileOutputFilename/g"
~/GGOSS/Scripts/SPAdesAssembler.sh

PercentCompletenessprep=$( echo "scale=2; ( $line / $UnpairedNumberOfSamplesToRun ) * 100" | bc |
awk -F ' ' '{print $1}' )
PercentComplete=$( echo "$PercentCompleteStart + $PercentCompletenessprep" | bc )

```

```

if [ "$SingleReadLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$PairedEndLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$MatePairLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$HighQualityMatePairLibraries" = "Seperate sample file for forward and reverse reads" ] || [
"$LucigenNxSeqLongMatePairLibraries" = "Seperate sample file for forward and reverse reads" ]
then
line=$(( $line + 2 ))
else
line=$(( $line + 1 ))
fi

if [ -f ~/GGOSS_InputOutput/SPAdes/$filename/contigs.fasta ] || [
~/GGOSS_InputOutput/SPAdes/$filename/scaffolds.fasta ]
then
echo $PercentComplete
echo "#SPAdes assembly complete for file: $filename      ${PercentComplete}% complete"
else
echo $PercentComplete
echo "#Failed to assemble sample: $filename      ${PercentComplete}% complete"
fi

echo "" >> ~/SolveSpades.txt
echo "End Of 1 Sample in Loop, Loop number: $line" >> ~/SolveSpades.txt
echo "" >> ~/SolveSpades.txt

done

echo ""
echo "End Of Loop (after done)" >> ~/SolveSpades.txt
echo ""

pkill yad

} | yad --progress --auto-kill --center --width=700 --image=$ICON --image-on-top --title="GENOME
SEQUENCING PROGRAM -- SPAdes - Assembler      GGOSS created by Giles Holt" --
text="Running SPAdes assembly
Start time:${Time1hourtmp}:${Time1min}
Running sample: ${filename} ($line of $NumberOfSamplesToRun)
Time elapsed: ${TimeTaken} minutes
Estimated time remaining: $EstimatedtimetoFinishMins minutes (${EstimatedtimetoFinishHours}
hours)
Average time taken per sample: $AverageTimeTakenPerSample

Please check the SPAdes assembly Logfile to
confirm the success of assembly on your samples.
You can also find other information there,
including the version of SPAdes that was used
Settings:
$Settings
"
# -----

```

```
#####
#####
# -----

echo "SPAdes script complete"

echo "
"

yad --auto-kill --center --width=300 --image-on-top --title="GGOSS -- SPAdes" --button="Open file
location":5 --button="Return to menu":4 --text="

                SPAdes Complete

"

mode="$?"
case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/SPAdes & ~/GGOSS/GenomicsProgram.sh ;;
esac
```

10.9.3.2.2 GGOSS scripts for Velvet

10.9.3.2.2.1 Velvet template

```
#!/bin/bash

$HOME/velvet_1.2.10/contrib/shuffleSequences_fasta/VelvetScriptType
~/GGOSS_InputOutput/FastqFiles/FileInput1FileName
~/GGOSS_InputOutput/FastqFiles/FileInput2FileName
~/GGOSS_InputOutput/Velvet/ShuffledFiles/FileOutputFilename
```

10.9.3.2.2.2 GGOSS selected settings script editor

```
#!/bin/bash

cp ~/GGOSS/Scripts/VelvetAssemblerTemplate.sh ~/GGOSS/Scripts/VelvetAssembler.sh

ShuffleFiles=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/VelvetSettings.txt)

if [ $ShuffleFiles = "No" ];then
sed -i -e "s/VelvetScriptType \+//g" ~/GGOSS/Scripts/VelvetAssembler.sh
else
sed -i -e "s/VelvetScriptType\+/shuffleSequences_fastq.pl/g" ~/GGOSS/Scripts/VelvetAssembler.sh
fi

#change settings tmp text into individual lines per setting
sed 's/|\n/g' ~/GGOSS/tmp/VelvetSettings.txt > ~/GGOSS/tmp/VelvetSettingsChangeList.txt
```

```

#Add specific text to start of each line
sed -i '1 s/^/Velvet Run type: /' ~/GGOSS/tmp/VelvetSettingsChangeList.txt

#An opening yad window explaining assembly is starting under the settings they have chosen
Settings=$(cat ~/GGOSS/tmp/VelvetSettingsChangeList.txt)

yad --title="GILES -- Assembly - Running Velvet under the following settings" --width=800 --
length=800 --center --sticky --on-top --no-buttons --no-escape --text="$Settings" --text-align=left --
timeout=2

sleep 1

~/GGOSS/Scripts/LoopTheVelvetAssemblerScript.sh

```

10.9.3.2.2.3 GGOSS script for velvet

```

#!/bin/sh

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                Velvet run from GGOSS Genomics program"
"
echo | (date +"                Velvet Start Date: %d-%m-%y Time: %T

")

yad --title="GENOME SEQUENCING PROGRAM -- Velvet                Created by Giles Holt" --
timeout=10 --no-buttons --no-escape --width=300 --length=300 --text="Please check the Velvet
assembly Logfile to ascertain and

confirm the success of assembly on your samples. You can

also find other information there, including the version of Velvet that was used

" &

#sorts them so they are in order, as R1 and R2 names are identical except for the 1 or 2, they will
be sure to be grouped together, allowing to select them by the line and the line +1
sort ~/GGOSS/tmp/VelvetSelectedFiles.txt

##### As using forward and reverse it calculates the actual number of run throughs
NumberOfSamplesToRunPrep=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/VelvetSelectedFiles.txt)
NumberOfSamplesToRun=$(echo "$NumberOfSamplesToRunPrep / 2" | bc)

Settings=$(cat ~/GGOSS/tmp/VelvetSettings.txt)

```

```

UnpairedNumberOfSamplesToRun=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/VelvetSelectedFiles.txt)

Time1min=$(date +%M)
Time1hourtmp=$(date +%H)

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

line=1
for i in $(seq 1 $NumberOfSamplesToRun)
do

Time2min=$(date +%M)
Time2hourtmp=$(date +%H)

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$UnpairedNumberOfSamplesToRun - $line" | bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberFilesLeftToDo" | bc )

AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )
lineR2=$(( $line + 1 ))
filenameR1=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/VelvetSelectedFiles.txt)
filenameR2=$(awk -v y=$lineR2 'NR==y {print}' ~/GGOSS/tmp/VelvetSelectedFiles.txt)

#this will only set the filename correctly if there is an underscore after the filename
filename="$filenameR1"

    if [ -f ~/GGOSS_InputOutput/Velvet/Velvet_$filename ]
    then
        echo "Assembly output file for $filename already exists, if you wish to re-run this samples
please delete or move its pre-existing Velvet output. Running the next sample
"
        echo | pkill yad
        echo | notify-send "Assembly output file for $filename already exists, if you wish to re-run
this samples please delete or move its pre-existing Velvet output. Running the next sample
"
    fi

    if [ -f ~/GGOSS_InputOutput/FastqFiles/$filenameR1 ] && [ -f
~/GGOSS_InputOutput/FastqFiles/$filenameR2 ]
    then
        notify-send "Running Velvet assembly for file: $filename"

        echo "Running Velvet assembly for file: $filename
"

```

```

#change the Velvet assembler script for the filename at hand
#change file 1
sed -i -e "s/FileInput1FileName\+/$filenameR1/g" ~/GGOSS/Scripts/VelvetAssembler.sh

#change file 2
sed -i -e "s/FileInput2FileName\+/$filenameR2/g" ~/GGOSS/Scripts/VelvetAssembler.sh

#change output file
sed -i -e "s/FileOutputFilename\+/Velvet_$filename/g"
~/GGOSS/Scripts/VelvetAssembler.sh

cat ~/GGOSS/Scripts/VelvetAssembler.sh

yad --title="GGOSS -- Velvet" --width=700 --center --sticky --on-top --no-buttons --no-escape --
text-align=center --text=" Running Velvet

Total number of samples for Velvet to run: $NumberOfSamplesToRun

Currently running Velvet on file: ${filename}. Started at:${Time2hourtmp}:${Time2min}

Time taken thus far: ${TimeTaken} minutes
Average time taken per sample:$AverageTimeTakenPerSample

Estimated time left until completion:$EstimatedtimetoFinishMins minutes
(${EstimatedtimetoFinishHours} hours)

Settings:
$Settings
" &

~/GGOSS/Scripts/VelvetAssembler.sh

pkill yad

#change back ready for the next file type

#change file 1
sed -i -e "s/$filenameR1\+/FileInput1FileName/g" ~/GGOSS/Scripts/VelvetAssembler.sh

#change file 2
sed -i -e "s/$filenameR2\+/FileInput2FileName/g" ~/GGOSS/Scripts/VelvetAssembler.sh

#change output file
sed -i -e "s/Velvet_$filename\+/FileOutputFilename/g" ~/GGOSS/Scripts/VelvetAssembler.sh

else
notify-send "Can't find the $filename file selected for assembly, either a significant error has
occured or the file name or location has been tampered with since starting the run.

Trying next sample"

echo "Can't find the $filename file selected for assembly, either a significant error has
occured or the file name or location has been tampered with since starting the run.

```

```

                                Trying next sample
"
    fi

line=$(( $line + 2 ))
done

} | tee ~/GGOSS/LogFiles/VelvetAssembly_LOGFILE.txt

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Velvet" --
button="Open file location":5 --button="Return to menu":4 --text="

```

Velvet Complete

```

"

mode="$?"
case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/Velvet & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.2.3 QUAST

10.9.3.2.3.1 QUAST template

```

#!/bin/bash

quast.py -o ~/GGOSS_InputOutput/QUAST/SampleName/
FilePath/SampleNameFolder/AssemblyFileType

```

10.9.3.2.3.2 GGOSS script for QUAST

```

#!/bin/bash

{
echo "                                GGOSS Genomics - Created by Giles Holt
"
echo "                                QUAST run from GGOSS Genomics program
"
echo |(date +"                                QUAST Run Start Date: %d-%m-%y Time: %T

")

Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

```

```

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/QUASTSelectedFiles.txt)

#####-----#####

##--# Edit Script according to settings

#####-----#####

cp ~/GGOSS/Scripts/QUASTTemplate.sh ~/GGOSS/Scripts/QUASTRunPrep.sh

File=$(awk -F '[' '{print $13}' ~/GGOSS/tmp/QUAST_SettingsChange.txt)

if [ "$File" = "Contig" ];then
sed -i -e "s/AssemblyFileType\+/contigs.fasta/g" ~/GGOSS/Scripts/QUASTRunPrep.sh
else
sed -i -e "s/AssemblyFileType\+/scaffolds.fasta/g" ~/GGOSS/Scripts/QUASTRunPrep.sh
fi

#####-----#####

##---# Edit and run Script according to samples selected #---##

#####-----#####

line=1
for i in $(seq 1 $NumberOfFiles)
do

cp ~/GGOSS/Scripts/QUASTRunPrep.sh ~/GGOSS/Scripts/QUASTRun.sh

#change QUASTRun in accordance to settings,then run it on file, make your way through all the files
using this loop

#base changes are input and output

#Change the 'InputFilePath' in QUASTRun.sh file to the selected file path
FilePath=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

sed -i -e "s|FilePath|$FilePath|g" ~/GGOSS/Scripts/QUASTRun.sh

#Set the input file as variable
FilePrep=$(awk -v y=$line 'NR==y {print}' ~/GGOSS/tmp/QUASTSelectedFilesP4.txt)
#Change the output 'SampleName' in the QUASTRun.sh file to the sample name variable
sed -i -e "s/SampleNameFolder\+/${FilePrep}/g" ~/GGOSS/Scripts/QUASTRun.sh

#Set sample name from input file name (Maybe remove the scaffold part of the name)
FileName=$(echo "$FilePrep" | awk -F '.' '{print $1}')

```

```

#Create a folder in QUASt with the sample name
mkdir ~/GGOSS_InputOutput/QUAST/"${FileName}"

#Change the output 'SampleName' in the QUAStRun.sh file to the sample name variable
sed -i -e "s/SampleName\+/${FileName}/g" ~/GGOSS/Scripts/QUASTRun.sh

Time2min=$(date +%M)
Time2hourtmp=$(date +%H)

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$NumberOfFiles - $line" | bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberOfFilesLeftToDo" | bc )

AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

yad --title="GGOSS -- QUASt" --width=400 --center --sticky --on-top --no-buttons --no-escape --
text-align=center --text="  Running QUASt

Total number of files for QUASt to run: $NumberOfFiles

Currently  running  QUASt  on  file:  ${line},  ${FileToTrimRead1}.  Started
at:${Time2hourtmp}:${Time2min}

Time taken thus far: ${TimeTaken} minutes
Average time taken per sample:$AverageTimeTakenPerSample

Estimated  time  left  until  completion:$EstimatedtimetoFinishMins  minutes
(${EstimatedtimetoFinishHours} hours)

Settings:
" &

#Run edited script

~/GGOSS/Scripts/QUASTRun.sh

#delete script

if [ -f ~/GGOSS/Scripts/QUASTRun.sh ]
then
rm ~/GGOSS/Scripts/QUASTRun.sh
fi

#need to close window
pkill yad

```

```

line=$(( $line + 1 ))

done

} | tee ~/GGOSS/LogFiles/QUAST_LOGFILE.txt

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- QUAST" --
button="Open file location":5 --button="Return to menu":4 --text="

                QUAST Complete

"

mode="$?"
case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/QUAST & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.2.3.3 GGOSS tabulation script for QUAST output

```

#!/bin/bash

#prep the first file # by changing the second column title, contig, to 'sample name contig' Output as a
new table txt file

PathToSamples=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)
PathToSamples2=$(awk -F '/' 'NR==1 {print $2}' ~/GGOSS/tmp/Path2selectedFile.txt)
PathToSamples3=$(awk -F '/' 'NR==1 {print $3}' ~/GGOSS/tmp/Path2selectedFile.txt)
FirstFileName=$(awk 'NR==1 {print}' ~/GGOSS/tmp/QUASTSelectedFilesP4.txt)

#number of files selected from quast output
NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/QUASTSelectedFilesP4.txt)

cat $HOME/${PathToSamples2}/${PathToSamples3}/${FirstFileName}/report.tsv | awk -F ';' -v
x="${FirstFileName}" 'NR==1 {$2=x}1' >
~/GGOSS_InputOutput/QUAST/QUAST_tables/${NumberOfFiles}Sample_QuastTable.csv

#Start loop from second file
line=2
for i in $(seq 1 $NumberOfFiles);do

#take next column and add to table file
FileName=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/QUASTSelectedFilesP4.txt)

#take the columns title, contig, and replace with sample name contig
cat $HOME/${PathToSamples2}/${PathToSamples3}/${FileName}/report.tsv | awk -F ';' -v
x="${FileName}" 'NR==1 {$2=x}1' | awk -F '\t' '{print $2}' >
~/GGOSS/tmp/ColumnBuild_QuastTable.csv

paste ~/GGOSS_InputOutput/QUAST/QUAST_tables/${NumberOfFiles}Sample_QuastTable.csv
~/GGOSS/tmp/ColumnBuild_QuastTable.csv > ~/GGOSS/tmp/ColumnBuild1_QuastTable.csv

```

```
mv ~/GGOSS/tmp/ColumnBuild1_QuastTable.csv
~/GGOSS_InputOutput/QUAST/QUAST_tables/${NumberOfFiles}Sample_QuastTable.csv
```

```
line=$(( $line + 1 ))
```

```
done
```

10.9.3.3 Mapping

10.9.3.3.1 GGOSS script for BWA

```
#!/bin/sh
#paired end nextera xt

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                BWA run from GGOSS Genomics program"
"
echo |(date +"                BWA Start Date: %d-%m-%y Time: %T

")

echo 0
echo "#Running BWA...                0% complete"

Time1min=$(date +"%M")
Time1hourtmp=$(date +"%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

#checks more thoroughly
#variable of all files sorted alphabetically
AlphabetSortedFiles=$(sort ~/GGOSS/tmp/BWA_SelectedFiles.txt)
RunType=$(awk -F '|' 'NR==1 {print $1}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
SAMtools=$(awk -F '|' 'NR==1 {print $2}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
RefGenome=$(awk -F '|' 'NR==1 {print $3}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
PairedOrSingle=$(awk -F '|' 'NR==1 {print $6}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
PathToFiles=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

if [ "$PairedOrSingle" = "Single" ];then
NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/BWA_SelectedFiles.txt)
else
NumberOfFilesPrep=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/BWA_SelectedFiles.txt)
NumberOfFiles=$(echo "$NumberOfFilesPrep / 2" | bc)
fi
```

```

if [ "$RunType" = "Not applicable" ];then
RunTypeCustom1=$(awk -F '|' 'NR==1 {print $4}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
RunTypeCustom2=$(awk -F '|' 'NR==1 {print $5}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
echo $RunTypeCustom1
echo $RunTypeCustom2

if [ -f ~/GGOSS/tmp/Custom1Flags.txt ];then
rm ~/GGOSS/tmp/Custom1Flags.txt
fi
if [ -f ~/GGOSS/tmp/Custom2Flags.txt ];then
rm ~/GGOSS/tmp/Custom2Flags.txt
fi

#####
## ----- samse
#####

#Runs custom prep
if [ "$RunTypeCustom1" = "samse" ] || [ "$RunTypeCustom2" = "samse" ];then
#Adapt for making custom 2 txt file if its been selected
if [ "$RunTypeCustom2" = "samse" ];then
CustomFlags="Custom2Flags.txt"
else
CustomFlags="Custom1Flags.txt"
fi

#create temp file containing all flags
#Add the - Flag in front of each text in each column only if column contents doesn't equal -, will
need some kind of loop
FlagSetting=7
FlagLetters="n|r"
Letter=1
for i in $(seq 2);do
#I specify line number as each setting menu adds to this temp file but on a different line
FlagNumber=$(awk -F '|' -v x=$FlagSetting 'NR==1 {print $x}'
~/GGOSS/tmp/BWA_SettingsChange.txt)
Flag=$(echo "$FlagLetters" | awk -F '|' -v y=$Letter '{print $y}')
echo "FlagNumber: $FlagNumber"
if [ "$FlagNumber" = "Yes" ];then
echo -n "-${Flag} " >> ~/GGOSS/tmp/${CustomFlags}
else
if [[ "$FlagNumber" > 0 ]] && [ "$FlagNumber" != "No" ];then
echo -n "-${Flag} $FlagNumber " >> ~/GGOSS/tmp/${CustomFlags}
fi
fi
Letter=$(( $Letter + 1 ))
FlagSetting=$(( $FlagSetting + 1 ))
done

fi

#####
## ----- sampe
#####

```

```

if [ "$RunTypeCustom1" = "sampe" ] || [ "$RunTypeCustom2" = "sampe" ];then
  #Adapt for making custom 2 txt file if its been selected
  if [ "$RunTypeCustom2" = "sampe" ];then
    CustomFlags="Custom2Flags.txt"
  else
    CustomFlags="Custom1Flags.txt"
  fi

  #create temp file containing all flags
  #Add the - Flag in front of each text in each column only if column contents doesn't equal -, will
  need some kind of loop
  FlagSetting=9
  FlagLetters="a|o|P|n|N|r"
  Letter=1
  for i in $(seq 6);do
    #I specify line number as each setting menu adds to this temp file but on a different line
    FlagNumber=$(awk -F '|' -v x=$FlagSetting 'NR==1 {print $x}'
~/GGOSS/tmp/BWA_SettingsChange.txt)
    Flag=$(echo "$FlagLetters" | awk -F '|' -v y=$Letter '{print $y}')
    echo "FlagNumber: $FlagNumber"
    if [ "$FlagNumber" = "Yes" ];then
      echo -n "-${Flag} " >> ~/GGOSS/tmp/${CustomFlags}
    else
      if [[ "$FlagNumber" > 0 ]] && [ "$FlagNumber" != "No" ];then
        echo -n "-${Flag} $FlagNumber " >> ~/GGOSS/tmp/${CustomFlags}
      fi
    fi
    Letter=$(( $Letter + 1 ))
    FlagSetting=$(( $FlagSetting + 1 ))
  done

fi

#####
## ----- bwasw
#####

if [ "$RunTypeCustom1" = "bwasw" ] || [ "$RunTypeCustom2" = "bwasw" ];then
  #Adapt for making custom 2 txt file if its been selected
  if [ "$RunTypeCustom2" = "bwasw" ];then
    CustomFlags="Custom2Flags.txt"
  else
    CustomFlags="Custom1Flags.txt"
  fi

  FlagSetting=15
  FlagLetters="a|b|q|r|t|w|T|c|z|s|N"
  Letter=1
  for i in $(seq 11);do
    #I specify line number as each setting menu adds to this temp file but on a different line
    FlagNumber=$(awk -F '|' -v x=$FlagSetting 'NR==1 {print $x}'
~/GGOSS/tmp/BWA_SettingsChange.txt)
    Flag=$(echo "$FlagLetters" | awk -F '|' -v y=$Letter '{print $y}')
    echo "FlagNumber: $FlagNumber"
    if [ "$FlagNumber" = "Yes" ];then

```

```

echo -n "-${Flag} " >> ~/GGOSS/tmp/${CustomFlags}
else
  if [[ "$FlagNumber" > 0 ]] && [ "$FlagNumber" != "No" ];then
    echo -n "-${Flag} $FlagNumber " >> ~/GGOSS/tmp/${CustomFlags}
  fi
fi
Letter=$(( $Letter + 1 ))
FlagSetting=$(( $FlagSetting + 1 ))
done

fi

#####
## ----- mem
#####

if [ "$RunTypeCustom1" = "mem" ] || [ "$RunTypeCustom2" = "mem" ];then
echo "1"
#Adapt for making custom 2 txt file if its been selected
if [ "$RunTypeCustom2" = "mem" ];then
echo "2a"
CustomFlags="Custom2Flags.txt"
else
echo "2b"
CustomFlags="Custom1Flags.txt"
fi

#create temp file containing all flags
#Add the - Flag in front of each text in each column only if column contents doesn't equal -, will
need some kind of loop
FlagSetting=2
FlagLetters="t|k|w|d|r|c|P|A|B|O|E|L|U|p|R|T|a|C|H|M|v"
Letter=1
#line check only applicable to mem and aln as they have seperate setting menus
LineSettingOnCheck=$(awk -F '|' 'NR==2 {print $1}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
if [ "$LineSettingOnCheck" = "mem" ];then
LineSettingOn=2
else
LineSettingOn=3
fi

for i in $(seq 21);do
#I specify line number as each setting menu adds to this temp file but on a different line
FlagNumber=$(awk -F '|' -v x=$FlagSetting -v a=$LineSettingOn 'NR==a {print $x}'
~/GGOSS/tmp/BWA_SettingsChange.txt)
Flag=$(echo "$FlagLetters" | awk -F '|' -v y=$Letter '{print $y}')
echo "FlagNumber: $FlagNumber"
if [ "$FlagNumber" = "Yes" ];then
echo "3"
echo -n "-${Flag} " >> ~/GGOSS/tmp/${CustomFlags}
else
  if [[ "$FlagNumber" > 0 ]] && [ "$FlagNumber" != "No" ];then
    echo "4"
    echo -n "-${Flag} $FlagNumber " >> ~/GGOSS/tmp/${CustomFlags}
  fi
fi

```

```

        fi
        Letter=$(( $Letter + 1 ))
        FlagSetting=$(( $FlagSetting + 1 ))
done

fi

#####
## ----- aln
#####

if [ "$RunTypeCustom1" = "aln" ] || [ "$RunTypeCustom2" = "aln" ];then
#Adapt for making custom 2 txt file if its been selected
if [ "$RunTypeCustom2" = "aln" ];then
CustomFlags="Custom2Flags.txt"
else
CustomFlags="Custom1Flags.txt"
fi

#create temp file containing all flags
#Add the - Flag in front of each text in each column only if column contents doesn't equal -, will
need some kind of loop
FlagSetting=2
FlagLetters="n|o|e|d|i|l|k|t|M|O|E|R|c|q|I|B|b|0|1|2"
Letter=1
#line check only applicable to mem and aln as they have seperate setting menus
LineSettingOnCheck=$(awk -F '|' 'NR==2 {print $1}' ~/GGOSS/tmp/BWA_SettingsChange.txt)
if [ "$LineSettingOnCheck" = "aln" ];then
LineSettingOn=2
else
LineSettingOn=3
fi

for i in $(seq 20);do
#I specify line number as each setting menu adds to this temp file but on a different line
FlagNumber=$(awk -F '|' -v x=$FlagSetting -v a=$LineSettingOn 'NR==a {print $x}'
~/GGOSS/tmp/BWA_SettingsChange.txt)
Flag=$(echo "$FlagLetters" | awk -F '|' -v y=$Letter '{print $y}')
echo "FlagNumber: $FlagNumber"
if [ "$FlagNumber" = "Yes" ];then
echo "3"
echo -n "-${Flag} " >> ~/GGOSS/tmp/${CustomFlags}
else
if [[ "$FlagNumber" > 0 ]] && [ "$FlagNumber" != "No" ];then
echo "4"
echo -n "-${Flag} $FlagNumber " >> ~/GGOSS/tmp/${CustomFlags}
fi
fi
Letter=$(( $Letter + 1 ))
FlagSetting=$(( $FlagSetting + 1 ))
done

fi

#at end make the text file line a variable

```

```

Custom1Flags=$(cat ~/GGOSS/tmp/Custom1Flags.txt)
Custom2Flags=$(cat ~/GGOSS/tmp/Custom2Flags.txt)

if [ "$PairedOrSingle" = "Paired" ];then
NumberOfFiles=$( echo "$NumberOfFiles / 2" | bc )
fi

line=1
for i in $(seq $NumberOfFiles);do

line2=$(( $line + 1 ))
FileToRun_orRead1=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')
FileToRunRead2=$(echo "$AlphabetSortedFiles" | awk -v x=$line2 'NR==x {print}')

filename=$(echo "$FileToRun_orRead1" | awk -F '_' '{NF-=4; OFS="_"; print}')

if [ "$RunTypeCustom1" != "Not Applicable" ];then
    if [ "$RunTypeCustom1" = "mem" ];then
        if [ "$PairedOrSingle" = "Single" ];then
            bwa mem $Custom1Flags $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln-se.sam
        else
            bwa mem $Custom1Flags $RefGenome ${PathToFiles}/${FileToRun_orRead1}
${PathToFiles}/${FileToRunRead2} > ~/GGOSS_InputOutput/BWA/${filename}_aln-pe.sam
        fi
    fi

    if [ "$RunTypeCustom1" = "aln" ];then
        if [ "$PairedOrSingle" = "Single" ];then
            bwa aln $Custom1Flags $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa1.sai
        else
            bwa aln $Custom1Flags $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa1.sai
            bwa aln $Custom1Flags $RefGenome ${PathToFiles}/${FileToRunRead2} >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa2.sai
        fi
    fi

    if [ "$RunTypeCustom1" = "samse" ];then
        echo "samse requires a '.sai' file input alongside the short read fastq, so cannot be run as a run
type for custom 1, please use aln as custom one and samse as custom 2 if this is what you want to do"
    fi

    if [ "$RunTypeCustom1" = "sampe" ];then
        echo "samse requires a '.sai' file input alongside the short read fastq, so cannot be run as a run
type for custom 1, please use aln as custom one and samse as custom 2 if this is what you want to do"
    fi

    if [ "$RunTypeCustom1" = "bwasw" ];then
        if [ "$PairedOrSingle" = "Single" ];then
            bwa bwasw $Custom1Flags $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln.sam
        else
            echo "bwsaw does not run on two paired end files simultaneously"
        fi
    fi
done

```

```

    fi
  fi
fi

#this is built so that custom 2 can be an additional individual run. If a tool selected doesn't use
previous output for input then it assumes the user wants to run this on the raw file also
if [ "$RunTypeCustom2" != "Not Applicable" ];then
  if [ "$RunTypeCustom2" = "mem" ];then
    if [ "$PairedOrSingle" = "Single" ];then
      bwa mem $Custom2Flags $RefGenome ${PathToFiles}/$FileToRun_orRead1 >
~/GGOSS_InputOutput/BWA/${filename}_aln-se.sam
    else
      bwa mem $Custom2Flags $RefGenome ${PathToFiles}/$FileToRun_orRead1
${PathToFiles}/$FileToRunRead2 > ~/GGOSS_InputOutput/BWA/${filename}_aln-pe.sam
    fi
  fi

  if [ "$RunTypeCustom2" = "aln" ];then
    if [ "$PairedOrSingle" = "Single" ];then
      bwa aln $Custom2Flags $RefGenome ${PathToFiles}/$FileToRun_orRead1 >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa.sai
    else
      bwa aln $Custom2Flags $RefGenome ${PathToFiles}/$FileToRun_orRead1 >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa1.sai
      bwa aln $Custom2Flags $RefGenome ${PathToFiles}/$FileToRunRead2 >
~/GGOSS_InputOutput/BWA/${filename}_aln_sa2.sai
    fi
  fi

  if [ "$RunTypeCustom2" = "samse" ];then
    if [ "$PairedOrSingle" = "Single" ];then
      bwa aln $Custom2Flags $RefGenome ~/GGOSS_InputOutput/BWA/${filename}_aln_sa.sai
${PathToFiles}/$FileToRun_orRead1 > ~/GGOSS_InputOutput/BWA/${filename}_aln-se.sam
    else
      echo "samse does not run on two paired end files simultaneously as sampe is built for that"
    fi
  fi

  if [ "$RunTypeCustom2" = "sampe" ];then
    if [ "$PairedOrSingle" = "Paired" ];then
      bwa aln $Custom2Flags $RefGenome ~/GGOSS_InputOutput/BWA/${filename}_aln_sa1.sai
~/GGOSS_InputOutput/BWA/${filename}_aln_sa2.sai ${PathToFiles}/$FileToRun_orRead1
${PathToFiles}/$FileToRunRead2 > ~/GGOSS_InputOutput/BWA/${filename}_aln-pe.sam
    else
      echo "samse does not run on single end files as samse is built for that"
    fi
  fi

  if [ "$RunTypeCustom2" = "bwasw" ];then
    if [ "$PairedOrSingle" = "Single" ];then
      bwa bwasw $Custom2Flags $RefGenome ${PathToFiles}/$FileToRun_orRead1 >
~/GGOSS_InputOutput/BWA/${filename}_aln.sam
    else
      echo "bwsaw does not run on two paired end files simultaneously"
    fi
  fi

```

```

fi
fi

line=$(( $line + 1 ))
done

else
#runs 1 of the defaults

RunType=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/BWA_SettingsChange.txt | awk -F '.' '{print $1}')

#adapt number of loops based on whether paired or not
if [ "$RunType" = "C" ] || [ "$RunType" = "D" ];then
NumberOfFiles=$( echo "$NumberOfFiles / 2" | bc )
fi

line=1
for i in $(seq 1 "$NumberOfFiles");do

line2=$(( $line + 1 ))
FileToRun_orRead1=$(echo "$AlphabetSortedFiles" | awk -v x=$line 'NR==x {print}')
FileToRunRead2=$(echo "$AlphabetSortedFiles" | awk -v x=$line2 'NR==x {print}')

filename=$(echo "$FileToRun_orRead1" | awk -F '_' '{NF-=4; OFS="_"; print}')

#is the file selected still there
if [ -f ${PathToFiles}/${FileToRun_orRead1} ]
then

Time2min=$(date +%M)
Time2hourtmp=$(date +%H)

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$NumberOfFiles - $line" | bc )
EstimatedtimetoFinishMins=$( echo "scale=2; ($TimeTaken / $line) * $NumberFilesLeftToDo" |
bc )
AverageTimeTakenPerSample=$( echo "scale=2; ($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "scale=2; $EstimatedtimetoFinishMins / 60" | bc )

PercentWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfFiles" | bc )
PercentCompleteBySample=$( echo "scale=2; $PercentWorthOfEachFile * $line" | bc )
PercentWorthOfAdapterRun=$( echo "scale=2; ($PercentWorthOfEachFile / 50) *
$AdapChange" | bc )
PercentComplete=$( echo "scale=2; (($PercentWorthOfEachFile * $line) -
$PercentWorthOfEachFile) + $PercentWorthOfAdapterRun" | bc )

Info=$(echo "Start time:${Time1hourtmp}:${Time1min}
Running sample: $FileToTrim ($line of $NumberOfFiles)

```

Time elapsed: \${TimeTaken} minutes
 Estimated time remaining: \$EstimatedtimetoFinishMins minutes (\${EstimatedtimetoFinishHours} hours)
 Average time taken per sample: \$AverageTimeTakenPerSample

Please check the BWA Logfile to confirm the success of the run on your samples. You can also find other information there, including the version of BWA that was used")

```

echo "#Running BWA... file: $filename    ${PercentComplete}% complete"
echo ${PercentComplete}

#if set for all possible run types, within each if I need to set the SAMtools bit
if [ "$RunType" = "A" ];then
  bwa mem $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln.sam
fi

  if [ "$RunType" = "B" ];then
    bwa aln $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_reads.sai; bwa samse $RefGenome
~/GGOSS_InputOutput/BWA/${filename}_reads.sai ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln-se.sam
  fi

  if [ "$RunType" = "C" ];then
    bwa mem $RefGenome ${PathToFiles}/${FileToRun_orRead1}
${PathToFiles}/${FileToRunRead2} > ~/GGOSS_InputOutput/BWA/${filename}_aln-pe.sam
  fi

  if [ "$RunType" = "D" ];then
    bwa aln $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_read1.sai; bwa aln $RefGenome
${PathToFiles}/${FileToRunRead2} > ~/GGOSS_InputOutput/BWA/${filename}_read2.sai
    bwa sampe $RefGenome ~/GGOSS_InputOutput/BWA/${filename}_read1.sai
~/GGOSS_InputOutput/BWA/${filename}_read2.sai ${PathToFiles}/${FileToRun_orRead1}
${PathToFiles}/${FileToRunRead2} > ~/GGOSS_InputOutput/BWA/${filename}_aln-pe.sam
  fi

  if [ "$RunType" = "E" ];then
    bwa mem -x pacbio $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln.sam
  fi

  if [ "$RunType" = "F" ];then
    bwa mem -x ont2d $RefGenome ${PathToFiles}/${FileToRun_orRead1} >
~/GGOSS_InputOutput/BWA/${filename}_aln.sam
  fi

else
echo "Failed to find the selected file: $FileToTrim"

```

```

fi

echo "End Of File: $line"

line=$(( $line + 1 ))

done
fi

pkill yad

} | tee ~/GGOSS/LogFiles/BWA_LOGFILE.txt | yad --progress --auto-kill --center --width=700 --
image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- BWA
GGOSS created by Giles Holt" --text="Running BWA
$Info"

#go back to main menu
yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- BWA" --
button="Open file location":5 --button="Return to menu":4 --text="

                BWA Complete

"

mode="$?"
case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/AdapterTrimmedFiles & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.3.2 GGOSS script for BLAST

```

#!/bin/bash

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                BLAST run from GGOSS Genomics program"
"
echo |(date +"                BLAST Start Date: %d-%m-%y Time: %T"
"
)

echo ""

yad --title="GENOME SEQUENCING PROGRAM -- BLAST                Created by Giles Holt"
--timeout=10 --no-buttons --no-escape --text="Please check the BLAST Logfile to ascertain and
confirm the success of BLAST analysis of your samples. You can

```

also find other information there, including the version of blast that was used

```
" &

echo 0
echo "#Starting BLAST prep"

BlastType=$(awk -F '|' 'NR==1 {print $1}' ~/GGOSS/tmp/BLASTSettingsRunType.txt)

#Paired-end or single-end

PairedOrSingle_End=$(awk -F '|' 'NR==1 {print $2}' ~/GGOSS/tmp/BLASTSettingsRunType.txt)

#File format

FileFormat=$(awk -F '|' 'NR==1 {print $3}' ~/GGOSS/tmp/BLASTSettingsRunType.txt)

echo 0
echo "#Preparing files... BLAST type: $BlastType| Paired Or Single End:$PairedOrSingle_End| File
format:$FileFormat"

#Path to files selected
PathToFiles=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt | sed 's/~//')

echo "PathToFiles: ${HOME}$PathToFiles"
FileList=$(tail -n +2 ~/GGOSS/tmp/Path2selectedFile.txt | sort)

if [ -f ~/GGOSS/tmp/BLAST_SingleFileNames.txt ];then
rm ~/GGOSS/tmp/BLAST_SingleFileNames.txt
fi

#if paired or single
if [[ "$PairedOrSingle_End" = "Paired" ]];then
echo "Paired files"
#loop for number of files divided by 2
NumberOfFilesPrep=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Path2selectedFile.txt)
NumberOfFiles=$(echo "($NumberOfFilesPrep - 1) / 2" | bc)
#calc percentage increase per sample
PercentToAddForEachSampleCompletion=$(echo "($NumberOfFiles * 2) / 100" | bc)

Percent=0
SampleNumber=1
File1Line=1
File2Line=2
for i in $(seq $NumberOfFiles);do
echo $Percent
echo "#Preparing files... ${Percent}% complete| Sample:${SampleNumber} of ${NumberOfFiles}|
BLAST type: ${BlastType}| Paired Or Single End:${PairedOrSingle_End}| File
format:${FileFormat}"
#Sort file pair
File1=$(echo "$FileList" | awk -v x=$File1Line 'NR==x {print}')
File2=$(echo "$FileList" | awk -v y=$File2Line 'NR==y {print}')
File1Name=$(echo $File1 | awk -F '.' '{print $1}')
File2Name=$(echo $File2 | awk -F '.' '{print $1}')
#change both to fasta if necessary
```

```

    if [[ "$FileFormat" = "FASTQ" ]];then
        seqtk seq -a "${HOME}${PathToFiles}/${File1}" >
"${HOME}${PathToFiles}/${File1Name}.fasta"
        seqtk seq -a "${HOME}${PathToFiles}/${File2}" >
"${HOME}${PathToFiles}/${File2Name}.fasta"
        File1="${File1Name}.fasta"
        File2="${File2Name}.fasta"
    else
        echo "$FileList" > ~/GGOSS/tmp/BLAST_SingleFileNames.txt
    fi
    #merge into single file
    SingleFileName=$(echo "$File1Name" | awk -F '_' '{NF-=4; OFS="_"; print}')
    #make new file list to grab names from
    echo "$SingleFileName" >> ~/GGOSS/tmp/BLAST_SingleFileNames.txt

    cat ${HOME}${PathToFiles}/${File1} ${HOME}${PathToFiles}/${File2} >
${HOME}${PathToFiles}/${SingleFileName}.fasta

    #adapt percentage
    Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)

    echo "Complete preparation for file: ${SingleFileName}"
    File1Line=$(( $File1Line + 2 ))
    File2Line=$(( $File2Line + 2 ))
    SampleNumber=$(( $SampleNumber + 1 ))
done

NumberOfFilesForBlast=$NumberOfFiles

else

echo "Single files"
#loop for number of files
NumberOfFiles=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/Path2selectedFile.txt)

PercentToAddForEachSampleCompletion=$(echo "($NumberOfFiles * 2) / 100" | bc)
Percent=0
FileLine=1
for i in $(seq $NumberOfFiles);do
    echo $Percent
    echo "#Preparing files... ${Percent}% complete| Sample:${FileLine} of ${NumberOfFiles}|
BLAST type: ${BlastType}| Paired Or Single End:${PairedOrSingle_End}| File
format:${FileFormat}"
    #grab single file
    File=$(echo "$FileList" | awk -v x=$FileLine 'NR==x {print}')
    FileName=$(echo $File1 | awk -F '.' '{print $1}')
    #Change to fasta if necessary
    if [[ "$FileFormat" = "FASTQ" ]];then
        seqtk seq -a ${HOME}${PathToFiles}/${File} > ${HOME}${PathToFiles}/${FileName}.fasta
    else
        echo "$FileList" > ~/GGOSS/tmp/BLAST_SingleFileNames.txt
    fi
#    mv ${HOME}${PathToFiles}/${FileName}.fasta ~/ncbi-blast-2.3.0+/db/${FileName}.fasta
    echo "${FileName}.fasta" >> ~/GGOSS/tmp/BLAST_SingleFileNames.txt
    #adapt percentage

```

```

Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileLine=$(( $FileLine + 1 ))

echo "Complete preparation for file: ${FileName}"
done
NumberOfFilesForBlast=$NumberOfFiles
fi

echo "File preparation complete"

if [[ "$BlastType" = "blastn" ]];then
echo "Running blastn"

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
echo "#Running blastn... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')
echo "Running blastn on file: $InputFile"
#the beginning is the same as these are the base blast settings, after that are the unique blastn settings
blastn -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evalute subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsps
html gi_list negativegelist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote word_size gapopen gapextend reward
penalty strand dust filtering_db window_masker_taxid window_masker_db soft_masking
lcase_masking db_soft_mask db_hard_mask perc_identity xdrop_ungap xdrop_gap xdrop_gap_final
min_raw_gapped_score ungapped outfmt -out ~/GGOSS_InputOutput/Blast/${InputFile}_Blastn.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

fi

if [ "$BlastType" = "blastp" ];then
echo "Running blastp"

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
echo "#Running blastp... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')
echo "Running blastp on file: $InputFile"
#the beginning is the same as these are the base blast settings, after that are the unique blastn settings

```

```

blastp -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evalute subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsps
html gi_list negativegelist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote outfmt -out
~/GGOSS_InputOutput/Blast/${InputFile}_Blastp.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

```

```
fi
```

```

if [ "$BlastType" = "blastx" ];then
echo "Running blastx"

```

```

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
    echo "#Running blastx... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

```

```

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')
echo "Running blastx on file: $InputFile"
#the beginning is the same as these are the base blast settings, after that are the unique blastn settings
blastx -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evalute subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsps
html gi_list negativegelist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote outfmt -out
~/GGOSS_InputOutput/Blast/${InputFile}_Blastx.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

```

```
fi
```

```

if [ "$BlastType" = "tblastn" ];then
echo "Running tblastn"

```

```

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
    echo "#Running tblastn... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

```

```

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')
echo "Running tblastn on file: $InputFile"
#the beginning is the same as these are the base blast settings, after that are the unique blastn settings

```

```

tblastn -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evaluate subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsps
html gi_list negativegelist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote outfmt -out
~/GGOSS_InputOutput/Blast/${InputFile}_tblastn.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

```

```
fi
```

```

if [ "$BlastType" = "tblastx" ];then
echo "Running tblastx"

```

```

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
    echo "#Running tblastx... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

```

```

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')

```

```
echo "Running tblastx on file: $InputFile"
```

```

#the beginning is the same as these are the base blast settings, after that are the unique blastn settings
tblastx -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evaluate subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsps
html gi_list negativegelist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote word_size matrix threshold seg
soft_masking lcase_masking db_soft_mask db_hard_mask strand query_genetic_code db_gen_code
max_intron_length outfmt -out ~/GGOSS_InputOutput/Blast/${InputFile}_tblastx.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

```

```
fi
```

```

if [ "$BlastType" = "rpsblast" ];then
echo "Running rpsblast"

```

```

#loop to number of files
FileToRunThroughBlast=1
for i in $(seq $NumberOfFilesForBlast);do
echo $Percent
    echo "#Running rpsblast... ${Percent}% complete| Sample:${FileToRunThroughBlast} of
${NumberOfFilesForBlast}"
#the settings editing script will edit this script, this scripts just loops for sample numbers, runs the
blast type, and prepares the files

```

```

InputFile=$(cat ~/GGOSS/tmp/BLAST_SingleFileNames.txt | awk -v x=$FileToRunThroughBlast
'NR==x {print}')

```

```
echo "Running rpsblast on file: $InputFile"
```

```
#the beginning is the same as these are the base blast settings, after that are the unique blastn settings
```

```

rpsblast -query ${HOME}${PathToFiles}/${InputFile}.fasta -db PathToDataBasedb/DATABASE
queryloc evaluate subject subjectloc showgis numdescriptions numalignments maxtargetseqs maxhsp
html gi_list negativeglist entrezquery cullinglimit besthitoverhang besthitscoreedge dbsize searchsp
importsearchstrategy exportsearchstrategy parsedeflines remote outfmt -out
~/GGOSS_InputOutput/Blast/${InputFile}_rpsBlast.out
Percent=$(echo "scale=2; $Percent + $PercentToAddForEachSampleCompletion" | bc)
FileToRunThroughBlast=$(( $FileToRunThroughBlast + 1 ))
done

```

```
fi
```

```

echo | (date +"
                BLAST Finish Date: %d-%m-%y Time: %T
"
)
echo 100
echo "#BLAST complete... 100% complete"
} | tee ~/GGOSS/LogFiles/BLAST_LOGFILE.txt | yad --progress --auto-kill --center --width=700 --
image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- BLAST
GGOSS created by Giles Holt" --text="Running BLAST" --button="Continue"

```

10.9.3.3.3 GGOSS scripts for PRICE

10.9.3.3.3.1 PRICE template

```
#!/bin/bash
```

```

PathToPRICE=$(awk -F '|' '{print $6}' ~/GGOSS/tmp/PathToTools_SettingsChange.txt)
if [ "$PathToPRICE" = "N/A" ];then
PathToPRICE="$HOME/"
fi

```

```

${PathToPRICE}PriceTI INPUT_FILE_POSS_1_START: InputReadType FilePath/FileName1
FilePath/FileName2 InputReadType_Set1 InputReadType_Set2 InputReadType_Set3
InputReadType_Set4 InputReadType_Set5 InputReadType_Set6 FilterType1_rqf rqfnumber_a
rqfnumber_b rqfnumber_c rqfnumber_d FilterType2_rnf rnfnumber_a rnfnumber_b rnfnumber_c
FilterType3_MaxHP MaxHPnumber_a FilterType4_MaxDi MaxDiNumber_a FilterType5_badf
badfNumber_b badfNumber_a FilterType6_repmask repmaskNumber_a repmaskNumber_b
repmaskNumber_c repmaskNumber_d repmaskNumber_e repmaskNumber_f repmaskNumber_g
FilterType7_reset resetNumber_a FILTER1_INITIAL_CONTIGS_START: Filter1ICflag_icbf
Filter1IC_icbf_a Filter1IC_icbf_b Filter1IC_icbf_c Filter1ICflag_icmHp Filter1IC_icmHp_a
Filter1IC_icmHp_b Filter1ICflag_icmDi Filter1IC_icmDi_a Filter1IC_icmDi_b Filter1ICflag_icqf
Filter1IC_icqf_a Filter1IC_icqf_b Filter1IC_icqf_c Filter1ICflag_icnf Filter1IC_icnf_a
Filter1IC_icnf_b :FILTER1_INITIAL_CONTIGS_END INPUT_FILE_POSS_2_START:
Input2ReadType File2Path/File2Name1 File2Path/File2Name2 Input2ReadType_Set1
Input2ReadType_Set2 Input2ReadType_Set3 Input2ReadType_Set4 Input2ReadType_Set5
Input2ReadType_Set6 Filter2Type1_rqf rqf2number_a rqf2number_b rqf2number_c rqf2number_d
Filter2Type2_rnf rnf2number_a rnf2number_b rnf2number_c Filter2Type3_MaxHP
MaxHP2number_a Filter2Type4_MaxDi MaxDi2Number_a Filter2Type5_badf badf2Number_b
badf2Number_a Filter2Type6_repmask repmask2Number_a repmask2Number_b repmask2Number_c
repmask2Number_d repmask2Number_e repmask2Number_f repmask2Number_g
Filter2Type7_reset reset2Number_a FILTER2_INITIAL_CONTIGS_START: Filter2ICflag_icbf

```

```

Filter2IC_icbf_a Filter2IC_icbf_b Filter2IC_icbf_c Filter2ICFlag_icmHp Filter2IC_icmHp_a
Filter2IC_icmHp_b Filter2ICFlag_icmDi Filter2IC_icmDi_a Filter2IC_icmDi_b Filter2ICFlag_icqf
Filter2IC_icqf_a Filter2IC_icqf_b Filter2IC_icqf_c Filter2ICFlag_icnf Filter2IC_icnf_a
Filter2IC_icnf_b :FILTER2_INITIAL_CONTIGS_END INPUT_FILE_POSS_3_START:
Input3ReadType File3Path/File3Name1 File3Path/File3Name2 Input3ReadType_Set1
Input3ReadType_Set2 Input3ReadType_Set3 Input3ReadType_Set4 Input3ReadType_Set5
Input3ReadType_Set6 Filter3Type1_rqf rqf3number_a rqf3number_b rqf3number_c rqf3number_d
Filter3Type2_rnf rnf3number_a rnf3number_b rnf3number_c Filter3Type3_MaxHP
MaxHP3number_a Filter3Type4_MaxDi MaxDi3Number_a Filter3Type5_badf badf3Number_b
badf3Number_a Filter3Type6_repmask repmask3Number_a repmask3Number_b repmask3Number_c
repmask3Number_d repmask3Number_e repmask3Number_f repmask3Number_g
Filter3Type7_reset reset3Number_a FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf
Filter3IC_icbf_a Filter3IC_icbf_b Filter3IC_icbf_c Filter3ICFlag_icmHp Filter3IC_icmHp_a
Filter3IC_icmHp_b Filter3ICFlag_icmDi Filter3IC_icmDi_a Filter3IC_icmDi_b Filter3ICFlag_icqf
Filter3IC_icqf_a Filter3IC_icqf_b Filter3IC_icqf_c Filter3ICFlag_icnf Filter3IC_icnf_a
Filter3IC_icnf_b :FILTER3_INITIAL_CONTIGS_END OtherParamType1_nc
OtherParamType2_link OtherParamType3_mol OtherParamType4_tol OtherParamType5_mpi
OtherParamType6_tpi OtherParamType7_dbmax OtherParamType8_dbk OtherParamType9_dbms
OtherParamType10_r OtherParamType11_q OtherParamType12_G OtherParamType13_E
FILTERING_PROCESSING_ASSEMBLED_CONTIGS_START: lenfflag lenfnumber_a
lenfnumber_b trimflag trimnumber_a trimnumber_b trimnumber_c trimBflag trimBnumber_a
trimBnumber_b trimBnumber_c trimIflag trimInumber_a trimInumber_b targetflag targetnumber_a
targetnumber_b targetnumber_c targetFflag targetFnumber_a targetFnumber_b targetFnumber_c
:FILTERING_PROCESSING_ASSEMBLED_CONTIGS_END NumberOfThreads
MaxThreadsPerFile USER_INTERFACE_log -nco -o PriceTI_Filename.OutputType -nco -o
PriceTI_Filename.Output2Type

```

10.9.3.3.2 GGOSS run script for PRICE

```

#!/bin/bash

#If in additional input 1 an assembly file selected, ask if they want to use contig or scaffold
InputFileType=$(awk -F '[' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" != "Not Applicable" ];then
    FileLocationOrFileName=$(awk 'NR==1 {print}'
~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt)
    if [ "$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes" ] || [
"$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet" ] || [ "$FileLocationOrFileName" =
~/GGOSS_InputOutput/IDBA" ];then

yad --title="GENOME SEQUENCING PROGRAM - PRICETI" created by Giles
Holt" --text="2nd input file type selected ($InputFileType). The type of file selected for this are
assembly files. Which File type do you wish to use? " --center --size=fit --form \
--field="Assembled file type to use":CB \
'contigs.fasta!scaffolds.fasta!' \
--text-info --show-uri --height=100 width=500 --center --wrap \
--button="gtk-save:0" --editable --filename=~/GGOSS/tmp/PRICETI_ContigOrScaffold1.txt >
~/GGOSS/tmp/PRICETI_ContigOrScaffold1.txt

fi
fi

```

```
#If in additional input 1 an assembly file selected, ask if they want to use contig or scaffold
InputFileType=$(awk -F '[' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
if [ "$InputFileType" != "Not Applicable" ];then
  FileLocationOrFileName=$(awk 'NR==1 {print}'
~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt)
  if [ "$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes" ] || [
"$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet" ] || [ "$FileLocationOrFileName" =
~/GGOSS_InputOutput/IDBA" ] ;then
```

```
yad --title="GENOME SEQUENCING PROGRAM - PRICETI" created by Giles
Holt" --text="2nd input file type selected ($InputFileType). The type of file selected for this are
assembly files. Which File type do you wish to use? " --center --size=fit --form \
--field="Assembled file type to use":CB \
'contigs.fasta!scaffolds.fasta!' \
--text-info --show-uri --height=100 width=500 --center --wrap \
--button="gtk-save:0" --editable --filename=~ /GGOSS/tmp/PRICETI_ContigOrScaffold2.txt >
~/GGOSS/tmp/PRICETI_ContigOrScaffold2.txt
```

```
fi
fi
```

```
yad --title="GENOME SEQUENCING PROGRAM -- PRICETI" Created by Giles Holt" -
-no-buttons --no-escape --length=100 --width=400 --center --text="
```

Applying your PRICETI Settings" &

```
{
echo "          GGOSS Genomics - Created by Giles Holt
"
echo "          PRICETI run from GGOSS Genomics program
"
echo |(date +"          PRICETI Start Date: %d-%m-%y Time: %T
```

```
")
```

#for now additional input 1 or 2 wont function properly two read files - but i don't think that should matter, as from what i can tell you would never use those types of file for Second or third inputs (due to how I put in the filenames in the loop, but the templaterun script and everything else is built to manage it)

#Adapt script according to settings

```
cp ~/GGOSS/Scripts/PriceTiTemplate.sh ~/GGOSS/Scripts/PriceTiRun.sh
```

```
chmod 755 ~/GGOSS/Scripts/PriceTiRun.sh
```

```

if [ -f ~/GGOSS/tmp/YadWindow4SelectedSettings.txt ]
then
rm ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

#Create txt file containing all the relevant selected settings information
touch ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#####
#-----
#####--set Input file type according to settings---#####
#-----
#####

InputFileType=$(awk -F '[' '{print $1}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" = "Read Files" ];then
#add to txt file containing relevant selected settings information
echo "Input File Type Set:${InputFileType}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

sed -i -e "s/FILTER1_INITIAL_CONTIGS_START: Filter1ICFlag_icbf Filter1IC_icbf_a
Filter1IC_icbf_b Filter1IC_icbf_c Filter1ICFlag_icmHp Filter1IC_icmHp_a Filter1IC_icmHp_b
Filter1ICFlag_icmDi Filter1IC_icmDi_a Filter1IC_icmDi_b Filter1ICFlag_icqf Filter1IC_icqf_a
Filter1IC_icqf_b Filter1IC_icqf_c Filter1ICFlag_icnf Filter1IC_icnf_a Filter1IC_icnf_b
:FILTER1_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#####
#--- READ FILES
#####
#remove spare settings required for spf
sed -i -e "s/InputReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/InputReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[' '{print $4}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

#add to txt file containing relevant selected settings information
echo "Read File Type Set:${ReadFileType}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove second input for read types only using 1
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "msp" ];then
sed -i -e "s/FilePath/FileName2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

sed -i -e "s/INPUT_FILE_POSS_1_START: InputReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ]
then
InputFileSetting=$(awk -F '[' '{print $7}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[' '{print $11}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

```

```

if [ "$InputFileSetting" = "-" ];then
echo "Error! Amplicon insert size for $InputFileType $ReadFileType must be set"
notify-send "Error! Amplicon insert size for $InputFileType $ReadFileType must be set"
else

#add to txt file containing relevant selected settings information
echo "Amplicon insert size set:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/InputReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#set the second setting for Read File type
#fs,fp,ms,mp all have the same settings in the same order
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "mp" ];then

#uses paired end settings or mate pair settings for 'No. of cycles to be skipped before input file is
used:'
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fp" ];then
InputFileSetting=$(awk -F '[' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "No. of cycles to be skipped before input file is used: Not Set"
sed -i -e "s/InputReadType_Set2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "No. of cycles to be skipped before input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/InputReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#uses paired end settings or mate pair settings for 'No. of cycles for which input file is used:'
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fp" ];then

InputFileSetting=$(awk -F '[' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "No. of cycles for which input file is used: Not Set"
sed -i -e "s/InputReadType_Set3 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information

```

```

    echo "No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    #add the amplicons insert size info
    sed -i -e "s/InputReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

    #add the amplicons insert size info
    sed -i -e "s/InputReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
#if paired end of mate pair - adjusts settings accordingly
    if [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fpp" ];then

        InputFileSetting=$(awk -F '[' '{print $10}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
    else
        InputFileSetting=$(awk -F '[' '{print $14}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
    fi

    if [ "$InputFileSetting" = "-" ];then
        echo "Error! Required % identity for match $InputFileType $ReadFileType must be set"
        notify-send "Error! Required % identity for match $InputFileType $ReadFileType must be set"
    else
        #add to txt file containing relevant selected settings information
        echo "Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

        #add the amplicons insert size info
        sed -i -e "s/InputReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    fi

    if [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fpp" ];then

        InputFileSetting=$(awk -F '[' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
    else
        InputFileSetting=$(awk -F '[' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
    fi
    if [ "$InputFileSetting" = "-" ];then
        echo "No. of cycles to be skipped before input file is used: Not Set"
        sed -i -e "s/InputReadType_Set3 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
        #add to txt file containing relevant selected settings information
        echo "No. of cycles to be skipped before input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

        #add the amplicons insert size info
        sed -i -e "s/InputReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

    fi

#uses paired end settings or mate pair settings for 'No. of cycles for which input file is used:'
    if [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fpp" ];then

```

```

InputFileSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "No. of cycles for which input file is used: Not Set"
sed -i -e "s/InputReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/InputReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

fi

else

if [ "$InputFileType" = "False Paired-End Files" ];then

echo "Input File Type Set:${InputFileType}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove read filter stuff as hasn't been selected
sed -i -e "s/FilterType1_rqf rqfnumber_a rqfnumber_b rqfnumber_c rqfnumber_d FilterType2_rnf
rnfnumber_a rnfnumber_b rnfnumber_c FilterType3_MaxHP MaxHPnumber_a FilterType4_MaxDi
MaxDiNumber_a FilterType5_badf badfNumber_b badfNumber_a FilterType6_repmask
repmaskNumber_a repmaskNumber_b repmaskNumber_c repmaskNumber_d repmaskNumber_e
repmaskNumber_f repmaskNumber_g FilterType7_reset resetNumber_a \+//g"
~/GGOSS/Scripts/PriceTiRun.sh

#remove icf filter stuff as hasn't been selected
sed -i -e "s/FILTER1_INITIAL_CONTIGS_START: Filter1ICFlag_icbf Filter1IC_icbf_a
Filter1IC_icbf_b Filter1IC_icbf_c Filter1ICFlag_icmHp Filter1IC_icmHp_a Filter1IC_icmHp_b
Filter1ICFlag_icmDi Filter1IC_icmDi_a Filter1IC_icmDi_b Filter1ICFlag_icqf Filter1IC_icqf_a
Filter1IC_icqf_b Filter1IC_icqf_c Filter1ICFlag_icnf Filter1IC_icnf_a Filter1IC_icnf_b
:FILTER1_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#Remove additional fileinput not required for this input type
sed -i -e "s/FilePath/FileName2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

sed -i -e "s/INPUT_FILE_POSS_1_START: InputReadType\+/-/${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "Read File Type Set:${ReadFileType}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $15}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

echo "Length of 'reads' taken from each side of input reads:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $16}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Amplicon insert size:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

if [ "$ReadFileType" = "spf" ];then
InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles to be skipped before input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles for which input file is not used before being used again:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove the spare
sed -i -e "s/InputReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

InputFileSetting=$(awk -F '[]' '{print $20}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles to be skipped before input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set6\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles for which input file is not used before being used again:${InputFileSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

fi

#### icf #####

#####

else

if [ "$InputFileType" = "Initial Contig Files" ];then

#Remove additional fileinput not required for this input type
sed -i -e "s/FilePath/FileName2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

sed -i -e "s/INPUT_FILE_POSS_1_START: InputReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

echo "Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#####--- icf

if [ "$ReadFileType" = "icf" ] || [ "$ReadFileType" = "icfNt" ];then
#remove spare settings
sed -i -e "s/InputReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/InputReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/InputReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Const by which to multiply quality scores:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

else

#####--- picf

```

```

#remove spare settings
sed -i -e "s/InputReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/InputReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#switch input file and first setting
sed -i -e "s/FilePath/FileName1 InputReadType_Set1\+//g" ~/GGOSS/Scripts/PriceTiRun.sh
~/GGOSS/Scripts/PriceTiRun.sh

InputFileSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set1\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of initial contigs from this file:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set2\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set3\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/InputReadType_Set4\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Const by which to multiply quality scores:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

fi

fi

#adapt the filter section, as the filter types available for different input types

#####-----#####

#####-- Secondary input file type

#####-----#####

if [ "$InputFileType" = "Not Applicable" ];then
sed -i -e "s/FILTER2_INITIAL_CONTIGS_START: Filter2ICFlag_icbf Filter2IC_icbf_a
Filter2IC_icbf_b Filter2IC_icbf_c Filter2ICFlag_icmHp Filter2IC_icmHp_a Filter2IC_icmHp_b
Filter2ICFlag_icmDi Filter2IC_icmDi_a Filter2IC_icmDi_b Filter2ICFlag_icqf Filter2IC_icqf_a
Filter2IC_icqf_b Filter2IC_icqf_c Filter2ICFlag_icnf Filter2IC_icnf_a Filter2IC_icnf_b
:FILTER2_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#remove read filter stuff as hasn't been selected

```

```

sed -i -e "s/Filter2Type1_rqf rqf2number_a rqf2number_b rqf2number_c rqf2number_d
Filter2Type2_rnf rnf2number_a rnf2number_b rnf2number_c Filter2Type3_MaxHP
MaxHP2number_a Filter2Type4_MaxDi MaxDi2Number_a Filter2Type5_badf badf2Number_b
badf2Number_a Filter2Type6_repmask repmask2Number_a repmask2Number_b repmask2Number_c
repmask2Number_d repmask2Number_e repmask2Number_f repmask2Number_g
Filter2Type7_reset reset2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

#remove input stuff as hasn't been selected

```

```

sed -i -e "s/INPUT_FILE_POSS_2_START: Input2ReadType File2Path/File2Name1
File2Path/File2Name2 Input2ReadType_Set1 Input2ReadType_Set2 Input2ReadType_Set3
Input2ReadType_Set4 Input2ReadType_Set5 Input2ReadType_Set6 \+//g"
~/GGOSS/Scripts/PriceTiRun.sh

```

```

else

```

```

InputFileType=$(awk -F '[' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

```

```

if [ "$InputFileType" = "Read Files" ];then

```

```

#add to txt file containing relevant selected settings information
echo "(1) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

#remove icf filter as not been chosen

```

```

sed -i -e "s/FILTER2_INITIAL_CONTIGS_START: Filter2ICFlag_icbf Filter2IC_icbf_a
Filter2IC_icbf_b Filter2IC_icbf_c Filter2ICFlag_icmHp Filter2IC_icmHp_a Filter2IC_icmHp_b
Filter2ICFlag_icmDi Filter2IC_icmDi_a Filter2IC_icmDi_b Filter2ICFlag_icqf Filter2IC_icqf_a
Filter2IC_icqf_b Filter2IC_icqf_c Filter2ICFlag_icnf Filter2IC_icnf_a Filter2IC_icnf_b
:FILTER2_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

#####

```

```

#--- READ FILES

```

```

#####

```

```

#remove spare settings required for spf

```

```

sed -i -e "s/Input2ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Input2ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

ReadFileType=$(awk -F '[' '{print $4}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

```

```

#add to txt file containing relevant selected settings information

```

```

echo "Additional Input File (1):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

#remove second input for read types only using 1

```

```

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "msp" ];then

```

```

sed -i -e "s/File2Path/File2Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

```

```

sed -i -e "s/INPUT_FILE_POSS_2_START: Input2ReadType\+/--${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

```

```

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

```

```

InputFileSetting=$(awk -F '[' '{print $7}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

```

```

else
InputFileSetting=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):Error! Amplicon insert size for $InputFileType $ReadFileType
must be set"
notify-send "Error! Amplicon insert size for $InputFileType $ReadFileType must be set"
else

#add to txt file containing relevant selected settings information
echo "Additional Input File (1):Amplicon insert size set:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#set the second setting for Read File type
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "mp" ];then

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):No. of cycles to be skipped before input file is used: Not Set"
sed -i -e "s/Input2ReadType_Set2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (1):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):No. of cycles for which input file is used: Not Set"

```

```

sed -i -e "s/Input2ReadType_Set3 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (1):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[' '{print $10}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[' '{print $14}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):Error! Required % identity for match $InputFileType
$ReadFileType must be set"
notify-send "Error! Required % identity for match $InputFileType $ReadFileType must be set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (1):Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi
if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):No. of cycles to be skipped before input file is used: Not Set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (1):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (1):No. of cycles for which input file is used: Not Set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (1):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input2ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

fi

else

if [ "$InputFileType" = "False Paired-End Files" ];then

echo "(1) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove icf filter as not been chosen
sed -i -e "s/FILTER2_INITIAL_CONTIGS_START: Filter2ICFlag_icbf Filter2IC_icbf_a
Filter2IC_icbf_b Filter2IC_icbf_c Filter2ICFlag_icmHp Filter2IC_icmHp_a Filter2IC_icmHp_b
Filter2ICFlag_icmDi Filter2IC_icmDi_a Filter2IC_icmDi_b Filter2ICFlag_icqf Filter2IC_icqf_a
Filter2IC_icqf_b Filter2IC_icqf_c Filter2ICFlag_icnf Filter2IC_icnf_a Filter2IC_icnf_b
:FILTER2_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#Remove additional fileinput not required for this input type
sed -i -e "s/File2Path/File2Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

sed -i -e "s/INPUT_FILE_POSS_2_START: Input2ReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $15}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Length of 'reads' taken from each side of input
reads:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

InputFileSetting=$(awk -F '[]' '{print $16}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Amplicon insert size:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

if [ "$ReadFileType" = "spf" ];then
InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles for which input file is not used before being used
again:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove the spare
sed -i -e "s/Input2ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

InputFileSetting=$(awk -F '[]' '{print $20}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input File (1):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set6\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles for which input file is not used before being used
again:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

fi

```
#### icf #####
```

```
#####
```

else

```
if [ "$InputFileType" = "Initial Contig Files" ];then
```

```
#remove icf filter as not been chosen
```

```
sed -i -e "s/Filter2Type1_rqf rqf2number_a rqf2number_b rqf2number_c rqf2number_d
Filter2Type2_rnf rnf2number_a rnf2number_b rnf2number_c Filter2Type3_MaxHP
MaxHP2number_a Filter2Type4_MaxDi MaxDi2Number_a Filter2Type5_badf badf2Number_b
badf2Number_a Filter2Type6_repmask repmask2Number_a repmask2Number_b repmask2Number_c
repmask2Number_d repmask2Number_e repmask2Number_f repmask2Number_g
Filter2Type7_reset reset2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
#Remove additional fileinput not required for this input type
```

```
sed -i -e "s/File2Path/File2Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
ReadFileType=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/INPUT_FILE_POSS_2_START: Input2ReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "(1) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
echo "Additional Input File (1):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
#####--- icf
```

```
if [ "$ReadFileType" = "icf" ] || [ "$ReadFileType" = "icfNt" ];then
```

```
#remove spare settings
```

```
sed -i -e "s/Input2ReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
sed -i -e "s/Input2ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
sed -i -e "s/Input2ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/Input2ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input File (1):No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/Input2ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input File (1):No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```

sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Const by which to multiply quality scores:${InputFileSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

else

#####--- picf

#remove spare settings
sed -i -e "s/Input2ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Input2ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#switch input file and first setting
sed -i -e "s/File2Path\FileName1 Input2ReadType_Set1\+/Input2ReadType_Set1
File2Path\FileName1/g" ~/GGOSS/Scripts/PriceTiRun.sh

InputFileSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of initial contigs from this file:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input2ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (1):Const by which to multiply quality scores:${InputFileSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

fi

fi

fi

InputFileType=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

```

```

if [ "$InputFileType" = "Not Applicable" ];then
sed -i -e "s/FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf Filter3IC_icbf_a
Filter3IC_icbf_b Filter3IC_icbf_c Filter3ICFlag_icmHp Filter3IC_icmHp_a Filter3IC_icmHp_b
Filter3ICFlag_icmDi Filter3IC_icmDi_a Filter3IC_icmDi_b Filter3ICFlag_icqf Filter3IC_icqf_a
Filter3IC_icqf_b Filter3IC_icqf_c Filter3ICFlag_icnf Filter3IC_icnf_a Filter3IC_icnf_b
:FILTER3_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#remove read filter stuff as hasn't been selected
sed -i -e "s/Filter3Type1_rqf rqf3number_a rqf3number_b rqf3number_c rqf3number_d
Filter3Type2_rnf rnf3number_a rnf3number_b rnf3number_c Filter3Type3_MaxHP
MaxHP3number_a Filter3Type4_MaxDi MaxDi3Number_a Filter3Type5_badf badf3Number_b
badf3Number_a Filter3Type6_repmask repmask3Number_a repmask3Number_b repmask3Number_c
repmask3Number_d repmask3Number_e repmask3Number_f repmask3Number_g
Filter3Type7_reset reset3Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#remove input stuff as hasn't been selected
sed -i -e "s/INPUT_FILE_POSS_3_START: Input3ReadType File3Path/File3Name1
File3Path/File3Name2 Input3ReadType_Set1 Input3ReadType_Set2 Input3ReadType_Set3
Input3ReadType_Set4 Input3ReadType_Set5 Input3ReadType_Set6 \+//g"
~/GGOSS/Scripts/PriceTiRun.sh

else

if [ "$InputFileType" = "Read Files" ];then
#add to txt file containing relevant selected settings information
echo "(2) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

sed -i -e "s/FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf Filter3IC_icbf_a
Filter3IC_icbf_b Filter3IC_icbf_c Filter3ICFlag_icmHp Filter3IC_icmHp_a Filter3IC_icmHp_b
Filter3ICFlag_icmDi Filter3IC_icmDi_a Filter3IC_icmDi_b Filter3ICFlag_icqf Filter3IC_icqf_a
Filter3IC_icqf_b Filter3IC_icqf_c Filter3ICFlag_icnf Filter3IC_icnf_a Filter3IC_icnf_b
:FILTER3_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#####
#---- READ FILES
#####
#remove spare settings required for spf
sed -i -e "s/Input3ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Input3ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[' '{print $4}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

#add to txt file containing relevant selected settings information
echo "Additional Input File (2):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove second input for read types only using 1
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "msp" ];then
sed -i -e "s/File3Path/File3Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

```

```

sed -i -e "s/INPUT_FILE_POSS_3_START: Input3ReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ]
then
InputFileSetting=$(awk -F '[]' '{print $7}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):Error! Amplicon insert size for $InputFileType $ReadFileType
must be set"
notify-send "Error! Amplicon insert size for $InputFileType $ReadFileType must be set"
else

#add to txt file containing relevant selected settings information
echo "Additional Input File (2):Amplicon insert size set:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

#set the second setting for Read File type
if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "ms" ] || [
"$ReadFileType" = "mp" ];then

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):No. of cycles to be skipped before input file is used: Not Set"
sed -i -e "s/Input3ReadType_Set2\+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (2):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else

```

```

InputFileSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):No. of cycles for which input file is used: Not Set"
sed -i -e "s/Input3ReadType_Set3\+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (2):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $10}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $14}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):Error! Required % identity for match $InputFileType
$ReadFileType must be set"
notify-send "Error! Required % identity for match $InputFileType $ReadFileType must be set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (2):Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):No. of cycles to be skipped before input file is used: Not Set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (2):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

if [ "$ReadFileType" = "fs" ] || [ "$ReadFileType" = "fsp" ] || [ "$ReadFileType" = "fp" ] || [
"$ReadFileType" = "fpp" ];then

InputFileSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
else
InputFileSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
fi

if [ "$InputFileSetting" = "-" ];then
echo "Additional Input File (2):No. of cycles for which input file is used: Not Set"
else
#add to txt file containing relevant selected settings information
echo "Additional Input File (2):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#add the amplicons insert size info
sed -i -e "s/Input3ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

fi

else

if [ "$InputFileType" = "False Paired-End Files" ];then

echo "(2) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

sed -i -e "s/FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf Filter3IC_icbf_a
Filter3IC_icbf_b Filter3IC_icbf_c Filter3ICFlag_icmHp Filter3IC_icmHp_a Filter3IC_icmHp_b
Filter3ICFlag_icmDi Filter3IC_icmDi_a Filter3IC_icmDi_b Filter3ICFlag_icqf Filter3IC_icqf_a
Filter3IC_icqf_b Filter3IC_icqf_c Filter3ICFlag_icnf Filter3IC_icnf_a Filter3IC_icnf_b
:FILTER3_INITIAL_CONTIGS_END\+//g"

#Remove additional fileinput not required for this input type
sed -i -e "s/File3Path/File3Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

ReadFileType=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

sed -i -e "s/INPUT_FILE_POSS_3_START: Input3ReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $15}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

echo "Additional Input File (2):Length of 'reads' taken from each side of input
reads:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $16}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):Amplicon insert size:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

if [ "$ReadFileType" = "spf" ];then
InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of cycles for which input file is not used before being used
again:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

#remove the spare
sed -i -e "s/Input3ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

InputFileSetting=$(awk -F '[]' '{print $20}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):Required % identity for match:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $17}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set4\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of cycles to be skipped before input file is
used:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $18}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set5\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input File (2):No. of cycles for which input file is used:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $19}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set6\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```
    echo "Additional Input File (2):No. of cycles for which input file is not used before being used
again:${InputFileSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
fi
```

```
#### icf #####
```

```
#####
```

```
else
```

```
if [ "$InputFileType" = "Initial Contig Files" ];then
```

```
#Remove additional fileinput not required for this input type
```

```
sed -i -e "s/File3Path/File3Name2 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
ReadFileType=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/INPUT_FILE_POSS_3_START: Input3ReadType\+/-${ReadFileType}/g"
~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "(2) Additional Input File Type Set:${InputFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
echo "Additional Input File (2):Read File Type Set:${ReadFileType}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
#####--- icf
```

```
if [ "$ReadFileType" = "icf" ] || [ "$ReadFileType" = "icfNt" ];then
```

```
#remove spare settings
```

```
sed -i -e "s/Input3ReadType_Set4 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
sed -i -e "s/Input3ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
sed -i -e "s/Input3ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/Input3ReadType_Set1\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input File (2):No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/Input3ReadType_Set2\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input File (2):No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
sed -i -e "s/Input3ReadType_Set3\+/${InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input File (2):Const by which to multiply quality scores:${InputFileSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
else
```

```

#####--- picf

#remove spare settings
sed -i -e "s/Input3ReadType_Set5 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Input3ReadType_Set6 \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

#switch input file and first setting
sed -i -e "s/File3Path\FileName1 Input3ReadType_Set1\+ /Input3ReadType_Set1
File3Path\FileName1/g" ~/GGOSS/Scripts/PriceTiRun.sh

InputFileSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set1\+/{InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of initial contigs from this file:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $21}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set2\+/{InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of addition steps:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set3\+/{InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):No. of cycles per step:${InputFileSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

InputFileSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
sed -i -e "s/Input3ReadType_Set4\+/{InputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Additional Input File (2):Const by which to multiply quality scores:${InputFileSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

fi

fi
fi

# Adjust output file creation type and numeracy

OutputFileSetting=$(awk -F '[]' '{print $25}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$OutputFileSetting" = "fasta" ] || [ "$OutputFileSetting" = "priceq" ];then

sed -i -e "s/OutputType\+/{OutputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

sed -i -e "s/ -nco -o PriceTI_Filename.Output2Type\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

else

sed -i -e "s/OutputType\|+/fasta/g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Output2Type\|+/priceq/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

OutputFileSetting=$(awk -F '[]' '{print $26}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$OutputFileSetting" = "-" ];then

sed -i -e "s/-nco \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

sed -i -e "s/-nco\|+/-nco ${OutputFileSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#####
#-----
#####--set Other Parameters according to settings----#####
#-----
#####

OtherParamSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType1_nc \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType1_nc\|+/-nc ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "nc: Num. of cycles:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType2_link \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType2_link\|+/-link ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "link: Max num. contigs allowed to replace read in contig-edge
assembly:${OtherParamSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

OtherParamSetting=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType3_mol \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType3_mol\|+/-mol ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

echo "mol: Min overlap length for mini-assembly:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType4_tol \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType4_tol\+/-tol ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "tol: Threshold seq num for scaling overlap for contig-edge assemblies:${OtherParamSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

OtherParamSetting=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType5_mpi \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType5_mpi\+/-mpi ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "mpi and MPI: Min % ID for contig-edge assembly:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType6_tpi \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType6_tpi\+/-tpi ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "tpi and TPI: Threshold seq num for scaling % ID for contig-edge
assemblies:${OtherParamSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

MetaSetting=$(awk -F '[]' '{print $27}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$MetaSetting" = "Yes" ];then
echo "Meta-assembly:${MetaSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

OtherParamSetting=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" != "-" ];then
sed -i -e "s/-tpi\+/-TPI/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

OtherParamSetting=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" != "-" ];then
sed -i -e "s/-mpi\+/-MPI/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

```

fi

```
OtherParamSetting=$(awk -F '[]' '{print $7}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)
```

```
if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType7_dbmax \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType7_dbmax\+/-dbmax ${OtherParamSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "dbmax: Max length seq fed into de Bruijn assembly:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
OtherParamSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)
```

```
if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType8_dbk \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType8_dbk\+/-dbk ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "dbk: K-mer size for de Bruijn assembly:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
OtherParamSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)
```

```
if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType9_dbms \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType9_dbms\+/-dbms ${OtherParamSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "dbms:Min num. seq's to which de Bruijn assembly:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
OtherParamSetting=$(awk -F '[]' '{print $10}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)
```

```
if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType10_r \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType10_r\+/-r ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "r:Alignment score reward for nucleotide mismatch:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
OtherParamSetting=$(awk -F '[]' '{print $11}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)
```

```
if [ "$OtherParamSetting" = "-" ];then
```

```

sed -i -e "s/OtherParamType11_q \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType11_q\+/-q ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "q:Alignment score penalty for nucleotide mismatch:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $12}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType12_G \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType12_G\+/-G ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "G:Alignment score penalty for opening a gap:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

OtherParamSetting=$(awk -F '[]' '{print $13}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/OtherParamType13_E \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/OtherParamType13_E\+/-E ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "E:Alignment score penalty for extending a gap:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $14}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/NumberOfThreads \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/NumberOfThreads\+/-a ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Number of threads to use:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

OtherParamSetting=$(awk -F '[]' '{print $15}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "-" ];then
sed -i -e "s/MaxThreadsPerFile \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/MaxThreadsPerFile\+/-mtpf ${OtherParamSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

echo "Maximum threads per file:${OtherParamSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

fi

OtherParamSetting=$(awk -F '[]' '{print $16}'
~/GGOSS/tmp/PRICETI_ParameterSettingsChange.txt)

if [ "$OtherParamSetting" = "Concise standard output" ];then
sed -i -e "s/USER_INTERFACE_log\+/-log c/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Concise standard output set" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

if [ "$OtherParamSetting" = "No standard output" ];then
sed -i -e "s/USER_INTERFACE_log\+/-log n/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "No standard output set" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

if [ "$OtherParamSetting" = "Verbose standard output" ];then
sed -i -e "s/USER_INTERFACE_log\+/-log v/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Verbose standard output set" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

#####
#-----
#####--set filters according to settings---#####
#-----
#####
#####

#dont need to delete any as have done that on the input part
# input 1

InputFileSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

#if = read files and false paired end files---#####
###-----###
if [ "$InputFileSetting" = "Read Files" ] || [ "$InputFileSetting" = "False Paired-End Files" ]
then

ReadFilterSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FilterType1_rqf rqfnumber_a rqfnumber_b rqfnumber_c rqfnumber_d \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FilterType1_rqf\+/-rqf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: rqf: Yes" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/rqfnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, rqf requires an integer in option (a) - (% of nucleotides in read that
must be high quality)"
notify-send "ERROR! with settings, rqf requires an integer in option (a) - (% of nucleotides in
read that must be high quality)"

```

```

else
sed -i -e "s/rqfnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: % of nucleotides in read that must be high quality: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqfnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, rqf requires an integer in option (b) - (Min allowed probability of a
nucleotide being correct)"
notify-send "ERROR! with settings, rqf requires an integer in option (b) - (Min allowed
probability of a nucleotide being correct)"
else
sed -i -e "s/rqfnumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Min allowed probability of a nucleotide being correct: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqfnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqfnumber_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqfnumber_d \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqfnumber_d\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FilterType2_rnf rnfnumber_a rnfnumber_b rnfnumber_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FilterType2_rnf\+/-rnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: rnf: Yes" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

ReadFilterSetting=$(awk -F '[' '{print $7}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rmfnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, rnf requires an integer in option (a) - (% of nucleotides in a read
that must be called)"
notify-send "ERROR! with settings, rnf requires an integer in option (a) - (% of nucleotides in a
read that must be called)"

else
sed -i -e "s/rmfnumber_a\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: % of nucleotides in a read that must be called: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $8}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rmfnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/rmfnumber_b\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $9}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rmfnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/rmfnumber_c\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[' '{print $10}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/FilterType3_MaxHP MaxHPnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/FilterType3_MaxHP MaxHPnumber_a\+/-maxHp ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxHP: Filtering Reads: filter read pair if either nucleotide length has homo-polymer
track >: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

```

```

ReadFilterSetting=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/FilterType4_MaxDi MaxDiNumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/FilterType4_MaxDi MaxDiNumber_a\+/-maxDi ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxDi: Filtering Reads: filter read pair if eithers nucleotide length has repeating di-
nucleotide track >: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FilterType5_badf badfNumber_b badfNumber_a \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FilterType5_badf\+/-badf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: badf: Yes" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badfNumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, badf requires File path and name in option (a) - (File path and
name)"
notify-send "ERROR! with settings, badf requires File path and name in option (a) - (File path
and name)"

else
sed -i -e "s|badfNumber_a|${ReadFilterSetting}|g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: File path and name: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $14}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badfNumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, badf requires an integer in option (b) - (Min ungapped % identity
match to the above file before being prevented from being mapped to contigs)"
notify-send "ERROR! with settings, badf requires an integer in option (b) - (Min ungapped %
identity match to the above file before being prevented from being mapped to contigs)"

else
sed -i -e "s/badfNumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: Min ungapped % identity match to the above file before being prevented from being
mapped to contigs: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

fi

fi

```
ReadFilterSetting=$(awk -F '[' '{print $15}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FilterType6_repmask repmaskNumber_a repmaskNumber_b repmaskNumber_c
repmaskNumber_d repmaskNumber_e repmaskNumber_f repmaskNumber_g \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FilterType6_repmask\+/-repmask/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: repmask: Yes" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
ReadFilterSetting=$(awk -F '[' '{print $16}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmaskNumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, repmask requires an integer in option (a) - (Cycle number at which
repeats will be detected)"
notify-send "ERROR! with settings, repmask requires an integer in option (a) - (Cycle number at
which repeats will be detected)"
```

```
else
sed -i -e "s/repmaskNumber_a\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Cycle number at which repeats will be detected: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
ReadFilterSetting=$(awk -F '[' '{print $17}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
echo "repmask: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
if [ "$ReadFilterSetting" = "Start" ]
then
sed -i -e "s/repmaskNumber_b\+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/repmaskNumber_b\+//f/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

fi

```
ReadFilterSetting=$(awk -F '[' '{print $18}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmaskNumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, repmask requires an integer in option (c) - (Min num. of variance
units > median that will be counted as high-coverage)"
notify-send "ERROR! with settings, repmask requires an integer in option (c) - (Min num. of
variance units > median that will be counted as high-coverage)"
```

```
else
sed -i -e "s/repmaskNumber_c\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min num. of variance units > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```

fi

ReadFilterSetting=$(awk -F '[' '{print $19}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmaskNumber_d \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, repmask requires an integer in option (d) - (Min fold increase in
coverage > median that will be counted as high-coverage)"
notify-send "ERROR! with settings, repmask requires an integer in option (d) - (Min fold
increase in coverage > median that will be counted as high-coverage)"

else
sed -i -e "s/repmaskNumber_d\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min fold increase in coverage > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $20}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmaskNumber_e \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, repmask requires an integer in option (e) - (Min size in nt for a
detected repeat)"
notify-send "ERROR! with settings, repmask requires an integer in option (e) - (Min size in nt
for a detected repeat)"

else
sed -i -e "s/repmaskNumber_e\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min size in nt for a detected repeat: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $21}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmaskNumber_f \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, repmask requires an integer in option (f) - (Min %identity match to
a repeat for read to not be mapped to contigs)"
notify-send "ERROR! with settings, repmask requires an integer in option (f) - (Min %identity
match to a repeat for read to not be mapped to contigs)"

else
sed -i -e "s/repmaskNumber_f\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min %identity match to a repeat for read to not be mapped to contigs:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $22}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "Not applicable" ]
then
sed -i -e "s/repmaskNumber_g \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

else
    sed -i -e "s/repmaskNumber_g\+\/repmask.${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh
    echo "repmask: Output file to which the detected repeats will be written:
repmask.${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
    sed -i -e "s/FilterType7_reset resetNumber_a \+\/g" ~/GGOSS/Scripts/PriceTiRun.sh
else
    sed -i -e "s/FilterType7_reset\+\/-reset/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: reset: Yes" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
    echo "ERROR! with settings, reset requires an integer or list of integers in option (a) - (List the
cycles to be reset)"
    notify-send "ERROR! with settings, repmask requires an integer in option (a) - (List the cycles to
be reset)"
else
    sed -i -e "s/resetNumber_a\+\/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Reads: reset: List the cycles to be reset: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

fi

fi

#dont need to delete any as have done that on the input part
# additional input 1

InputFileSetting=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

#if = read files and false paired end files----#####
###-----#####
if [ "$InputFileSetting" = "Read Files" ] || [ "$InputFileSetting" = "False Paired-End Files" ]
then

ReadFilterSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
    sed -i -e "s/Filter2Type1_rqf rqf2number_a rqf2number_b rqf2number_c rqf2number_d \+\/g"
~/GGOSS/Scripts/PriceTiRun.sh
else
    sed -i -e "s/Filter2Type1_rqf\+\/-rqf/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

echo "Additional Input (1): Filtering Reads: rqf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/rqf2number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), rqf requires an integer in option (a) - (% of
nucleotides in read that must be high quality)"
notify-send "ERROR! with settings in additional input (1), rqf requires an integer in option (a) -
(% of nucleotides in read that must be high quality)"

else
sed -i -e "s/rqf2number_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: % of nucleotides in read that must be high quality: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqf2number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), rqf requires an integer in option (b) - (Min
allowed probability of a nucleotide being correct)"
notify-send "ERROR! with settings in additional input (1), rqf requires an integer in option (b) -
(Min allowed probability of a nucleotide being correct)"
else
sed -i -e "s/rqf2number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Min allowed probability of a nucleotide being correct: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqf2number_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqf2number_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqf2number_d \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqf2number_d\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter2Type2_rnf rnf2number_a rnf2number_b rnf2number_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2Type2_rnf\+/-rnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (1): Filtering Reads: rnf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
ReadFilterSetting=$(awk -F '[]' '{print $7}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf2number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), rnf requires an integer in option (a) - (% of
nucleotides in a read that must be called)"
notify-send "ERROR! with settings in additional input (1), rnf requires an integer in option (a) -
(% of nucleotides in a read that must be called)"
else
sed -i -e "s/rnf2number_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: % of nucleotides in a read that must be called: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf2number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rnf2number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf2number_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rnf2number_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

fi

```

ReadFilterSetting=$(awk -F '[]' '{print $10}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter2Type3_MaxHP MaxHP2number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/Filter2Type3_MaxHP MaxHP2number_a\+/-maxHp ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxHP: Additional Input (1): Filtering Reads: filter read pair if eithers nucleotide length
has homo-polymer track >: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter2Type4_MaxDi MaxDi2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/Filter2Type4_MaxDi MaxDi2Number_a\+/-maxDi ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxDi: Additional Input (1): Filtering Reads: filter read pair if eithers nucleotide length
has repeating di-nucleotide track >: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter2Type5_badf badf2Number_b badf2Number_a \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2Type5_badf\+/-badf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (1): Filtering Reads: badf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badf2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), badf requires File path and name in option
(a) - (File path and name)"
notify-send "ERROR! with settings in additional input (1), badf requires File path and name in
option (a) - (File path and name)"

else
sed -i -e "s|badf2Number_a|${ReadFilterSetting}|g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: File path and name: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

fi

ReadFilterSetting=$(awk -F '[' '{print $14}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badf2Number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), badf requires an integer in option (b) - (Min
ungapped % identity match to the above file before being prevented from being mapped to contigs)"
notify-send "ERROR! with settings in additional input (1), badf requires an integer in option (b) -
(Min ungapped % identity match to the above file before being prevented from being mapped to
contigs)"

else
sed -i -e "s/badf2Number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: Min ungapped % identity match to the above file before being prevented from being
mapped to contigs: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[' '{print $15}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter2Type6_repmask repmask2Number_a repmask2Number_b repmask2Number_c
repmask2Number_d repmask2Number_e repmask2Number_f repmask2Number_g \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2Type6_repmask\+/-repmask/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (1): Filtering Reads: repmask: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $16}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), repmask requires an integer in option (a) -
(Cycle number at which repeats will be detected)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(a) - (Cycle number at which repeats will be detected)"

else
sed -i -e "s/repmask2Number_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Cycle number at which repeats will be detected: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[' '{print $17}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
echo "repmask: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
if [ "$ReadFilterSetting" = "Start" ]
then
sed -i -e "s/repmask2Number_b\+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

else
sed -i -e "s/repmask2Number_b\|+/f/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

ReadFilterSetting=$(awk -F '[' '{print $18}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask2Number_c \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), repmask requires an integer in option (c) -
(Min num. of variance units > median that will be counted as high-coverage)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(c) - (Min num. of variance units > median that will be counted as high-coverage)"

else
sed -i -e "s/repmask2Number_c\|+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min num. of variance units > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $19}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask2Number_d \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), repmask requires an integer in option (d) -
(Min fold increase in coverage > median that will be counted as high-coverage)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(d) - (Min fold increase in coverage > median that will be counted as high-coverage)"

else
sed -i -e "s/repmask2Number_d\|+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min fold increase in coverage > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $20}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask2Number_e \|+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), repmask requires an integer in option (e) -
(Min size in nt for a detected repeat)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(e) - (Min size in nt for a detected repeat)"

else
sed -i -e "s/repmask2Number_e\|+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min size in nt for a detected repeat: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $21}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

```

```

then
sed -i -e "s/repmask2Number_f \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (1), repmask requires an integer in option (f) -
(Min %identity match to a repeat for read to not be mapped to contigs)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(f) - (Min %identity match to a repeat for read to not be mapped to contigs)"

else
sed -i -e "s/repmask2Number_f \+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min %identity match to a repeat for read to not be mapped to contigs:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $22}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "Not applicable" ]
then
sed -i -e "s/repmask2Number_g \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/repmask2Number_g \+ /repmask_ForInput2.${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Output file to which the detected repeats will be written:
repmask_ForInput2.${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $23}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter2Type7_reset reset2Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2Type7_reset \+/-reset/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (1): Filtering Reads: reset: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
echo "ERROR! with settings in additional input (1), reset requires an integer or list of integers in
option (a) - (List the cycles to be reset)"
notify-send "ERROR! with settings in additional input (1), repmask requires an integer in option
(a) - (List the cycles to be reset)"
else
sed -i -e "s/reset2Number_a \+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (1): Filtering Reads: reset: List the cycles to be reset: ${ReadFilterSetting}"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

fi

```

fi

```
#dont need to delete any as have done that on the input part
# additional input 1
```

```
InputFileSetting=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
#if = read files and false paired end files----#####
```

```
###-----#####
```

```
if [ "$InputFileSetting" = "Read Files" ] || [ "$InputFileSetting" = "False Paired-End Files" ]
then
```

```
ReadFilterSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```
if [ "$ReadFilterSetting" = "No" ]
```

```
then
```

```
sed -i -e "s/Filter3Type1_rqf rqf3number_a rqf3number_b rqf3number_c rqf3number_d \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
```

```
else
```

```
sed -i -e "s/Filter3Type1_rqf\+/-rqf/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "Additional Input (2): Filtering Reads: rqf: Yes" >>
```

```
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
ReadFilterSetting=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```
if [ "$ReadFilterSetting" = "-" ]
```

```
then
```

```
sed -i -e "s/rqf3number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "ERROR! with settings in additional input (2), rqf requires an integer in option (a) - (% of
nucleotides in read that must be high quality)"
```

```
notify-send "ERROR! with settings in additional input (2), rqf requires an integer in option (a) -
(% of nucleotides in read that must be high quality)"
```

```
else
```

```
sed -i -e "s/rqf3number_a\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "rqf: % of nucleotides in read that must be high quality: ${ReadFilterSetting}" >>
```

```
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```
if [ "$ReadFilterSetting" = "-" ]
```

```
then
```

```
sed -i -e "s/rqf3number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "ERROR! with settings in additional input (2), rqf requires an integer in option (b) - (Min
allowed probability of a nucleotide being correct)"
```

```
notify-send "ERROR! with settings in additional input (2), rqf requires an integer in option (b) -
(Min allowed probability of a nucleotide being correct)"
```

```
else
```

```
sed -i -e "s/rqf3number_b\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
echo "rqf: Min allowed probability of a nucleotide being correct: ${ReadFilterSetting}" >>
```

```
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
```

```

if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqf3number_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqf3number_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $5}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rqf3number_d \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/rqf3number_d\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rqf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $6}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3Type2_rnf rnf3number_a rnf3number_b rnf3number_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3Type2_rnf\+/-rnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (2): Filtering Reads: rnf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $7}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf3number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), rnf requires an integer in option (a) - (% of
nucleotides in a read that must be called)"
notify-send "ERROR! with settings in additional input (2), rqf requires an integer in option (a) -
(% of nucleotides in a read that must be called)"

else
sed -i -e "s/rnf3number_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: % of nucleotides in a read that must be called: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf3number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else

```

```

sed -i -e "s/rnf3number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Cycles passed: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $9}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/rnf3number_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/rnf3number_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "rnf: Number of cycles to run for: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi
fi

ReadFilterSetting=$(awk -F '[]' '{print $10}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3Type3_MaxHP MaxHP3number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/Filter3Type3_MaxHP MaxHP3number_a\+/-maxHp ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxHP: Additional Input (2): Filtering Reads: filter read pair if either nucleotide length
has homo-polymer track >: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $11}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3Type4_MaxDi MaxDi3Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/Filter3Type4_MaxDi MaxDi3Number_a\+/-maxDi ${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh

echo "maxDi: Additional Input (2): Filtering Reads: filter read pair if either nucleotide length
has repeating di-nucleotide track >: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $12}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3Type5_badf badf3Number_b badf3Number_a \+//g"
~/GGOSS/Scripts/PriceTiRun.sh

```

```

else
sed -i -e "s/Filter3Type5_badf\+/-badf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (2): Filtering Reads: badf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $13}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badf3Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), badf requires File path and name in option
(a) - (File path and name)"
notify-send "ERROR! with settings in additional input (2), badf requires File path and name in
option (a) - (File path and name)"

else
sed -i -e "s|badf3Number_a|${ReadFilterSetting}|g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: File path and name: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $14}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/badf3Number_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), badf requires an integer in option (b) - (Min
ungapped % identity match to the above file before being prevented from being mapped to contigs)"
notify-send "ERROR! with settings in additional input (2), badf requires an integer in option (b) -
(Min ungapped % identity match to the above file before being prevented from being mapped to
contigs)"

else
sed -i -e "s/badf3Number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "badf: Min ungapped % identity match to the above file before being prevented from being
mapped to contigs: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi
fi

ReadFilterSetting=$(awk -F '[]' '{print $15}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3Type6_repmask repmask3Number_a repmask3Number_b repmask3Number_c
repmask3Number_d repmask3Number_e repmask3Number_f repmask3Number_g \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3Type6_repmask\+/-repmask/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (2): Filtering Reads: repmask: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $16}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask3Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

    echo "ERROR! with settings in additional input (2), repmask requires an integer in option (a) -
(Cycle number at which repeats will be detected)"
    notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(a) - (Cycle number at which repeats will be detected)"

    else
    sed -i -e "s/repmask3Number_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "repmask: Cycle number at which repeats will be detected: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $17}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
echo "repmask: ${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
if [ "$ReadFilterSetting" = "Start" ]
then
sed -i -e "s/repmask3Number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/repmask3Number_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

ReadFilterSetting=$(awk -F '[' '{print $18}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask3Number_c \+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), repmask requires an integer in option (c) -
(Min num. of variance units > median that will be counted as high-coverage)"
notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(c) - (Min num. of variance units > median that will be counted as high-coverage)"

else
sed -i -e "s/repmask3Number_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min num. of variance units > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $19}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask3Number_d \+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), repmask requires an integer in option (d) -
(Min fold increase in coverage > median that will be counted as high-coverage)"
notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(d) - (Min fold increase in coverage > median that will be counted as high-coverage)"

else
sed -i -e "s/repmask3Number_d\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min fold increase in coverage > median that will be counted as high-coverage:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $20}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

```

```

if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask3Number_e \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), repmask requires an integer in option (e) -
(Min size in nt for a detected repeat)"
notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(e) - (Min size in nt for a detected repeat)"

else
sed -i -e "s/repmask3Number_e\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min size in nt for a detected repeat: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

fi

```

ReadFilterSetting=$(awk -F '[' '{print $21}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/repmask3Number_f \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in additional input (2), repmask requires an integer in option (f) -
(Min %identity match to a repeat for read to not be mapped to contigs)"
notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(f) - (Min %identity match to a repeat for read to not be mapped to contigs)"

else
sed -i -e "s/repmask3Number_f\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Min %identity match to a repeat for read to not be mapped to contigs:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

fi

```

ReadFilterSetting=$(awk -F '[' '{print $22}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "Not applicable" ]
then
sed -i -e "s/repmask3Number_g \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

else
sed -i -e "s/repmask3Number_g\+/repmask_ForInput3.${ReadFilterSetting}/g"
~/GGOSS/Scripts/PriceTiRun.sh
echo "repmask: Output file to which the detected repeats will be written:
repmask_ForInput3.${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

fi

fi

```

ReadFilterSetting=$(awk -F '[' '{print $23}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)

```

```

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3Type7_reset reset3Number_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3Type7_reset\+/-reset/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

echo "Additional Input (2): Filtering Reads: reset: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $24}' ~/GGOSS/tmp/PRICETI_FilterSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
echo "ERROR! with settings in additional input (2), reset requires an integer or list of integers in
option (a) - (List the cycles to be reset)"
notify-send "ERROR! with settings in additional input (2), repmask requires an integer in option
(a) - (List the cycles to be reset)"
else
sed -i -e "s/reset3Number_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional Input (2): Filtering Reads: reset: List the cycles to be reset: $ReadFilterSetting"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

fi

fi

#####
#-----
#####--set initial contig filter according to settings----#####
#-----
#####

# For input 1

InputFileType=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" = "Initial Contig Files" ];then

ReadFilterSetting=$(awk -F '[]' '{print $23}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FILTER1_INITIAL_CONTIGS_START: Filter1ICFlag_icbf Filter1IC_icbf_a
Filter1IC_icbf_b Filter1IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FILTER1_INITIAL_CONTIGS_START: Filter1ICFlag_icbf\+/-icbf/g"
~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icbf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $24}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then

sed -i -e "s/Filter1IC_icbf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icbf requires a file and file path in option (a) - (Sequence
file and path)"
notify-send "ERROR! with settings in input, icbf requires a file and file path in option (a) -
(Sequence file and path)"

```

```

else
sed -i -e "s/Filter1IC_icbf_a|${ReadFilterSetting}|g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icbf: Sequence file and path: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $25}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icbf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icbf requires an integer in option (b) - (Min % identity to
sequence in file)"
notify-send "ERROR! with settings in input, icbf requires an integer in option (b) - (Min %
identity to sequence in file)"
else
sed -i -e "s/Filter1IC_icbf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icbf: Min % identity to sequence in file: $ReadFilterSetting"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $26}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1IC_icbf_c\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icbf: Don't apply filter to sequences >: $ReadFilterSetting"
>> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi

ReadFilterSetting=$(awk -F '[' '{print $27}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter1ICFlag_icmHp Filter1IC_icmHp_a Filter1IC_icmHp_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1ICFlag_icmHp\+/-icmHp/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmHp: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $28}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then

sed -i -e "s/Filter1ICFlag_icmHp_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icmHp requires an integer in option (a) - (Filter out initial
contig if its homo-polymer track nucleotide length is >)"
notify-send "ERROR! with settings in input, icmHp requires an integer in option (a) - (Filter out
initial contig if its homo-polymer track nucleotide length is >)"
else

```

```

sed -i -e "s/Filter1IC_icmHp_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmHp: Filter out initial contig if its homo-polymer track
nucleotide length is >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

```

```

ReadFilterSetting=$(awk -F '[]' '{print $29}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icmHp_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1IC_icmHp_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmHp: Don't apply this filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

```

fi

```

ReadFilterSetting=$(awk -F '[]' '{print $30}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter1ICFlag_icmDi Filter1IC_icmDi_a Filter1IC_icmDi_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1ICFlag_icmDi\+/-icmDi/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmDi: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

ReadFilterSetting=$(awk -F '[]' '{print $31}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icmDi_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icmDi requires an integer in option (a) - (Filter out initial
contig if its repeating di-nucleotide track nucleotide length is >)"
notify-send "ERROR! with settings in input, icmDi requires an integer in option (a) - (Filter out
initial contig if its repeating di-nucleotide track nucleotide length is >)"
else
sed -i -e "s/Filter1IC_icmDi_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmDi: Filter out initial contig if its repeating di-nucleotide
track nucleotide length is >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

```

```

ReadFilterSetting=$(awk -F '[]' '{print $32}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icmDi_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1IC_icmDi_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icmDi: Don't apply this filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

```

fi

```
ReadFilterSetting=$(awk -F '[]' '{print $33}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter1ICFlag_icqf Filter1IC_icqf_a Filter1IC_icqf_b Filter1IC_icqf_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1ICFlag_icqf\+/-icqf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icqf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
ReadFilterSetting=$(awk -F '[]' '{print $34}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icqf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icqf requires an integer in option (a) - (% of nucleotides in
a read that must be high quality)"
notify-send "ERROR! with settings in input, icqf requires an integer in option (a) - (% of
nucleotides in a read that must be high quality)"
else
sed -i -e "s/Filter1IC_icqf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icqf: % of nucleotides in a read that must be high quality:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
```

```
ReadFilterSetting=$(awk -F '[]' '{print $35}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icqf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icqf requires an integer in option (b) - (Min allowed
probability of a nucleotide being correct)"
notify-send "ERROR! with settings in input, icqf requires an integer in option (b) - (Min allowed
probability of a nucleotide being correct)"
else
sed -i -e "s/Filter1IC_icqf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icqf: Min allowed probability of a nucleotide being correct:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
```

```
ReadFilterSetting=$(awk -F '[]' '{print $36}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter1IC_icqf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter1IC_icqf_c\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icqf: Don't apply this filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
```

fi

```

ReadFilterSetting=$(awk -F '[' '{print $37}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FilterIICFlag_icnf FilterIIC_icnf_a FilterIIC_icnf_b
:FILTER1_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/:FILTER1_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/FilterIICFlag_icnf\+/-icnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icnf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $38}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/FilterIIC_icnf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings in input, icnf requires an integer in option (a) - (% nucleotides in a
read that must be called)"
notify-send "ERROR! with settings in input, icnf requires an integer in option (a) - (%
nucleotides in a read that must be called)"
else
sed -i -e "s/FilterIIC_icnf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icnf: % nucleotides in a read that must be called:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $39}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/FilterIIC_icnf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FilterIIC_icnf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering Initial Contig Files: icnf: Don't apply this filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi
fi

#####
#####

# potential input 2

InputFileType=$(awk -F '[' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" = "Initial Contig Files" ];then

ReadFilterSetting=$(awk -F '[' '{print $23}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then

```

```

    sed -i -e "s/FILTER2_INITIAL_CONTIGS_START: Filter2ICFlag_icbf Filter2IC_icbf_a
Filter2IC_icbf_b Filter2IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/FILTER2_INITIAL_CONTIGS_START: Filter2ICFlag_icbf\+/-icbf/g"
~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icbf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    ReadFilterSetting=$(awk -F '[' '{print $24}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icbf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icbf requires a file and file path in
option (a) - (Sequence file and path)"
    notify-send "Additional input 1: ERROR! with settings in input, icbf requires a file and file path
in option (a) - (Sequence file and path)"
    else
    sed -i -e "s/Filter2IC_icbf_a|${ReadFilterSetting}|g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icbf: Sequence file and path:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

    ReadFilterSetting=$(awk -F '[' '{print $25}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icbf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icbf requires an integer in option (b) -
(Min % identity to sequence in file)"
    notify-send "Additional input 1: ERROR! with settings in input, icbf requires an integer in option
(b) - (Min % identity to sequence in file)"
    else
    sed -i -e "s/Filter2IC_icbf_b|+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icbf: Min % identity to sequence in file:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

    ReadFilterSetting=$(awk -F '[' '{print $26}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2IC_icbf_c|+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icbf: Don't apply filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi
fi

    ReadFilterSetting=$(awk -F '[' '{print $27}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]
    then

```

```

    sed -i -e "s/Filter2ICFlag_icmHp Filter2IC_icmHp_a Filter2IC_icmHp_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2ICFlag_icmHp\+/-icmHp/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmHp: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    ReadFilterSetting=$(awk -F '[' '{print $28}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then

    sed -i -e "s/Filter2ICFlag_icmHp_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icmHp requires an integer in option (a)
- (Filter out initial contig if its homo-polymer track nucleotide length is >)"
    notify-send "Additional input 1: ERROR! with settings in input, icmHp requires an integer in
option (a) - (Filter out initial contig if its homo-polymer track nucleotide length is >)"
    else
    sed -i -e "s/Filter2IC_icmHp_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmHp: Filter out initial contig if its
homo-polymer track nucleotide length is >: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

    ReadFilterSetting=$(awk -F '[' '{print $29}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icmHp_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2IC_icmHp_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmHp: Don't apply this filter to
sequences >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi
    fi

    ReadFilterSetting=$(awk -F '[' '{print $30}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]
    then
    sed -i -e "s/Filter2ICFlag_icmDi Filter2IC_icmDi_a Filter2IC_icmDi_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2ICFlag_icmDi\+/-icmDi/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmDi: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    ReadFilterSetting=$(awk -F '[' '{print $31}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icmDi_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icmDi requires an integer in option (a)
- (Filter out initial contig if its repeating di-nucleotide track nucleotide length is >)"

```

```

    notify-send "Additional input 1: ERROR! with settings in input, icmDi requires an integer in
option (a) - (Filter out initial contig if its repeating di-nucleotide track nucleotide length is >)"
    else
    sed -i -e "s/Filter2IC_icmDi_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmDi: Filter out initial contig if its
repeating di-nucleotide track nucleotide length is >: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $32}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icmDi_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2IC_icmDi_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icmDi: Don't apply this filter to
sequences >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi
fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $33}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]
    then
    sed -i -e "s/Filter2ICFlag_icqf Filter2IC_icqf_a Filter2IC_icqf_b Filter2IC_icqf_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/Filter2ICFlag_icqf\+/-icqf/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icqf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $34}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icqf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icqf requires an integer in option (a) -
(% of nucleotides in a read that must be high quality)"
    notify-send "Additional input 1: ERROR! with settings in input, icqf requires an integer in option
(a) - (% of nucleotides in a read that must be high quality)"
    else
    sed -i -e "s/Filter2IC_icqf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: Filtering Initial Contig Files: icqf: % of nucleotides in a read that must
be high quality: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $35}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
    sed -i -e "s/Filter2IC_icqf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Additional input 1: ERROR! with settings in input, icqf requires an integer in option (b) -
(Min allowed probability of a nucleotide being correct)"

```

notify-send "Additional input 1: ERROR! with settings in input, icqf requires an integer in option (b) - (Min allowed probability of a nucleotide being correct)"

```
else
sed -i -e "s/Filter2IC_icqf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 1: Filtering Initial Contig Files: icqf: Min allowed probability of a
nucleotide being correct: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
```

```
ReadFilterSetting=$(awk -F '[' '{print $36}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter2IC_icqf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2IC_icqf_c\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 1: Filtering Initial Contig Files: icqf: Don't apply this filter to sequences
>: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi
```

```
ReadFilterSetting=$(awk -F '[' '{print $37}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter2ICFlag_icnf Filter2IC_icnf_a Filter2IC_icnf_b
:FILTER2_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/:FILTER2_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Filter2ICFlag_icnf\+/-icnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 1: Filtering Initial Contig Files: icnf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
```

```
ReadFilterSetting=$(awk -F '[' '{print $38}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter2IC_icnf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 1: ERROR! with settings in input, icnf requires an integer in option (a) -
(% nucleotides in a read that must be called)"
notify-send "Additional input 1: ERROR! with settings in input, icnf requires an integer in option
(a) - (% nucleotides in a read that must be called)"
else
sed -i -e "s/Filter2IC_icnf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 1: Filtering Initial Contig Files: icnf: % nucleotides in a read that must be
called: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
```

```
ReadFilterSetting=$(awk -F '[' '{print $39}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter2IC_icnf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter2IC_icnf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```

    echo "Additional input 1: Filtering Initial Contig Files: icnf: Don't apply this filter to sequences
>: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi
fi
# potential input 3

InputFileType=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" = "Initial Contig Files" ];then

    ReadFilterSetting=$(awk -F '[]' '{print $23}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]
    then
        sed -i -e "s/FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf Filter3IC_icbf_a
Filter3IC_icbf_b Filter3IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/FILTER3_INITIAL_CONTIGS_START: Filter3ICFlag_icbf\+/-icbf/g"
~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: Filtering Initial Contig Files: icbf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

        ReadFilterSetting=$(awk -F '[]' '{print $24}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
        if [ "$ReadFilterSetting" = "-" ]
        then
            sed -i -e "s/Filter3IC_icbf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
            echo "Additional input 2: ERROR! with settings in input, icbf requires a file and file path in
option (a) - (Sequence file and path)"
            notify-send "Additional input 2: ERROR! with settings in input, icbf requires a file and file path
in option (a) - (Sequence file and path)"
        else
            sed -i -e "s/Filter3IC_icbf_a\${ReadFilterSetting}\|g" ~/GGOSS/Scripts/PriceTiRun.sh
            echo "Additional input 2: Filtering Initial Contig Files: icbf: Sequence file and path:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
        fi

        ReadFilterSetting=$(awk -F '[]' '{print $25}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
        if [ "$ReadFilterSetting" = "-" ]
        then
            sed -i -e "s/Filter3IC_icbf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
            echo "Additional input 2: ERROR! with settings in input, icbf requires an integer in option (b) -
(Min % identity to sequence in file)"
            notify-send "Additional input 2: ERROR! with settings in input, icbf requires an integer in option
(b) - (Min % identity to sequence in file)"
        else
            sed -i -e "s/Filter3IC_icbf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
            echo "Additional input 2: Filtering Initial Contig Files: icbf: Min % identity to sequence in file:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
        fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $26}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
        sed -i -e "s/Filter3IC_icbf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/Filter3IC_icbf_c\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: Filtering Initial Contig Files: icbf: Don't apply filter to sequences >:
$ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi
fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $27}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]
    then
        sed -i -e "s/Filter3ICFlag_icmHp Filter3IC_icmHp_a Filter3IC_icmHp_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/Filter3ICFlag_icmHp\+/-icmHp/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: Filtering Initial Contig Files: icmHp: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $28}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
        sed -i -e "s/Filter3ICFlag_icmHp_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: ERROR! with settings in input, icmHp requires an integer in option (a)
- (Filter out initial contig if its homo-polymer track nucleotide length is >)"
        notify-send "Additional input 2: ERROR! with settings in input, icmHp requires an integer in
option (a) - (Filter out initial contig if its homo-polymer track nucleotide length is >)"
    else
        sed -i -e "s/Filter3IC_icmHp_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: Filtering Initial Contig Files: icmHp: Filter out initial contig if its
homo-polymer track nucleotide length is >: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $29}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]
    then
        sed -i -e "s/Filter3IC_icmHp_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/Filter3IC_icmHp_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Additional input 2: Filtering Initial Contig Files: icmHp: Don't apply this filter to
sequences >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
    fi

```

fi

```

    ReadFilterSetting=$(awk -F '[]' '{print $30}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "No" ]

```

```

then
sed -i -e "s/Filter3ICFlag_icmDi Filter3IC_icmDi_a Filter3IC_icmDi_b \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3ICFlag_icmDi\+/-icmDi/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icmDi: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $31}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icmDi_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: ERROR! with settings in input, icmDi requires an integer in option (a)
- (Filter out initial contig if its repeating di-nucleotide track nucleotide length is >)"
notify-send "Additional input 2: ERROR! with settings in input, icmDi requires an integer in
option (a) - (Filter out initial contig if its repeating di-nucleotide track nucleotide length is >)"
else
sed -i -e "s/Filter3IC_icmDi_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icmDi: Filter out initial contig if its
repeating di-nucleotide track nucleotide length is >: $ReadFilterSetting" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $32}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icmDi_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3IC_icmDi_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icmDi: Don't apply this filter to
sequences >: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi

ReadFilterSetting=$(awk -F '[' '{print $33}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3ICFlag_icqf Filter3IC_icqf_a Filter3IC_icqf_b Filter3IC_icqf_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3ICFlag_icqf\+/-icqf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icqf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $34}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icqf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: ERROR! with settings in input, icqf requires an integer in option (a) -
(% of nucleotides in a read that must be high quality)"

```

```

notify-send "Additional input 2: ERROR! with settings in input, icqf requires an integer in option
(a) - (% of nucleotides in a read that must be high quality)"
else
sed -i -e "s/Filter3IC_icqf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icqf: % of nucleotides in a read that must
be high quality: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $35}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icqf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: ERROR! with settings in input, icqf requires an integer in option (b) -
(Min allowed probability of a nucleotide being correct)"
notify-send "Additional input 2: ERROR! with settings in input, icqf requires an integer in option
(b) - (Min allowed probability of a nucleotide being correct)"
else
sed -i -e "s/Filter3IC_icqf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icqf: Min allowed probability of a
nucleotide being correct: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[' '{print $36}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icqf_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3IC_icqf_c\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icqf: Don't apply this filter to sequences
>: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi

ReadFilterSetting=$(awk -F '[' '{print $37}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/Filter3ICFlag_icnf Filter3IC_icnf_a Filter3IC_icnf_b
:FILTER3_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/:FILTER3_INITIAL_CONTIGS_END \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/Filter3ICFlag_icnf\+/-icnf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icnf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[' '{print $38}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icnf_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: ERROR! with settings in input, icnf requires an integer in option (a) -
(% nucleotides in a read that must be called)"

```

```

notify-send "Additional input 2: ERROR! with settings in input, icnf requires an integer in option
(a) - (% nucleotides in a read that must be called)"
else
sed -i -e "s/Filter3IC_icnf_a\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icnf: % nucleotides in a read that must be
called: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi

ReadFilterSetting=$(awk -F '[]' '{print $39}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]
then
sed -i -e "s/Filter3IC_icnf_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/Filter3IC_icnf_b\+/$ReadFilterSetting/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Additional input 2: Filtering Initial Contig Files: icnf: Don't apply this filter to sequences
>: $ReadFilterSetting" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt
fi
fi
fi

```

```

#####
#-----
#####--set filtering.processing assembled contigs according to settings---
#####
#-----
#####

```

```

ReadFilterSetting=$(awk -F '[]' '{print $1}' ~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)

```

```

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/FILTERING_PROCESSING_ASSEMBLED_CONTIGS_START: lenfflag
lenfnumber_a lenfnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/FILTERING_PROCESSING_ASSEMBLED_CONTIGS_START: \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/lenfflag\+/-lenf/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering/Processing Assembled Contigs: lenf: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

```

```

ReadFilterSetting=$(awk -F '[]' '{print $2}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/lenfnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, lenf requires an integer in option (a) - (Filter contigs at the end of
every cycle shorter than (nt's))"
notify-send "ERROR! with settings, lenf requires an integer in option (a) - (Filter contigs at the
end of every cycle shorter than (nt's))"

else
sed -i -e "s/lenfnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

    echo "lenf: Filter contigs at the end of every cycle shorter than (nt's): ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $3}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/lenfnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, lenf requires an integer in option (b) - (Number of cycles to skip
before filtering contigs)"
notify-send "ERROR! with settings, lenf requires an integer in option (b) - (Number of cycles to
skip before filtering contigs)"

else
sed -i -e "s/lenfnumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "lenf: Number of cycles to skip before filtering contigs: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi
fi

ReadFilterSetting=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/trimflag trimnumber_a trimnumber_b trimnumber_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/trimflag\+/-trim/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering/Processing Assembled Contigs: trim: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $5}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/trimnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, trim requires an integer in option (a) - (Run on the end of cycle
number)"
notify-send "ERROR! with settings, trim requires an integer in option (a) - (Run on the end of
cycle number)"

else
sed -i -e "s/trimnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "trim: Run on the end of cycle number: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

```

```

    ReadFilterSetting=$(awk -F '[]' '{print $6}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt
    if [ "$ReadFilterSetting" = "-" ]

    then
        sed -i -e "s/trimnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "ERROR! with settings, trim requires an integer in option (b) - (Min coverage level for
trimming edges of contigs)"
        notify-send "ERROR! with settings, trim requires an integer in option (b) - (Min coverage level
for trimming edges of contigs)"

    else
        sed -i -e "s/trimnumber_b\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "trim: Min coverage level for trimming edges of contigs: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    fi

    ReadFilterSetting=$(awk -F '[]' '{print $7}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt
    if [ "$ReadFilterSetting" = "-" ]

    then
        sed -i -e "s/trimnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/trimnumber_c\+/{ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "trim: After trimming delete contigs shorter than: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    fi
fi

ReadFilterSetting=$(awk -F '[]' '{print $8}' ~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)

    if [ "$ReadFilterSetting" = "No" ]
    then
        sed -i -e "s/trimBflag trimBnumber_a trimBnumber_b trimBnumber_c \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
        sed -i -e "s/trimBflag\+/-trimB/g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "Filtering/Processing Assembled Contigs: trimB: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    ReadFilterSetting=$(awk -F '[]' '{print $9}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]

    then
        sed -i -e "s/trimBnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
        echo "ERROR! with settings, trimB requires an integer in option (a) - (Number of cycles to
skip)"
        notify-send "ERROR! with settings, trimB requires an integer in option (a) - (Number of cycles
to skip)"

    else

```

```

sed -i -e "s/trimBnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "trimB: Number of cycles to skip: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $10}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/trimBnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, trimB requires an integer in option (b) - (Min coverage level for
trimming edges of contigs at end of every cycle)"
notify-send "ERROR! with settings, trimB requires an integer in option (b) - (Min coverage level
for trimming edges of contigs at end of every cycle)"

else
sed -i -e "s/trimBnumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "trimB: Min coverage level for trimming edges of contigs at end of every cycle:
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[]' '{print $11}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/trimBnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/trimBnumber_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "trimB: After trimming delete contigs shorter than: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $12}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)

if [ "$ReadFilterSetting" = "No" ]
then
sed -i -e "s/trimIflag trimInumber_a trimInumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
sed -i -e "s/trimIflag\+/-trimI/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "Filtering/Processing Assembled Contigs: trimI: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

ReadFilterSetting=$(awk -F '[]' '{print $13}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/trimInumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh

```

```

    echo "ERROR! with settings, trimI requires an integer in option (a) - (Minimal coverage level to
trim to)"
    notify-send "ERROR! with settings, trimI requires an integer in option (a) - (Minimal coverage
level to trim to)"

    else
    sed -i -e "s/trimInumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "trimI: Minimal coverage level to trim to: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    fi

    ReadFilterSetting=$(awk -F '[]' '{print $14}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]

    then
    sed -i -e "s/trimInumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/trimInumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "trimI: After trimming delete contigs shorter than: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $15}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)

    if [ "$ReadFilterSetting" = "No" ]
    then
    sed -i -e "s/targetflag targetnumber_a targetnumber_b targetnumber_c targetnumber_d \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    else
    sed -i -e "s/targetflag\+/-target/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Filtering/Processing Assembled Contigs: target: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    ReadFilterSetting=$(awk -F '[]' '{print $16}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
    if [ "$ReadFilterSetting" = "-" ]

    then
    sed -i -e "s/targetnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "ERROR! with settings, target requires an integer in option (a) - (% identity to an input
initial contig to count as a match (ungapped))"
    notify-send "ERROR! with settings, target requires an integer in option (a) - (% identity to an
input initial contig to count as a match (ungapped))"

    else
    sed -i -e "s/targetnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "target: % identity to an input initial contig to count as a match (ungapped):
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

    fi

```

```

ReadFilterSetting=$(awk -F '[]' '{print $17}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt
  if [ "$ReadFilterSetting" = "-" ]

  then
    sed -i -e "s/targetnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "ERROR! with settings, target requires an integer in option (b) - (Num. cycles to skip
before applying this filter)"
    notify-send "ERROR! with settings, target requires an integer in option (b) - (Num. cycles to
skip before applying this filter)"

  else
    sed -i -e "s/targetnumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "target: Num. cycles to skip before applying this filter: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

  fi

ReadFilterSetting=$(awk -F '[]' '{print $18}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt
  if [ "$ReadFilterSetting" = "-" ]

  then
    sed -i -e "s/targetnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
  else
    FilterPartc=$(echo "$ReadFilterSetting" | awk -F ',' '{print $1}')
    FilterPartd=$(echo "$ReadFilterSetting" | awk -F ',' '{print $2}')
    sed -i -e "s/targetnumber_c\+/${FilterPartc} ${FilterPartd}/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "target: While target filtering, filtered/-unfiltered cycles will alternate
(No.Filtered,No.Unfiltered): ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

  fi

fi

ReadFilterSetting=$(awk -F '[]' '{print $19}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt

  if [ "$ReadFilterSetting" = "No" ]
  then
    sed -i -e "s/targetFflag targetFnumber_a targetFnumber_b targetFnumber_c
:FILTERING_PROCESSING_ASSEMBLED_CONTIGS_END \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
  else
    sed -i -e "s/:FILTERING_PROCESSING_ASSEMBLED_CONTIGS_END \+//g"
~/GGOSS/Scripts/PriceTiRun.sh
    sed -i -e "s/targetFflag\+/-targetF/g" ~/GGOSS/Scripts/PriceTiRun.sh
    echo "Filtering/Processing Assembled Contigs: targetF: Yes" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

  ReadFilterSetting=$(awk -F '[]' '{print $20}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt
  if [ "$ReadFilterSetting" = "-" ]

```

```

then
sed -i -e "s/targetFnumber_a \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, targetF requires an integer in option (a) - (% identity to an input
initial contig to count as a match (ungapped))"
notify-send "ERROR! with settings, targetF requires an integer in option (a) - (% identity to an
input initial contig to count as a match (ungapped))"

else
sed -i -e "s/targetFnumber_a\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "targetF: % identity to an input initial contig to count as a match (ungapped):
${ReadFilterSetting}" >> ~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $21}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/targetFnumber_b \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "ERROR! with settings, targetF requires an integer in option (b) - (Num. cycles to skip
before applying this filter)"
notify-send "ERROR! with settings, targetF requires an integer in option (b) - (Num. cycles to
skip before applying this filter)"

else
sed -i -e "s/targetFnumber_b\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "targetF: Num. cycles to skip before applying this filter: ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi

ReadFilterSetting=$(awk -F '[' '{print $22}'
~/GGOSS/tmp/PRICETI_FilterOtherSettingsChange.txt)
if [ "$ReadFilterSetting" = "-" ]

then
sed -i -e "s/targetFnumber_c \+//g" ~/GGOSS/Scripts/PriceTiRun.sh
else
FilterPartc=$(echo "$ReadFilterSetting" | awk -F ',' '{print $1}')
FilterPartd=$(echo "$ReadFilterSetting" | awk -F ',' '{print $2}')
sed -i -e "s/targetFnumber_c\+/${FilterPartc} ${FilterPartd}/g" ~/GGOSS/Scripts/PriceTiRun.sh
sed -i -e "s/targetFnumber_c\+/${ReadFilterSetting}/g" ~/GGOSS/Scripts/PriceTiRun.sh
echo "targetF: While target filtering, filtered/-unfiltered cycles will alternate
(No.Filtered,No.Unfiltered): ${ReadFilterSetting}" >>
~/GGOSS/tmp/YadWindow4SelectedSettings.txt

fi
fi

pkill yad

```

#automatically open logfile upon completion

yad --title="GENOME SEQUENCING PROGRAM -- PRICETI Created by Giles Holt" -
-timeout=8 --no-buttons --no-escape --length=100 --width=400 --center --text="

###---ABOUT TO RUN PRICETI---###

Please check the PRICETI Logfile to ascertain and confirm

the success of PRICETI run on your samples. You can

also find other information there, including the version of PRICETI that was used

"

#sorts them so they are in order, as R1 and R2 names are identical except for the 1 or 2, they will
be sure to be grouped together, allowing to select them by the line and the line +1

sort ~/GGOSS/tmp/PRICETISelectedFilesP4.txt

sort ~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt

sort ~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt

PathToFiles=\$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

sed -i -e "s|FilePath|\${PathToFiles}|g" ~/GGOSS/Scripts/PriceTiRun.sh

#additional input 1, set path if input has been selected

InputFileType=\$(awk -F '[' '{print \$2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if ["\$InputFileType" != "Not Applicable"];then

FileLocationOrFileName=\$(awk 'NR==1 {print}'

~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt)

if ["\$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes"] || [

"\$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet"] || ["\$FileLocationOrFileName" =
~/GGOSS_InputOutput/IDBA"];then

PathToFiles1Prep_a=\$(awk 'NR==1 {print}' ~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt)

else

PathToFiles1=\$(awk 'NR==1 {print}' ~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt)

sed -i -e "s|File2Path|\${PathToFiles1}|g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

fi

#additional input 2, set path if input has been selected - if assembly file is selected then prep for the
path, as it changes with every file

InputFileType=\$(awk -F '[' '{print \$3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if ["\$InputFileType" != "Not Applicable"];then

FileLocationOrFileName=\$(awk 'NR==1 {print}'

~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt)

if ["\$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes"] || [

"\$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet"] || ["\$FileLocationOrFileName" =
~/GGOSS_InputOutput/IDBA"];then

PathToFiles2Prep_a=\$(awk 'NR==1 {print}' ~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt)

```

else
PathToFiles2=$(awk 'NR==1 {print}' ~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt)
sed -i -e "s|File3Path|${PathToFiles2}|g" ~/GGOSS/Scripts/PriceTiRun.sh
fi
fi

ReadFileType=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
if [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "fpp" ] || [ "$ReadFileType" = "mp" ] || [
"$ReadFileType" = "mpp" ]
then

##### As using forward and reverse it calculates the actual number of run throughs
NumberOfSamplesToRunPrep=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
NumberOfSamplesToRun=$(echo "$NumberOfSamplesToRunPrep / 2" | bc)

else
NumberOfSamplesToRun=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
fi

Settings=$(cat ~/GGOSS/tmp/YadWindow4SelectedSettings.txt)

Time1min=$(date +%M")
Time1hourtmp=$(date +%H")

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

StartTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

Addit1_line=1
Addit2_line=1
line=1
for i in $(seq 1 $NumberOfSamplesToRun)
do

Time2min=$(date +%M")
Time2hourtmp=$(date +%H")

Time2hour=$( echo "$Time2hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time2hour + $Time2min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberFilesLeftToDo=$( echo "$NumberOfSamplesToRun - $line" | bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $line) * $NumberOfFilesLeftToDo" | bc )

AverageTimeTakenPerSample=$( echo "($TimeTaken / $line)" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

PercentComplete=$( echo "scale=2; ($line / $NumberOfSamplesToRun) * 100" | bc )

```

```

#adapts to whether it needs 1 or 2 input files (read 1 and read 2)

ReadFileType=$(awk -F '[]' '{print $4}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
if [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "fpp" ] || [ "$ReadFileType" = "mp" ] || [
"$ReadFileType" = "mpp" ]
    then
lineR2=$(( $line + 1 ))
filenameR1=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
filenameR2=$(awk -v y=$lineR2 'NR==y {print}' ~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
    else
filenameR1=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/PRICETISelectedFilesP4.txt)
    fi

#if additional file type 1 selected then: make file name for it like above, from file selection, and edit
run script
InputFileType=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if [ "$InputFileType" != "Not Applicable" ];then
    FileLocationOrFileName=$(awk 'NR==1 {print}'
~/GGOSS/tmp/AdditionalInput1Path2selectedFile.txt)
    if [ "$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes" ] || [
"$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet" ] || [ "$FileLocationOrFileName" =
"~/GGOSS_InputOutput/IDBA" ];then

        #set file name as contigs or scaffolds
        AdditionalInput1Filename=$(awk -F '|' 'NR==1 {print $1}'
~/GGOSS/tmp/PRICETI_ContigOrScaffold1.txt)

        #finalise file path
        PathToFiles1Prep_b=$(awk -v x=$Addit1_line 'NR==x {print}'
~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt)
        PathToFiles1=$(echo "$PathToFiles1Prep_a/$PathToFiles1Prep_b")
        sed -i -e "s|File2Path|${PathToFiles1}|g" ~/GGOSS/Scripts/PriceTiRun.sh

    else
        AdditionalInput1Filename=$(awk -v x=$Addit1_line 'NR==x {print}'
~/GGOSS/tmp/PRICETIAdditionalInput1SelectedFilesP4.txt)

    fi
sed -i -e "s/File2Name1\+/$AdditionalInput1Filename/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

#if additional file type 2 selected then: make file name for it like above, from file selection
InputFileType=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
if [ "$InputFileType" != "Not Applicable" ];then
    FileLocationOrFileName=$(awk 'NR==1 {print}'
~/GGOSS/tmp/AdditionalInput2Path2selectedFile.txt)
    if [ "$FileLocationOrFileName" = "~/GGOSS_InputOutput/SPAdes" ] || [
"$FileLocationOrFileName" = "~/GGOSS_InputOutput/Velvet" ] || [ "$FileLocationOrFileName" =
"~/GGOSS_InputOutput/IDBA" ];then

        #set file name as contigs or scaffolds
        AdditionalInput2Filename=$(awk -F '|' 'NR==1 {print $1}'
~/GGOSS/tmp/PRICETI_ContigOrScaffold2.txt)

```

```

#finalise file path
PathToFiles2Prep_b=$(awk -v x=$Addit1_line 'NR==x {print}'
~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt)
PathToFiles2=$(echo "$PathToFiles2Prep_a/$PathToFiles2Prep_b")
sed -i -e "s|File3Path|${PathToFiles2}|g" ~/GGOSS/Scripts/PriceTiRun.sh

else
AdditionalInput2Filename=$(awk -v x=$Addit1_line 'NR==x {print}'
~/GGOSS/tmp/PRICETIAdditionalInput2SelectedFilesP4.txt)

fi
sed -i -e "s/File3Name1\+/$AdditionalInput2Filename/g" ~/GGOSS/Scripts/PriceTiRun.sh
fi

#this will only set the filename correctly if there is an underscore after the filename
filename=$(echo "$filenameR1" | awk -F '_' '{NF-=4; OFS="_"; print}')

if [ -f ~/GGOSS_InputOutput/PRICETI/PRICETI_$filename ]
then
echo "PRICETI output file for $filename already exists, if you wish to re-run this samples
please delete or move its pre-existing PRICETI output. Running the next sample
"
echo | notify-send "PRICETI output file for $filename already exists, if you wish to re-run
this samples please delete or move its pre-existing PRICETI output. Running the next sample
"
fi

if [ "$ReadFileType" = "fp" ] || [ "$ReadFileType" = "fpp" ] || [ "$ReadFileType" = "mp" ] || [
"$ReadFileType" = "mpp" ]
then
#change the PRICETI run script for the filename at hand
#change file 1
sed -i -e "s/FileName1\+/$filenameR1/g" ~/GGOSS/Scripts/PriceTiRun.sh

#change file 2
sed -i -e "s/FileName2\+/$filenameR2/g" ~/GGOSS/Scripts/PriceTiRun.sh

#change output file
sed -i -e "s/PriceTI_Filename\+/PRICETI_$filename/g" ~/GGOSS/Scripts/PriceTiRun.sh

#####

yad --title="GGOSS -- PRICETI" --width=700 --center --sticky --on-top --no-buttons --no-
escape --text-align=center --text=" Running PRICETI

${PercentComplete}% COMPLETE

Total number of samples for PRICETI to run: $NumberOfSamplesToRun

Currently running PRICETI on file: ${filename}. Started at:${Time2hourtmp}:${Time2min}

Time taken thus far: ${TimeTaken} minutes
Average time taken per sample:$AverageTimeTakenPerSample

```

Estimated time left until completion:\$EstimatedtimetoFinishMins minutes
(\${EstimatedtimetoFinishHours} hours)

Settings:
\$Settings
" &

~/GGOSS/Scripts/PriceTiRun.sh

pkill yad

#change back ready for the next file type

#change file 1

sed -i -e "s/\${filenameR1}\+/\${FileName1}/g" ~/GGOSS/Scripts/PriceTiRun.sh

#change file 2

sed -i -e "s/\${filenameR2}\+/\${FileName2}/g" ~/GGOSS/Scripts/PriceTiRun.sh

#change output file

sed -i -e "s/PRICETI_\${filename}\+/\${PriceTI_FileName}/g" ~/GGOSS/Scripts/PriceTiRun.sh

#if additional file type 1 selected then: change file name back

InputFileType=\$(awk -F '[]' '{print \$2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if ["\$InputFileType" != "Not Applicable"];then

echo "testing"

sed -i -e "s/\${AdditionalInput1Filename}\+/\${File2Name1}/g" ~/GGOSS/Scripts/PriceTiRun.sh

sed -i -e "s|\${PathToFiles1}|File2Path|g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

#if additional file type 2 selected then: change file name back

InputFileType=\$(awk -F '[]' '{print \$3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)

if ["\$InputFileType" != "Not Applicable"];then

echo "testing"

sed -i -e "s/\${AdditionalInput2Filename}\+/\${File3Name1}/g" ~/GGOSS/Scripts/PriceTiRun.sh

sed -i -e "s|\${PathToFiles2}|File3Path|g" ~/GGOSS/Scripts/PriceTiRun.sh

fi

Addit1_line=\$((\$Addit1_line + 1))

Addit2_line=\$((\$Addit2_line + 1))

line=\$((\$line + 2))

else

#change the PRICETI run script for the filename at hand - for single file runs like fs or fsp

#change file 1

sed -i -e "s/\${FileName1}\+/\${filenameR1}/g" ~/GGOSS/Scripts/PriceTiRun.sh

#change output file

sed -i -e "s/\${PriceTI_FileName}\+/\${PRICETI_\${filename}}/g" ~/GGOSS/Scripts/PriceTiRun.sh

#####

```
yad --title="GGOSS -- PRICETI" --width=700 --center --sticky --on-top --no-buttons --no-escape --text-align=center --text="  Running PRICETI
```

```
  ${PercentComplete}% COMPLETE
```

```
  Total number of samples for PRICETI to run: $NumberOfSamplesToRun
```

```
  Currently running PRICETI on file: ${filename}. Started at:${Time2hourtmp}:${Time2min}
```

```
  Time taken thus far: ${TimeTaken} minutes
```

```
  Average time taken per sample:$AverageTimeTakenPerSample
```

```
  Estimated time left until completion:$EstimatedtimetoFinishMins minutes  
  (${EstimatedtimetoFinishHours} hours)
```

```
  Settings:
```

```
  $Settings
```

```
  " &
```

```
  ~/GGOSS/Scripts/PriceTiRun.sh
```

```
  pkill yad
```

```
  #change back ready for the next file type
```

```
  #change file 1
```

```
  sed -i -e "s/${filenameR1}\+\/FileName1/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
  #change output file
```

```
  sed -i -e "s/PRICETI_${filename}\+\/PriceTI_Filename/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
  #if additional file type 1 selected then: change file name back
```

```
  InputFileType=$(awk -F '[]' '{print $2}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
  if [ "$InputFileType" != "Not Applicable" ];then
```

```
  echo "testing"
```

```
    sed -i -e "s/${AdditionalInput1Filename}\+\/File2Name1/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
    sed -i -e "s/${PathToFiles1}|File2Path/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
  fi
```

```
  #if additional file type 2 selected then: change file name back
```

```
  InputFileType=$(awk -F '[]' '{print $3}' ~/GGOSS/tmp/PRICETI_SettingsChange.txt)
```

```
  if [ "$InputFileType" != "Not Applicable" ];then
```

```
  echo "testing"
```

```
    sed -i -e "s/${AdditionalInput2Filename}\+\/File3Name1/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
    sed -i -e "s/${PathToFiles2}|File3Path/g" ~/GGOSS/Scripts/PriceTiRun.sh
```

```
  fi
```

```

#repeat the above - including the additional input files - but using just 1 input rather than two
  Addit1_line=$(( $Addit1_line + 1 ))
  Addit2_line=$(( $Addit2_line + 1 ))
  line=$(( $line + 1 ))
fi

done

} | tee ~/GGOSS/LogFiles/PRICETI_LOGFILE.txt

echo "End of PRICE script"

echo "
"

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- PRICE" --
button="Open file location":5 --button="Return to menu":4 --text="

          PRICE Complete

"

mode="$?"
case $mode in
  4)~/GGOSS/GenomicsProgram.sh ;;
  5)nautilus ~/GGOSS_InputOutput/PRICETI & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.4 Annotation

10.9.3.4.1 GGOSS script for Prokka

```

#!/bin/sh

#still need to build percentage increase for node seperation parts

PathToProkka=$(awk -F '|' '{print $10}' ~/GGOSS/tmp/PathToTools_SettingsChange.txt)
if [ "$PathToProkka" = "N/A" ];then
PathToProkka="$HOME/"
fi

#PROKKA script
{
echo "          GGOSS Genomics - Created by Giles Holt
"
echo "          Prokka run from GGOSS Genomics program
"
echo |(date +"          Prokka Start Date: %d-%m-%y Time: %T

```

```

"
)

#sets number of files selected
NumberOfFilesToRun=$(grep -c -v "ThisIsMyAntiMatch" ~/GGOSS/tmp/ProkkaSelectedFilesP4.txt)

#Calculates the percentage addition per sample
PercentToAddPerSample=$( echo "scale=2; 100 / $NumberOfFilesToRun" | bc )

#sets the file name containing those files
FilePath=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)

echo 1
echo "#Running Prokka... checking if any node specifications have been given...      1%
Complete"
sleep 1
line=1
#loop
for i in $(seq 1 $NumberOfFilesToRun)
do

#sets percentage if only on first file
if [ $line = 1 ];then
Percent=5
fi
##### time calc below #####
Time1min=$(date +%M)
Time1hourtmp=$(date +%H)
echo "Current time: ${Time1hourtmp}:${Time1min}"

Time1hour=$( echo "$Time1hourtmp * 60" | bc )

EndTimeOfDayInMinutes=$( echo "$Time1hour + $Time1min" | bc )

TimeTaken=$( echo "$EndTimeOfDayInMinutes - $StartTimeOfDayInMinutes" | bc )

NumberTimesThrough=1
#this version is for the continued calculation of total time left
if (( ${TimeTaken} >= 1 ))
then
echo "Time taken to complete $NumberTimesThrough run through/s: ${TimeTaken} minutes"
else echo "Time taken to complete $NumberTimesThrough run through/s: < 1 minute"
fi

NumberOfFilesLeftToRun=$( echo "$NumberOfFilesToRun - $NumberTimesThrough" | bc )

EstimatedtimetoFinishMins=$( echo "($TimeTaken / $NumberTimesThrough) *
$NumberOfFilesLeftToRun" | bc )

EstimatedtimetoFinishHours=$( echo "$EstimatedtimetoFinishMins / 60" | bc )

NumberOfFilesLeftToRun1=$( echo "$NumberOfFilesToRun - $NumberTimesThrough" | bc )

```

```

EstimatedtimetoFinishMins1=$( echo "($TimeTaken / $NumberTimesThrough)" | bc )

EstimatedtimetoFinishHours1=$( echo "$EstimatedtimetoFinishMins1 / 60" | bc )

if [ "${line}" = "1" ];then

EstimatedtimetoFinishMins="Still calculating..."
EstimatedtimetoFinishHours="Still calculating..."

EstimatedtimetoFinishMins1="Still calculating..."
EstimatedtimetoFinishHours1="Still calculating..."

fi

##### Time calc above #####

#finds the first file name
FileName=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/ProkkaSelectedFilesP4.txt)

FileType=$(awk -F '[' '{print $1}' ~/GGOSS/tmp/ProkkaSettingsChange.txt)
#informs the user which file is being run
yad --title="GILES -- Annotation - Prokka" --width=400 --left --sticky --on-top --no-buttons --no-escape --text-align=center --text="          Running Prokka on file: $FileName

          File type selected:$FileType

Estimated time remaining for ${FileName}: $EstimatedtimetoFinishMins1 minutes
(${EstimatedtimetoFinishHours1} hours)

Estimated time remaining until completion of all samples: $EstimatedtimetoFinishMins minutes
(${EstimatedtimetoFinishHours} hours)

" &

##### ----- NODE SORTING IF REQUIRED -----
#####

WholeFileOrNode=$(awk -F '[' '{print $2}' ~/GGOSS/tmp/ProkkaSettingsChange.txt)

if [ "$WholeFileOrNode" != "Run file type as a whole" ]
then

NumberOfNodes=$(awk -F '[' '{print $3}' ~/GGOSS/tmp/ProkkaSettingsChange.txt)
MinimumNodeLength=$(awk -F '[' '{print $4}' ~/GGOSS/tmp/ProkkaSettingsChange.txt)
MinimumCoverage=$(awk -F '[' '{print $5}' ~/GGOSS/tmp/ProkkaSettingsChange.txt)

##### -- If NumberOfNodes != "Not applicable"; then cut nodes by number -- #####

if [ "$NumberOfNodes" != "Not applicable" ]
then
#makes sure it gets the line number of the first one that doesnt fit the criteria

```

```

MaximumNodeAmount=$(echo "$NumberOfNodes + 1" | bc)

#make list of line numbers and node info for each node line, as a variable. #take the first $... number
of lines from the list. #make the last in the new list a variable. #Line Number to cut file to
CutOffLineNumberPrep=$(grep -n 'NODE' $FilePath/$FileName/$FileType | head -n
"$MaximumNodeAmount" | tail -1 | awk -F ':' '{print $1}')

#CutOffLineNumberPrep=$(grep -n 'NODE'
~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta | head -n "$MaximumNodeAmount" | tail
-1 | awk -F ':' '{print $1}')

CutOffLineNumber=$( echo "$CutOffLineNumberPrep - 1" | bc )

#Keep the lines before that variable number in the contig or scaffold file
head -n "$CutOffLineNumber" ~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta >
~/GGOSS_InputOutput/Prokka/Ncut${NumberOfNodes}_${FileName}.fasta

#rename the file to include how much cut by and reset the file variable name to it - in prep for the
prokka run
FilePath="$HOME/GGOSS_InputOutput"
FileName="Prokka"
FileType="Ncut${NumberOfNodes}_${FileName}.fasta"

    fi

##### -- if NodeLength != "Not applicable"; then cut nodes by size --
#####

if [ "$MinimumNodeLength" != "Not applicable" ]
then

#make list of line number for each node line as a variable
NodeTitleLineNumberList=$(grep -n 'NODE' $FilePath/$FileName/$FileType)

#NodeTitleLineNumberList=$(grep -n 'NODE'
~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta)

#variable of the contig sizes, trimmed to those containing large enough contigs, and presented as the
line number they're found on
LineNumberForContigLengthCutOffPrep=$(echo "$NodeTitleLineNumberList" | awk -F '_' -v
x=$MinimumNodeLength '(NR==1) || ($4 > x)' | tail -1 | awk -F ':' '{print $1}')

LineNumberForContigLengthCutOff=$(echo "$LineNumberForContigLengthCutOffPrep - 1" | bc)

head -n "$LineNumberForContigLengthCutOff" $FilePath/$FileName/$FileType >
$FilePath/$FileName/Lcut${MinimumNodeLength}_${FileType}

rm $FilePath/$FileName/$FileType

FilePath="$HOME/GGOSS_InputOutput"
FileName="Prokka"
FileType="Lcut${MinimumNodeLength}_${FileType}"

    fi

```

```

##### -- filter by coverage -- #####

if [ "$MinimumCoverage" != "Not applicable" ]
then

#make list of line number for each node line as a variable
NodeTitleLineNumberList=$(grep -n 'NODE' $FilePath/$FileName/$FileType)

# add a column where each line is numbered
NewListNumbered=$(echo "$NodeTitleLineNumberList" | grep -n '^')

#NodeTitleLineNumberList=$(grep -n 'NODE'
~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta)

#variable of the contig coverages, trimmed to those containing high enough coverage, and presented
as the line number they're found on
LineNumberForCoverageCutOffPrep=$(echo "$NewListNumbered" | awk -F ' ' -v
x=$MinimumCoverage '(NR==1) || ($6 > x)' | awk -F ':' '{print $1}')

if [ -f ~/GGOSS/tmp/TempBuildContigFile.txt ]
then
touch ~/GGOSS/tmp/TempBuildContigFile.txt
fi

#find each line from LineNumberForCoverageCutOffPrep in NodeTitleLineNumberList and cut to
the line number of the node listed below it
NumberOfLinesInFilteredList=$(echo "$LineNumberForCoverageCutOffPrep" | grep -v -c
"ThisIsMyAntiMatch")
list=1
for i in $(seq 1 $NumberOfLinesInFilteredList)
do
#takes a node starting linenummer from list of those with acceptable coverage
LineToFind=$(echo "$LineNumberForCoverageCutOffPrep" | awk -v y=$list 'NR==y {print}')
LineToCutToPrep=$(( $LineToFind + 1 ))

LineToCutFrom=$(echo "$NodeTitleLineNumberList" | awk -v x=$LineToFind 'NR==x {print}' |
awk -F ':' '{print $1}')
LineToCutTo=$(echo "$NodeTitleLineNumberList" | awk -v x=$LineToCutToPrep 'NR==x {print}' |
awk -F ':' '{print $1}')

#use the above to cut out the node that fits criteria
cat $FilePath/$FileName/$FileType | awk -v x=$LineToCutFrom -v y=$LineToCutTo 'NR >= x &&
NR < y {print}' >> ~/GGOSS/tmp/TempBuildContigFile.txt

sed -n -e "$LineToCutFrom,$LineToCutTo p" -e "$LineToCutTo q"
~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta
test=$(tail -n +$LineToCutFrom ~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta | head -
n $((LineToCutTo-LineToCutFrom+1)))
NodeThatFitsCriteria=$(cat ~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta | awk -v
x=$LineToCutFrom -v y=$LineToCutTo 'NR >= x && NR < y {print}')
test=$(< ~/GGOSS_InputOutput/SPAdes/SPAdes_SOO/contigs.fasta tail -n +"$LineToCutFrom" |
head -n "$((LineToCutTo - LineToCutFrom))")

```

```

cat ~/GGOSS_InputOutput/SPAdes/SPAdes_SO0/contigs.fasta | awk -v x=$LineToCutFrom -v
y=$LineToCutTo 'NR >= x && NR < y {print}' > ~/test

awk -v y=$list 'NR==y {print}'
awk -v y=$lineToFind 'NR==y {print}'
LineToCutToPrep1=$(echo "$NodeTitleLineNumberList" | grep -n "$lineToFind")

LineToCutToPrep2=$(echo "$LineToCutToPrep1 + 1" | bc | awk -F ':' '{print $1}')

LineToCutTo=$(echo "$LineToCutToPrep2" | awk -F ':' '{print $1}')

list=$(( $list + 1 ))
done

#rename ~/GGOSS/tmp/TempBuildContigFile.txt to ... the samples name plus extras
mv ~/GGOSS/tmp/TempBuildContigFile.txt
$FilePath/$FileName/Ccut${MinimumCoverage}_$FileType

    $FilePath/$FileName/$FileType > $FilePath/$FileName/Ccut${MinimumCoverage}_$FileType

rm $FilePath/$FileName/$FileType

FilePath="$HOME/GGOSS_InputOutput"
FileName="Prokka"
FileType="Ccut${MinimumCoverage}_$FileType"

fi

awk 'BEGIN {cnt=1} /^>NODE_/ { gsub("_.*$", "_cnt++,$0") } { print}'
$FilePath/$FileName/$FileType > ~/GGOSS_InputOutput/Prokka/${FileName}_NNCut.fasta

cd ~/GGOSS_InputOutput/Prokka && ${PathToProkka}prokka --prefix Prokka_${FileName}
${FileName}_NNCut.fasta

pkill yad

    yad --title="GILES -- Annotation - Prokka" --width=400 --center --sticky --on-top --no-buttons --
no-escape --text-align=center --timeout=2 --text="        ${FileName} sample is complete
"

pkill yad

else

##### ----- run prokka on samples as a whole -----
-- #####

#cuts the names of the nodes in the contigs.fasta file, so that it meets the 20 character limit prokka has
awk 'BEGIN {cnt=1} /^>NODE_/ { gsub("_.*$", "_cnt++,$0") } { print}'
$FilePath/$FileName/$FileType > ~/GGOSS_InputOutput/Prokka/${FileName}_NNCut.fasta

#sorts prokka environment
cd ~/ .bashrc
export PATH=$PATH:$HOME/prokka-1.11/bin

```

```

echo ${Percent}
echo "#Running Prokka... Running file (${FileName}) as whole      ${Percent}% Complete"

  cd ~/GGOSS_InputOutput/Prokka && prokka --prefix Prokka_${FileName}
  ${FileName}_NNCut.fasta

  Percent=$( echo "scale=2; $Percent + $PercentToAddPerSample" | bc )

echo ${Percent}
echo "#Running Prokka... Complete file (${FileName}) as whole      ${Percent}% Complete"
sleep 1

fi

rm ~/GGOSS_InputOutput/Prokka/${FileName}_NNCut.fasta

NumberTimesThrough=$(( $NumberTimesThrough + 1 ))
line=$(( $line + 1 ))
done

} | yad --progress --auto-close --auto-kill --center --width=700 --image=$ICON --image-on-top --
title="GENOME SEQUENCING PROGRAM -- Prokka      GGOSS created by Giles Holt" -
-text="Running Prokka

Settings:
$Settings
"

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- Prokka" --
button="Open file location":5 --button="Return to menu":4 --text="

                Prokka Complete
"

mode="$?"
  case $mode in
    4)~/GGOSS/GenomicsProgram.sh ;;
    5)nautilus ~/GGOSS_InputOutput/Prokka & ~/GGOSS/GenomicsProgram.sh ;;
  esac

```

10.9.3.5 Community analysis

10.9.3.5.1 Mothur

10.9.3.5.1.1 Stability file creation

```

#!/bin/bash
#empty stability file
touch ~/GGOSS/tmp/StabilityFile_FirstColumn.txt
#empty stability file 2

```

```

touch ~/GGOSS/tmp/StabilityFile_firstAndSecondColumn.txt
#empty stability file 3
touch ~/GGOSS/tmp/StabilityFile_thirdColumn.txt
touch ~/GGOSS/tmp/Stabilityfile_complete.txt
#organises by the second column through to the 3rd column - this means they are arranged by
S${File} etc
sort -t_ -k2,3 ~/GGOSS/tmp/SelectFile.txt | tee ~/GGOSS/tmp/MothurSelectedFiles1.txt

cp ~/GGOSS/tmp/MothurSelectedFiles1.txt ~/GGOSS/tmp/SelectFile.txt

rm ~/GGOSS/tmp/MothurSelectedFiles1.txt

NumberOfFiles=$(ls ~/GGOSS/tmp/SelectFile.txt | grep -v -c "ThisIsMyAntiMatch")
File=1
for i in (seq $NumberOfFiles);do

#if file is there then run: everything below
FILE=~/"GGOSS/tmp/SelectFile.txt"
STRING="S${File}_L001_R1_001.fastq"
if [ ! -z $(grep "$STRING" "$FILE") ]
then (

#copy first line from mothur selected

head -n1 /home/giles/GGOSS/tmp/SelectFile.txt >
/home/giles/GGOSS/tmp/StabilityFile_FirstColumn.txt

#remove everything from and including _S${File}_L001_R1_001.fastq. #tab line 1 after last text

perl -pi -e 's/_S${File}_L001_R1_001.fastq/ /g' ~/GGOSS/tmp/StabilityFile_FirstColumn.txt

VAR=$(head -n1 /home/giles/GGOSS/tmp/StabilityFile_FirstColumn.txt)
VAR1=$(head -n1 /home/giles/GGOSS/tmp/SelectFile.txt)

#copy first line from mothur selected and paste after a tab

export VAR VAR1
awk '
1
$0 == ENVIRON["VAR"] {print ENVIRON["VAR1"]}
' ORS=' ' ~/GGOSS/tmp/StabilityFile_FirstColumn.txt >
~/GGOSS/tmp/StabilityFile_firstAndSecondColumn.txt

#tab line 1 after the last text
#copy second line from mothur selected and paste after that tab

touch ~/GGOSS/tmp/StabilityFile_FirstColumn.txt
VAR=$(head -n1 /home/giles/GGOSS/tmp/StabilityFile_firstAndSecondColumn.txt)

#makes a file containing just the second line from mothur selected
sed -n '2{p;q;}' ~/GGOSS/tmp/SelectFile.txt > ~/GGOSS/tmp/StabilityFile_thirdColumn.txt
#sets variable as the line selected out and put in new file)
VAR1=$(head -n1 ~/GGOSS/tmp/StabilityFile_thirdColumn.txt)

#put the two variable lines together

```

```

export VAR VAR1
awk '
  1
  $0 == ENVIRON["VAR"] {print ENVIRON["VAR1"]}
' ORS=' ' ~/GGOSS/tmp/StabilityFile_firstAndSecondColumn.txt >
~/GGOSS/tmp/Stabilityfile_complete.txt
)
fi

File=$(( $File + 1 ))
done

```

10.9.3.5.1.2 GGOSS run script for Mothur

```

#!/bin/sh

{
echo "          GGOSS Genomics - Created by Giles Holt"
"
echo "          Mothur run from GGOSS Genomics program"
"
echo | (date +"          Mothur Start Date: %d-%m-%y Time: %T"
"
)

echo ""

#automatically open logfile upon completion, get the progress bar to work

yad --title="GENOME SEQUENCING PROGRAM -- Mothur          Created by Giles Holt"
--timeout=10 --no-buttons --no-escape --text="Please check the Mothur Logfile to ascertain and
confirm the success of Mothur analysis of your samples. You can
also find other information there, including the version of Mothur that was used
" &

echo "Creating Stability file"

~/GGOSS/Scripts/StabilityFileAllFastq.sh

rm ~/GGOSS_InputOutput/Mothur/

#move stability file based on all samples to mothur file

cp ~/GGOSS/tmp/Stabilityfile_complete.txt ~/GGOSS/mothur/stability.files

#check mothur settings for processor number, if empty check processor number on computer and put
in that text file

```

```

#move files to mothur
cp ~/GGOSS_InputOutput/FastqFiles/ITSfastq/*.fastq ~/GGOSS/mothur/

cd ~/GGOSS/mothur

#doesn't work mothur: "#set.dir(output=~ /GGOSS/Output/Mothur)"

#make.contigs(file=stability.files, processors=4); summary.seqs(fasta=stability.trim.contigs.fasta);
screen.seqs(fasta=stability.trim.contigs.fasta, group=stability.contigs.groups, maxambig=0,
maxlength=275); summary.seqs(); unique.seqs(fasta=stability.trim.contigs.good.fasta);

mothur "#makecontigs; summaryseqs; screensseqs; summaryseqs; uniqueseqs;
count.seqs(name=stability.trim.contigs.good.names, group=stability.contigs.good.groups,
processors=1); summary.seqs(count=stability.trim.contigs.good.count_table, processors=4);
align.seqs(fasta=stability.trim.contigs.good.unique.fasta, reference=silva.v4.fasta);
summary.seqs(fasta=stability.trim.contigs.good.unique.align,
count=stability.trim.contigs.good.count_table);
screen.seqs(fasta=stability.trim.contigs.good.unique.align,
count=stability.trim.contigs.good.count_table, summary=stability.trim.contigs.good.unique.summary,
start=1968, end=11550, maxhomop=8); summary.seqs(fasta=current, count=current);
filter.seqs(fasta=stability.trim.contigs.good.unique.good.align, vertical=T, trump=.);
unique.seqs(fasta=stability.trim.contigs.good.unique.good.filter.fasta,
count=stability.trim.contigs.good.good.count_table);
pre.cluster(fasta=stability.trim.contigs.good.unique.good.filter.unique.fasta,
count=stability.trim.contigs.good.unique.good.filter.count_table, diffs=3);
chimera.uchime(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.count_table, dereplicate=t,
processors=1)
remove.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
accnos=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.accnos);
summary.seqs(fasta=current, count=current, processors=4)
classify.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.pick.count_table,
reference=trainset9_032012.pds.fasta, taxonomy=trainset9_032012.pds.tax, cutoff=70)
remove.lineage(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.taxonomy,
taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)
dist.seqs(fasta=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.fasta,
cutoff=0.20, processors=4);
cluster(column=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.dist,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.pick.pick.count_table);
make.shared(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.an.unique_list.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.pick.pick.count_table, label=0.03);
classify.otu(list=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.an.unique_list.list,
count=stability.trim.contigs.good.unique.good.filter.unique.precluster.denovo.uchime.pick.pick.count_table,
taxonomy=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pds.wang.pick.taxono

```

```
my, label=0.03);
count.groups(shared=stability.trim.contigs.good.unique.good.filter.unique.precluster.pick.pick.an.uniq
ue_list.shared)"
```

```
#copy the files to the accessible Output
cp ~/GGOSS/mothur/stability.* ~/GGOSS_InputOutput/Mothur
```

```
#remove the files from the mothur primary build location
rm ~/GGOSS/mothur/stability.*
```

```
echo | (date +"
                Mothur Finish Date: %d-%m-%y Time: %T
"
)
} | tee ~/GGOSS/LogFiles/Mothur_LOGFILE.txt
```

10.9.3.5.2 MEGAN

```
#!/bin/sh
rm ~/GGOSS/LogFiles/MEGAN_ViralTaxonomy_LOGFILE.txt
File=1
NumberOfFiles=$(ls ~/ncbi-blast-2.3.0+/db/BlastOutput/ | grep -c -v "ThisIsMyAntiMatch")

for i in $(seq $NumberOfFiles);do
if [ -e ~/GGOSS/Output/MEGAN_output/S${File}_blastn_output.rma6 ];then
    echo | notify-send "MEGAN already complete for S${File}. Trying next sample"
elif [ -e ~/ncbi-blast-2.3.0+/db/BlastOutput/S${File}_blastn_output.txt ];then
    echo | notify-send "Running MEGAN Viral Taxonomy"
else
    echo | notify-send "Blast S${File} file not present Trying next sample"]
fi
echo | script ~/GGOSS/LogFiles/MEGAN_ViralTaxonomy_LOGFILE.txt

if [ -e ~/GGOSS/Output/MEGAN_output/S${File}_blastn_output.rma6 ];then
    echo MEGAN already complete for S${File}. Trying next sample
elif [ -e ~/ncbi-blast-2.3.0+/db/BlastOutput/S${File}_blastn_output.txt ];then
    echo | /usr/local/bin/MEGAN -g -E -v -c
~/GGOSS/MEGAN_Com_txt_files/MEGANcommand${File}.txt
else
    echo ["Blast S${File} file not present Trying next sample"]
fi
File=$(( $File + 1 ))
done

echo | exit

notify-send "MEGAN taxonomy complete by Giles Holt Genomic analysis program"
```

~/GGOSS/GenomicsProgram.sh

10.9.3.5.3 PIPITS

```
#!/bin/sh
```

```
#The PIPITS pipeline is divided into four parts:
```

```
# PIPITS_GETREADPAIRSLIST & PIPITS_PREP: prepares raw reads from Illumina MiSeq sequencers for ITS extraction
```

```
# PIPITS_FUNITS: extracts fungal ITS regions from the reads
```

```
# PIPITS_PROCESS: analyses the reads to produce Operational Taxonomic Unit (OTU) abundance tables and the RDP taxonomic assignment table for downstream analysis
```

```
#####-----  
#####
```

```
#####-----Core PIPITS run -----  
#####
```

```
#####-----  
#####
```

```
{  
#Copy selected samples to the PIPITS file in GOSS_InputOutput  
PathToFiles=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt | awk -F '/' '{print $3}')  
NumberOfFilesSelected=$(grep -v -c "ThisIsMyAntiMatch"  
~/GGOSS/tmp/PIPITSSelectedFilesP4.txt)
```

```
#Output file name, with any whitespace removed  
OutputFileName=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/PIPITS_SettingsChange.txt | tr -d '[:space:]')
```

```
echo 1  
echo "#Running PIPITS... Experiment: $OutputFileName 1% Complete"  
mkdir ~/GGOSS_InputOutput/PIPITS/${OutputFileName}  
mkdir ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_prep  
mkdir ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_funits
```

```
echo 2  
echo "#Identifying and sorting files for PIPITS... Experiment: $OutputFileName 2% Complete"
```

```
rm ~/GGOSS_InputOutput/PIPITS/FastqFilesForRun/*  
line=1  
for i in $(seq 1 $NumberOfFilesSelected);do  
FileToCopy=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/PIPITSSelectedFilesP4.txt)  
cp ~/GGOSS_InputOutput/$PathToFiles/$FileToCopy  
~/GGOSS_InputOutput/PIPITS/FastqFilesForRun/${FileToCopy}  
line=$(( $line + 1 ))  
done
```

```

echo 10
echo "#Running PIPITS... Experiment: $OutputFileName          10% Complete"

##### ---- PIPITS_GETREADPAIRSLIST & PIPITS_PREP:
MaxMemoryNumber=$(awk -F '|' '{print $3}' ~/GGOSS/tmp/PIPITS_SettingsChange.txt)
ITStype=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/PIPITS_SettingsChange.txt)
SingleOrPaired=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/PIPITS_SettingsChange.txt | tr -d '[:space:]')

if [ "$SingleOrPaired" = "Singleread" ];then
PIPITSpairedORsingleReadListCommand="pipits_getreadsingleslist"
PIPITSpairedORsingleReadPrepCommand="pipits_prep_single"
else
PIPITSpairedORsingleReadListCommand="pipits_getreadpairlist"
PIPITSpairedORsingleReadPrepCommand="pipits_prep"
fi

#pipits_getreadpairlist      -i      ~/GGOSS_InputOutput/PIPITS/FastqFilesForRun/      -o
~/GGOSS_InputOutput/PIPITS/test1/Pairedreadlist.txt

echo 11
echo "#Running PIPITS $SingleOrPaired list creation... Experiment: $OutputFileName          11%
Complete"
$PIPITSpairedORsingleReadListCommand -i ~/GGOSS_InputOutput/PIPITS/FastqFilesForRun/ -o
~/GGOSS_InputOutput/PIPITS/${OutputFileName}/${SingleOrPaired}list.txt
echo 12
echo "#Complete PIPITS $SingleOrPaired list creation... Experiment: $OutputFileName          12%
Complete"
sleep 1
#Once we have the list file, we can then begin to process the sequences:
echo 13
echo "#Running PIPITS sequence processing... Experiment: $OutputFileName          13% Complete"
$PIPITSpairedORsingleReadPrepCommand -i ~/GGOSS_InputOutput/PIPITS/FastqFilesForRun/ -o
~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_prep      -l
~/GGOSS_InputOutput/PIPITS/${OutputFileName}/${SingleOrPaired}list.txt
echo 25
echo "#Complete PIPITS sequence processing... Experiment: $OutputFileName          25%
Complete"
sleep 1

##### ---- PIPITS_FUNITS

#The output from PIPITS PREP is taken as an input for this step. It is also mandatory to provide the
script with which ITS subregion (i.e. ITS1 or ITS2) is to be extracted:
echo 26
echo "#Running PIPITS subregion extraction... Experiment: $OutputFileName          26% Complete"
pipits_funits -i ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_prep/prepped.fasta -o
~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_funits -x $ITStype
echo 60
echo "#Complete PIPITS subregion extraction... Experiment: $OutputFileName          60%
Complete"
sleep 1
##### ---- PIPITS_PROCESS
echo 61

```

```

echo "#Running PIPITS clustering and assigning of taxonomy to OTUs... Experiment:
$OutputFileName      61% Complete"
pipits_process -i ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_funits/ITS.fasta -o
~/GGOSS_InputOutput/PIPITS/${OutputFileName}/out_process --Xmx ${MaxMemoryNumber}G
echo 90
echo "#Complete PIPITS clustering and assigning of taxonomy to OTUs... Experiment:
$OutputFileName      90% Complete"
sleep 1

#####-----
#####

####----- Other features -----#####

#####-----
#####

#edit the table for running through another analysis tool for assigning functional information to OTU's
(FUNguild analysis)
FUNGuildAnalysisTable=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/PIPITS_SettingsChange.txt)
if [ "$FUNGuildAnalysisTable" = "Yes" ];then
mkdir ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_process
pipits_funguild.py -i ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_process/otu_table.txt
-o ~/GGOSS_InputOutput/PIPITS/${OutputFileName}/pipits_process/otu_table_funguild.txt
fi

echo 99
echo "#Complete PIPITS On Experiment: $OutputFileName      100% Complete"
sleep 2

} | tee ~/GGOSS/LogFiles/PIPITS_LOGFILE.txt | yad --progress --auto-close --auto-kill --center --
width=700 --image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM --
PIPITS      GGOSS created by Giles Holt" --text="Running PIPITS

Settings:
$Settings
"

yad --auto-close --auto-kill --center --width=300 --image-on-top --title="GGOSS -- PIPITS" --
button="Open file location":5 --button="Return to menu":4 --text="

      PIPITS Complete

"

mode="$?"
case $mode in
4)~/GGOSS/GenomicsProgram.sh ;;
5)nautilus ~/GGOSS_InputOutput/PIPITS & ~/GGOSS/GenomicsProgram.sh ;;
esac

```

10.9.3.6 GGOSS scripts for unique GGOSS built tools

10.9.3.6.1 Conserved gene finding tool

```
#!/bin/bash

#requires FASTA format

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                Conserved Sequence Finder run from GGOSS Genomics program"
"
echo | (date +"                Conserved Sequence Finder Start Date: %d-%m-%y Time: %T"
"
)

echo ""

yad --title="GENOME SEQUENCING PROGRAM -- Conserved Sequence Finder
Created by Giles Holt" --timeout=10 --no-buttons --no-escape --text="Please check the Conserved
Sequence Finder Logfile to ascertain and

confirm the success of Conserved Sequence Finder analysis of your samples. You can

also find other information there.

" &

Percent=0

echo $Percent
    echo "#Running Conserved Sequence Finder... ${Percent}% complete| Sorting and editing files"

#Script for conserved DNA region identification
NotFirstTimeHere=0
#When incorporating into GGOSS can get this from settings
MinLength=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/GeneFinder_SettingsChange.txt)
#As may be more than 1
InputFileList=$(cat ~/GGOSS/tmp/ConservedGeneFinder_SelectedFiles.txt)
echo "List of gene files:
$InputFileList"
rm ~/GGOSS/tmp/NumberOfCharacters.txt
rm ~/GGOSS/tmp/IdentifiedConservedRegions.txt
NumberOfInputFiles=$(echo "$InputFileList" | grep -c -v "ThisIsMyAntiMatch")
echo "Number of gene Files: $NumberOfInputFiles"
File=1
#Loop to number of input files to search for conserved region
for i in $(seq 1 $NumberOfInputFiles);do
```

#take frames from file with least number of characters (as you're looking for a conserved region, there is no need to search from multiple files, merely the one with the least in, as the region needs to be in that one as well)

```
InputFile=$(echo "$InputFileList" | awk -v x=$File 'NR==x {print}' | tr -d '[:space:]')
echo "InputFile: $InputFile"
```

```
#Removes first line of name info from FASTA file, and removes spaces in the FASTA file
sed '1d' ~/GGOSS_InputOutput/FastqFiles/"$InputFile" | tr -d '[:space:]' >
~/GGOSS_InputOutput/FastqFiles/tmp.txt
```

```
mv ~/GGOSS_InputOutput/FastqFiles/tmp.txt ~/GGOSS_InputOutput/FastqFiles/"$InputFile"
```

```
#count number characters in file
cat ~/GGOSS_InputOutput/FastqFiles/"$InputFile" | wc -m >>
~/GGOSS/tmp/NumberOfCharacters.txt
```

```
File=$(( $File + 1 ))
done
```

```
Percent=0
```

```
echo $Percent
echo "#Running Conserved Sequence Finder... ${Percent}% complete| Commencing sequence
finder"
```

```
#File with least characters
```

```
#order small to large, find line in file that is that number, use that line number to identify the file
```

```
LowestCharacterNumber=$(cat ~/GGOSS/tmp/NumberOfCharacters.txt | sort | awk 'NR==1 {print}')
echo "LowestCharacterNumber:$LowestCharacterNumber"
```

```
FileWithLeastCharactersPrep=$(grep -n "$LowestCharacterNumber"
~/GGOSS/tmp/NumberOfCharacters.txt | awk -F ':' 'NR==1 {print $1}')
```

```
echo "FileWithLeastCharactersPrep:$FileWithLeastCharactersPrep"
```

```
FileWithLeastCharacters=$(echo "$InputFileList" | awk -v x="$FileWithLeastCharactersPrep"
'NR==x {print}' | tr -d '[:space:]')
echo "FileWithLeastCharacters: $FileWithLeastCharacters"
```

```
NumberOfInputFiles=$(( $NumberOfInputFiles - 1 ))
echo "NumberOfInputFiles: $NumberOfInputFiles"
```

```
#Ensures THE LAST FRAME ends with minlength and last base of genome
LastFrameCutOff=$(( $LowestCharacterNumber - $MinLength + 1 ))
echo "LastFrameCutOff: $LastFrameCutOff"
```

```
#Percentage calculator
```

```
PercentIncrement=$(echo "scale=6; 100 / $LastFrameCutOff" | bc)
```

```
StartCharacter=1
```

```
#Loop for number of frames to search
```

```
for i in $(seq 1 "$LastFrameCutOff");do
```

```
echo "Frame search: $StartCharacter / $LastFrameCutOff"
```

```

if (( $StartCharacter == 1 ));then
EndCharacter=$MinLength
else
EndCharacter=$(( $StartCharacter + $MinLength - 1 ))
fi

echo "FileWithLeastCharacters: $FileWithLeastCharacters"
#take character frame
echo "StartCharacter: $StartCharacter"
echo "EndCharacter: $EndCharacter"
Frame=$(cut -c "${StartCharacter}"-"${EndCharacter}")
~/GGOSS_InputOutput/FastqFiles/"$FileWithLeastCharacters" | tr -d '[:space:]')
echo "Frame:$Frame"

#search for frame matches in all files (genomes)
MatchTotal=0
File=1
for i in $(seq 1 "$NumberOfInputFiles");do

#removes the search from file from list and selects first file left over
InputFile=$(echo "$InputFileList" | grep -v "$FileWithLeastCharacters" | awk -v x="$File"
'NR==x {print}' | tr -d '[:space:]')
echo "InputFile:$InputFile"

MatchPrep=$(grep -c "$Frame" ~/GGOSS_InputOutput/FastqFiles/"$InputFile")

echo "MatchPrep: $MatchPrep"

MatchTotal=$(( $MatchPrep + $MatchTotal ))

echo "MatchTotal: $MatchTotal"

File=$(( $File + 1 ))
done

#if consistent match add it to file of conserved regions
if (( $MatchTotal >= $NumberOfInputFiles ));then

#Current Frame -1 from end grep -c to last minlength number of characters from old frame, if
equals 1 then amalgamate into longer conserved frame
CurrentFrameNumberCharacters=$(echo "$Frame" | tr -d '[:space:]' | wc -m)
echo "CurrentFrameNumberCharacters: $CurrentFrameNumberCharacters"
CurrentFrameNumberCharactersMinus1=$(( $CurrentFrameNumberCharacters - 1 ))
echo "CurrentFrameNumberCharactersMinus1: $CurrentFrameNumberCharactersMinus1"
CurrentFrameCheck=$(echo "$Frame" | tr -d '[:space:]' | cut -c 1-
"${CurrentFrameNumberCharactersMinus1}")
FrameMemoryLinked=$(echo "$Frame" | tr -d '[:space:]' | grep -c "$CurrentFrameCheck")

if (( $FrameMemoryLinked == 1 ));then

CombineWithPreviousFrame=1

if (( "$NotFirstTimeHere" == "$LastLoop" ));then
NotFirstTimeHere=$StartCharacter

```

```

CombineWithPreviousFrame=2
else
#current loop number when frame region match found
NotFirstTimeHere=$StartCharacter
fi

#If the previous loop was also a match then combine frames and remove the smaller previous
frame from conserved file
if (( $CombineWithPreviousFrame == 2 ));then
LastCharacterOfCurrentFrame=$(( {#Frame}-1))
Frame=$(echo "${FrameMemory}${Frame:$LastCharacterOfCurrentFrame:1}")
#Remove last line in conserved file as its just a smaller conserved frame of this one
sed 'd' ~/GGOSS/tmp/IdentifiedConservedRegions.txt >
~/GGOSS/tmp/IdentifiedConservedRegions.txt
fi
fi
echo "Conserved region found: $Frame"
echo "$Frame" >> ~/GGOSS/tmp/IdentifiedConservedRegions.txt
fi

Percent=$(echo "$Percent + $PercentIncrement" | bc)

echo $Percent
echo "#Running Conserved Sequence Finder... ${Percent}% complete| Frame:
${StartCharacter}/${LastFrameCutOff}"

#makes a variable record of current frame before moving to the next frame. Frame memory is only
ever used if a match is found, this enable large conserved fragments to build, rather than lots of cross
over fragments of minlength size
FrameMemory=$Frame
LastLoop=$StartCharacter

StartCharacter=$(( $StartCharacter + 1 ))
done
NumberOfConservedRegionsIdentified=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/IdentifiedConservedRegions.txt)
echo "Number of conserved regions identified: $NumberOfConservedRegionsIdentified"
echo "Conserved region/s:"
cat ~/GGOSS/tmp/IdentifiedConservedRegions.txt

echo | (date +"
          Conserved Sequence Finder Finish Date: %d-%m-%y Time: %T
"
)

echo 100

echo "#Conserved Sequence Finder complete... 100% complete"

} | tee ~/GGOSS/LogFiles/Conserved_Sequence_Finder_LOGFILE.txt | yad --progress --auto-kill --
center --width=700 --image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -
Conserved Sequence Finder          GGOSS created by Giles Holt" --text="Running Conserved
Sequence Finder" --button="Continue"

~/GGOSS/GenomicsProgram.sh

```

10.9.3.6.2 Viral taxa finder

#file must be .csv and viral taxa must be genus or species level

```
{
echo "                GGOSS Genomics - Created by Giles Holt
"
echo "                Viral taxa finder run from GGOSS Genomics program
"
echo |(date +"                Viral taxa finder Start Date: %d-%m-%y Time: %T
"
)

echo ""
```

#automatically open logfile upon completion, get the progress bar to work

```
yad --title="GENOME SEQUENCING PROGRAM -- Viral taxa finder                Created by
Giles Holt" --timeout=10 --no-buttons --no-escape --text="Please check the Viral taxa finder Logfile
to ascertain and
```

confirm the success of Viral taxa finder of your samples. You can

also find other information there.

```
" &
```

```
Percent=0
```

```
echo $Percent
echo "#Running Viral Taxa Finder... ${Percent}% complete| Preparing..."
```

```
LineageType=$(awk -F '|' '{print $1}' ~/GGOSS/tmp/Viral_InfoFinderSettingsChange.txt)
SumMerge=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/Viral_InfoFinderSettingsChange.txt)
FileSelectedPrep1=$(awk 'NR==1 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)
FileSelectedPrep2=$(awk 'NR==2 {print}' ~/GGOSS/tmp/Path2selectedFile.txt)
FileSelected=$(echo "${FileSelectedPrep1}/${FileSelectedPrep2}" | tr -d '[:space:]' | awk -v
x="${HOME}" '{gsub("~", x, $0); print}')

```

```
echo "Taxa level to change to: $LineageType"
```

```
echo "Sum/Merge taxa based on selected taxa level: $SumMerge"
```

```
ForFinalSaveName=$LineageType
```

```
if [ $LineageType = "Host" ];then
OriginalLineageType="Host"
LineageType="Species"
fi
```

```
if [ $LineageType = "Class" ];then
LineageType="Classis"
```

```

fi

if [ $LineageType = "Order" ];then
LineageType="Ordo"
fi

if [ $LineageType = "Family" ];then
LineageType="Familia"
fi

#EditedFileName="PrepTaxa_${ForFinalSaveName}_${FileSelectedPrep2}"

#NewFileNameAndPath=$(echo "${FileSelectedPrep1}/${EditedFileName}" | tr -d '[:space:]' | awk -
v x="${HOME}" '{gsub("~", x, $0); print}')

cp "${FileSelected}" ~/GGOSS/tmp/tmp_ViralTaxaFinder_"${ForFinalSaveName}"_tmp.csv

#number of names to identify lineage from
NumberOfViruses=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS/tmp/tmp_ViralTaxaFinder_"${ForFinalSaveName}"_tmp.csv)

if [ -f ~/GGOSS/tmp/tmpFile.csv ];then
rm ~/GGOSS/tmp/tmpFile.csv
fi

PercentageToAddEachTime=$(echo "scale=2; 100/${NumberOfViruses}" | bc)

VirusNumber=2
for i in $(seq 1 "${NumberOfViruses}");do

ViralName=$(awk -F ' ' -v x="${VirusNumber}" 'NR==x {print $1}'
~/GGOSS/tmp/tmp_ViralTaxaFinder_"${ForFinalSaveName}"_tmp.csv)

# grab the family name
ViralLineagePrep=$(wget https://species.wikimedia.org/wiki/"${ViralName}" -O- | grep
"$LineageType" | grep -v "subfamilia" | awk -F "" '{print $4}')

if [ "${OriginalLineageType}" = "Host" ];then
ViralLineage=$(echo "$ViralLineagePrep" | awk -F ' ' '{print $1}')
else
ViralLineage="$ViralLineagePrep"
fi

#make file with single column of viral lineage names
echo "$ViralLineage" >> ~/GGOSS/tmp/tmpFile.csv

Percent=$(echo "scale=2; $Percent + $PercentageToAddEachTime" | bc )

TaxaNumberCurrentlyOn=$(( $VirusNumber - 1 ))
echo $Percent
echo "#Running ViralTaxaFinder... ${Percent}% complete| Taxa:${ViralName}=${ViralLineage},
${TaxaNumberCurrentlyOn}/${NumberOfViruses}"

```

```

VirusNumber=$(( $VirusNumber + 1 ))
done

echo "" | cat - ~/GGOSS/tmp/tmpFile.csv > ~/GGOSS/tmp/tmpFile1.csv

#remove column 1 from Lineage_#FileName.csv #add $NewColumn as the new 1st column in the
Lineage_#FileName.csv file
#need to sort this part

paste -d ';' $HOME/GGOSS/tmp/tmpFile1.csv
~/GGOSS/tmp/tmp_ViralTaxaFinder_"${ForFinalSaveName}"_tmp.csv >
$HOME/GGOSS_InputOutput/FastqFiles/Taxa_"${ForFinalSaveName}"_"${FileSelectedPrep2}"

rm ~/GGOSS/tmp/tmpFile.csv
rm ~/GGOSS/tmp/tmpFile1.csv
rm ~/GGOSS/tmp/tmp_ViralTaxaFinder_"${ForFinalSaveName}"_tmp.csv

echo | (date +"
                Viral Taxa Finder Finish Date: %d-%m-%y Time: %T
"
)

echo 100

echo "#Viral Taxa Finder complete... 100% complete"

} | tee ~/GGOSS/LogFiles/Viral_Taxa_Finder_LOGFILE.txt | yad --progress --auto-kill --center --
width=700 --image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM -- Viral
Taxa Finder          GGOSS created by Giles Holt" --text="Running Viral Taxa Finder" --
button="Continue"

~/GGOSS/GenomicsProgram.sh

```

10.9.3.6.3 DNA and RNA converter

```

#!/bin/bash

#requires FASTA format

{
echo "                GGOSS Genomics - Created by Giles Holt"
"
echo "                DNA/RNA Converter run from GGOSS Genomics program"
"
echo | (date +"                DNA/RNA Converter Start Date: %d-%m-%y Time: %T
"
)

echo ""

```

```

#automatically open logfile upon completion, get the progress bar to work

yad --title="GENOME SEQUENCING PROGRAM -- DNA/RNA Converter          Created
by Giles Holt" --timeout=10 --no-buttons --no-escape --text="Please check the DNA/RNA Converter
Logfile to ascertain and

confirm the success of DNA/RNA Converter of your samples. You can

also find other information there.

" &

Percent=0

echo $Percent
    echo "#Running DNA/RNA Converter... ${Percent}% complete| Preparing for file editing"

InputFileList=$(cat ~/GGOSS/tmp/DNA_RNA_Converter_SelectedFiles.txt)
echo "List of gene files:
$InputFileList"
NumberOfInputFiles=$(echo "$InputFileList" | grep -c -v "ThisIsMyAntiMatch")
echo "Number of gene Files: $NumberOfInputFiles"

DNA_RNA_Converter=$(awk -F '|' '{print $1}'
~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt)
InputStrandSense=$(awk -F '|' '{print $2}' ~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt)
StrandConversion=$(awk -F '|' '{print $3}'
~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt)
NucleicAcidType=$(awk -F '|' '{print $4}' ~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt)
FileType=$(awk -F '|' '{print $5}' ~/GGOSS/tmp/DNA_RNA_Converter_SettingsChange.txt)

#Percentage calculator
PercentIncrement=$(echo "scale=6; 100 / $NumberOfInputFiles" | bc)

File=1
#Loop to number of input files to search for conserved region
for i in $(seq 1 $NumberOfInputFiles);do
#take frames from file with least number of characters (as you're looking for a conserved region, there
is no need to search from multiple files, merely the one with the least in, as the region needs to be in
that one as well)

echo $Percent
    echo "#Running DNA/RNA Converter... ${Percent}% complete| File:
${File}/${NumberOfInputFiles}"

InputFile=$(echo "$InputFileList" | awk -v x=$File 'NR==x {print}' | tr -d '[:space:]')
echo "InputFile: $InputFile"

if [ "$FileType" = "FASTA format" ];then
#Removes first line of name info from FASTA file, and removes spaces in the FASTA file

```

```

sed '1d' ~/GGOSS_InputOutput/FastqFiles/"$InputFile" | tr -d '[:space:]' >
~/GGOSS/tmp/tmpEdit_"$InputFile"

else

cp ~/GGOSS_InputOutput/FastqFiles/"$InputFile" ~/GGOSS/tmp/tmpEdit_"$InputFile"

fi

OriginalInputFileName=$InputFile

FuturetmpRemoval="tmpEdit_${InputFile}"

InputFile="tmpEdit_${InputFile}"

#DNA to RNA
if [ "$DNA_RNA_Converter" = "DNA to RNA" ];then
  if [ "$InputStrandSense" = "5-3" ];then
    sed 's/T/U/g' ~/GGOSS/tmp/"${InputFile}" >
~/GGOSS_InputOutput/FastqFiles/RNA_"${OriginalInputFileName}"
    InputFile="RNA_${OriginalInputFileName}"
  else
    #Reverse Complement
    #first create alternate code for bases, so they can be changed without issue, then create the
    complements, Reverse the order, then Switch out the T with U
    cat ~/GGOSS/tmp/"${InputFile}" | sed 's/A/1xz5/g' | sed 's/T/2km6/g' | sed 's/C/3qr7/g' | sed
's/G/4bj8/g' | sed 's/1xz5/T/g' | sed 's/2km6/A/g' | sed 's/3qr7/G/g' | sed 's/4bj8/C/g' | rev | sed 's/T/U/g'
> ~/GGOSS_InputOutput/FastqFiles/RNA_5_3_"${OriginalInputFileName}"
    InputFile="RNA_5_3_${OriginalInputFileName}"
  fi
fi

#RNA to DNA
if [ "$DNA_RNA_Converter" = "RNA to DNA" ];then
  sed 's/U/T/g' ~/GGOSS/tmp/"${InputFile}" >
~/GGOSS_InputOutput/FastqFiles/DNA_"${OriginalInputFileName}"
  InputFile="DNA_${OriginalInputFileName}"
fi

#Complement
if [ "$StrandConversion" = "Complement" ];then

  #Complement DNA
  if [ "$NucleicAcidType" = "DNA" ];then
    #first create alternate code for bases, so they can be changed without issue, then create the
    complements
    cat ~/GGOSS/tmp/"${InputFile}" | sed 's/A/1xz5/g' | sed 's/T/2km6/g' | sed 's/C/3qr7/g' | sed
's/G/4bj8/g' | sed 's/1xz5/T/g' | sed 's/2km6/A/g' | sed 's/3qr7/G/g' | sed 's/4bj8/C/g' >
~/GGOSS_InputOutput/FastqFiles/Complement_"${OriginalInputFileName}"

    InputFile="Complement_${OriginalInputFileName}"

  else
    #Complement RNA

```

```

    cat ~/GGOSS/tmp/"${InputFile}" | sed 's/U/1xz5/g' | sed 's/A/2km6/g' | sed 's/C/3qr7/g' | sed
's/G/4bj8/g' | sed 's/1xz5/A/g' | sed 's/2km6/U/g' | sed 's/3qr7/G/g' | sed 's/4bj8/C/g' >
~/GGOSS_InputOutput/FastqFiles/Complement_"${OriginalInputFileName}"

```

```

    InputFile="Complement_"${OriginalInputFileName}"

```

```

fi

```

```

fi

```

```

#Reverse

```

```

if [ "$StrandConversion" = "Reverse" ];then

```

```

    cat ~/GGOSS/tmp/"${InputFile}" | rev >
~/GGOSS_InputOutput/FastqFiles/Reverse_"${OriginalInputFileName}"

```

```

    InputFile="Reverse_"${OriginalInputFileName}"

```

```

fi

```

```

#Reverse Complement

```

```

if [ "$StrandConversion" = "Reverse complement" ];then

```

```

    #Reverse Complement DNA

```

```

    if [ "$NucleicAcidType" = "DNA" ];then

```

```

        #first create alternate code for bases, so they can be changed without issue, then create the
        complements

```

```

        cat ~/GGOSS/tmp/"${InputFile}" | sed 's/A/1xz5/g' | sed 's/T/2km6/g' | sed 's/C/3qr7/g' | sed
's/G/4bj8/g' | sed 's/1xz5/T/g' | sed 's/2km6/A/g' | sed 's/3qr7/G/g' | sed 's/4bj8/C/g' | rev >
~/GGOSS_InputOutput/FastqFiles/RevComplement_"${OriginalInputFileName}"

```

```

        InputFile="RevComplement_"${OriginalInputFileName}"

```

```

    else

```

```

        #Reverse Complement RNA

```

```

        cat ~/GGOSS/tmp/"${InputFile}" | sed 's/U/1xz5/g' | sed 's/A/2km6/g' | sed 's/C/3qr7/g' | sed
's/G/4bj8/g' | sed 's/1xz5/A/g' | sed 's/2km6/U/g' | sed 's/3qr7/G/g' | sed 's/4bj8/C/g' | rev >
~/GGOSS_InputOutput/FastqFiles/RevComplement_"${OriginalInputFileName}"

```

```

        InputFile="RevComplement_"${OriginalInputFileName}"

```

```

    fi

```

```

fi

```

```

##### take first line from $OriginalInputFileName, and make it the first line of the $InputFile,
output it as $OriginalInputFileName #####

```

```

#fixes FASTA files back to FASTA format

```

```

if [ "$FileType" = "FASTA format" ];then

```

```

    TopLine=$(awk 'NR==1' ~/GGOSS_InputOutput/FastqFiles/"${OriginalInputFileName}")

```

```

# Here add topline info to inputfile, output as a tmp file

```

```

echo "$TopLine" | cat - ~/GGOSS_InputOutput/FastqFiles/"${InputFile}" >
~/GGOSS/tmp/tmpFullFile_DNA_RNA_Converter.txt

```

```

#mv back with correct name

```

```

mv ~/GGOSS/tmp/tmpFullFile_DNA_RNA_Converter.txt
~/GGOSS_InputOutput/FastqFiles/"${InputFile}"

```

```

fi

```

```

#removes the tmp files made during the process
rm ~/GGOSS/tmp/"${FuturetmpRemoval}"

Percent=$(echo "$Percent + $PercentIncrement" | bc)

echo $Percent
echo "#Running DNA/RNA Converter... ${Percent}% complete| File:
${File}/${NumberOfInputFiles}"

File=$(( $File + 1 ))
done

echo | (date +"
                DNA/RNA Converter Finish Date: %d-%m-%y Time: %T
"
)

echo 100

echo "#DNA/RNA Converter complete... 100% complete"

} | tee ~/GGOSS/LogFiles/DNA_RNA_Converter_LOGFILE.txt | yad --progress --auto-kill --center -
-width=700 --image=$ICON --image-on-top --title="GENOME SEQUENCING PROGRAM --
DNA/RNA Converter GGOSS created by Giles Holt" --text="Running DNA/RNA
Converter" --button="Continue"

~/GGOSS/GenomicsProgram.sh

```

10.9.3.6.4 Taxonomy abundance filtering

10.9.3.6.4.1 Percentage based trimming

```

#!/bin/sh

#clear any previous tmp files assoicated to this (these will only exist if a run has been stopped half way
through)
if [ -f ~/GGOSS/tmp/tmp.txt ];then
rm ~/GGOSS/tmp/tmp.txt
fi
if [ -f ~/GGOSS/tmp/testPreGraphPrep1.csv ];then
rm ~/GGOSS/tmp/testPreGraphPrep1.csv
fi
if [ -f ~/GGOSS/tmp/SamplePercentTrack.csv ];then
rm ~/GGOSS/tmp/SamplePercentTrack.csv
fi
if [ -f ~/GGOSS/tmp/sumColumns.csv ];then
rm ~/GGOSS/tmp/sumColumns.csv
fi
if [ -f ~/GGOSS/tmp/sumRows.csv ];then
rm ~/GGOSS/tmp/sumRows.csv
fi

```

```

if [ -f ~/GGOSS/tmp/tmp2.txt ];then
rm ~/GGOSS/tmp/tmp.txt2
fi

#set the file name
FileName=$(awk -F '|' '{ print $1 }' ~/GGOSS/tmp/OTUtableSettingsChange.txt)

echo "File selected: $FileName"

echo "Output file will be named: Trim_${FileName}"

#copy the file and add trimmed to it
cp ~/GGOSS_InputOutput/Mothur/OTUtable/"$FileName"
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}"

#####Trimming the bacteria names to OTU numbers#####

#check if they wish to trim bacteria names
OTUtrimYesOrNo=$(awk -F '|' '{ print $2 }' ~/GGOSS/tmp/OTUtableSettingsChange.txt)

#this yes isn't working
if [ "$OTUtrimYesOrNo" = "Yes" ]
then
#remove everything in column 1 from 'Bacteria'

#make a key file for the full bacterial names
awk -F '|' '{print $1}' ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" >
~/GGOSS_InputOutput/Mothur/OTUtable/"OTUkey_${FileName}"

#takes column 1, remove "" (they cause problems with awk if not), then remove everything after the
OTU numbers
awk -F '|' '{print $1}' ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" | sed 's/^"//g' |
awk -F 'B' '{print $1}' > ~/GGOSS/tmp/tmp.txt

#take all the columns from original files excluding the first column
cut -d',' -f2- ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" >
~/GGOSS/tmp/tmp2.txt

#combine the two files
paste -d, ~/GGOSS/tmp/tmp.txt ~/GGOSS/tmp/tmp2.txt >
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}"

echo "OTU names trimmed to number ID's, a file containing the un-adulterated names
(${FileName}_OTUkey) has been made and placed in the directory:
GGOSS_InputOutput/Mothur/OTUtable"

else

echo "OTU names left un-altered, as per user settings"

fi

#####

```

```

#set the percentage cut as variable
PercentCut=$(awk -F '|' '{ print $4 }' ~/GGOSS/tmp/OTUtableSettingsChange.txt)
echo "User settings for percentage trim: $PercentCut"

#variable of total number of columns
NumberColsPrep=$(awk -F',' '{print NF; exit}'
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")
NumberCols=$(( $NumberColsPrep - 1 ))
echo "Number of Samples: $NumberCols"

#loop number of columns
#set column
Column=2
for i in $(seq 1 "$NumberCols")
do

    #Sum each column
    if [ -f ~/GGOSS/tmp/sumColumns.csv ]

    then
        tail -n+2 ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" | awk -v x="$Column" -
F ',' '{ sum += $x } END { print sum }' >> ~/GGOSS/tmp/sumColumns.csv

    else
        tail -n+2 ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" | awk -v x="$Column" -
F ',' '{ sum += $x } END { print sum }' > ~/GGOSS/tmp/sumColumns.csv
    fi
    Column=$(( $Column + 1 ))
#end loop
done

#transpose so can sum the rows (as columns)
#remove OTU name column and sample name line first so can calculate sum later

cut -f 2- ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" | tail -n+2 | awk -F ',' '
{
    for (i=1; i<=NF; i++) {
        a[NR,i] = $i
    }
}
NF>p { p = NF }
END {
    for(j=1; j<=p; j++) {
        str=a[1,j]
        for(i=2; i<=NR; i++){
            str=str" "a[i,j];
        }
        print str
    }
}' > ~/GGOSS/tmp/Transposed.csv

#variable of total number of rows

NumberRowsPrep=$(wc -l < ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")

```

```

NumberRows=$(( $NumberRowsPrep - 1 ))

echo "Number of OTU's: $NumberRows"

Column=1
#loop number of rows
for i in $(seq 1 "$NumberRows")
do
    #sum of each row
    if [ -f ~/GGOSS/tmp/sumRows.csv ]
    then
        awk -v x="$Column" -F ' ' '{ sum += $x } END { print sum }' ~/GGOSS/tmp/Transposed.csv >>
~/GGOSS/tmp/sumRows.csv
    else
        awk -v x="$Column" -F ' ' '{ sum += $x } END { print sum }' ~/GGOSS/tmp/Transposed.csv >
~/GGOSS/tmp/sumRows.csv
    fi

Column=$(( $Column + 1 ))

#end loop
done

####note different delimiters happening in file from transposed

#loop to number of otu's

Row=1
#this loop sets the column to work through
for i in $(seq 1 "$NumberRows")
do

#calculate if the given OTU (row) is less than $PercentCut% of all the OTU's (rows) combined

RowOTUtoCheck=$(awk -v x="$Row" 'NR==x { print }' ~/GGOSS/tmp/sumRows.csv)

TotalOfRowsOTUs=$(awk '{ sum += $1 } END { print sum }' ~/GGOSS/tmp/sumRows.csv)

PercentageOfTotalBacteriaAccrossAllSamples=$(echo "(( $RowOTUtoCheck / $TotalOfRowsOTUs)
* 100)" | bc -l)

FileRowChangeToAvoidUsingDataHeading=$(( $Row + 1 ))

OTU=$(awk -v x="$FileRowChangeToAvoidUsingDataHeading" -F ' ' 'NR==x { print $1 }'
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_{$FileName}")

echo "OTU; {$OTU}, makes up {$PercentageOfTotalBacteriaAccrossAllSamples}% of the total
bacterial community in this study"

#if the % is <$PercentCut then make variable $PercentCut
LowerOrHigher=$(echo "$PercentageOfTotalBacteriaAccrossAllSamples $PercentCut 2 3" | awk '{if
($1 < $2) print $3; else print $4}')

```

```

if [ "$LowerOrHigher" -eq 2 ]
then

echo "

This means $OTU makes up less than ${PercentCut}% of the OTU community,"

touch ~/GGOSS/tmp/SamplePercentTrack.csv

Column=1

#this loop works its way down through the column
for i in $(seq 1 $NumberCols)
do

FileColumnChangeToAvoidUsingDataHeading=$(( $Column + 1 ))

#calculate if the given OTU (row) is less than $PercentCut% of any given sample (column) combined
SingleSampleSingleOTUtoCheck=$(awk -F ' ' -v x=$FileRowChangeToAvoidUsingDataHeading -v
y=$FileColumnChangeToAvoidUsingDataHeading 'NR==x {print $y}'
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")

TotalOfSample=$(awk -v x=$Column 'NR==x { print $1 }' ~/GGOSS/tmp/sumColumns.csv)

PercentageOf_a_BacteriaIn_a_Sample=$(echo "($SingleSampleSingleOTUtoCheck /
$TotalOfSample) * 100" | bc -l)

SampleName=$(awk -F ' ' -v y=$FileColumnChangeToAvoidUsingDataHeading 'NR==1 {print $y}'
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")

echo "The OTU; ${OTU}, makes up ${PercentageOf_a_BacteriaIn_a_Sample}% of sample;
$SampleName"

#make sure its not a decimal place
#increase the number
#PercentageOf_a_BacteriaIn_a_SamplePrep1=$(echo "$PercentageOf_a_BacteriaIn_a_SamplePrep +
10000" | bc -l)
#take number before decimal place
#PercentageOf_a_BacteriaIn_a_Sample=$(echo "$PercentageOf_a_BacteriaIn_a_SamplePrep1" |
awk -F '.' '{ print $1 }')

LowerOrHigher=$(echo "$PercentageOf_a_BacteriaIn_a_Sample $PercentCut 2 3" | awk '{if ($1 <
$2) print $3; else print $4}')

if [ "$LowerOrHigher" -eq 2 ]
#if makes up less than $PercentCut% of each samples community
then
echo "1" >> ~/GGOSS/tmp/SamplePercentTrack.csv

else

```

```
echo "The OTU; ${OTU}, still makes up more than ${PercentCut}% in sample ${SampleName}, and
thus has not been removed"
fi
```

```
Column=$(( $Column + 1 ))
```

```
done
```

```
#calc sum of ~/SamplePercentTrack.csv
```

```
TotalSamplePercentageCount=$(awk -F ',' '{ sum += $1 } END { print sum }'
```

```
~/GGOSS/tmp/SamplePercentTrack.csv)
```

```
#wrap in if that confirms every sample less than $PercentCut%. if sum of ~/SamplePercentTrack.csv
is equal to total sample number
```

```
if [ "$TotalSamplePercentageCount" -eq "$NumberCols" ]
```

```
then
```

```
echo "
```

```
The ${OTU} makes up less than ${PercentCut}% of any given sample and less than ${PercentCut}%
of the OTU community.
```

```
    Removing ${OTU}
```

```
    "
```

```
#replace the OTU line with "LineToBeRemoved"
```

```
sed "${FileRowChangeToAvoidUsingDataHeading}s/.*/LineToBeRemoved/"
```

```
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" >
```

```
~/GGOSS/tmp/testPreGraphPrep1.csv
```

```
mv ~/GGOSS/tmp/testPreGraphPrep1.csv
```

```
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}"
```

```
if [ -f ~/GGOSS/tmp/SamplePercentTrack.csv ]
```

```
then
```

```
rm ~/GGOSS/tmp/SamplePercentTrack.csv
```

```
fi
```

```
fi
```

```
fi
```

```
if [ -f ~/GGOSS/tmp/SamplePercentTrack.csv ];then
```

```
rm ~/GGOSS/tmp/SamplePercentTrack.csv
```

```
fi
```

```
RemovalCount=$(grep -c "LineToBeRemoved"
```

```
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")
```

```
echo "
```

```
    OTU's removed so far: $RemovalCount
```

```
    "
```

```
Row=$(( $Row + 1 ))
```

```
done
```

```

#Count all the lines containing "LineToBeRemoved"

RemovalCount=$(grep -c "LineToBeRemoved"
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}")

echo "
                $RemovalCount OTU's have been removed
"

#Remove all the lines containing "LineToBeRemoved"
sed '/LineToBeRemoved/d' ~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}" >
~/GGOSS/tmp/testPreGraphPrep1.csv

mv ~/GGOSS/tmp/testPreGraphPrep1.csv
~/GGOSS_InputOutput/Mothur/OTUtable/"Trim_${FileName}"

#run a load of if files there delete it (for all the tem files made)

if [ -f ~/GGOSS/tmp/tmp.txt ];then
rm ~/GGOSS/tmp/tmp.txt
fi
if [ -f ~/GGOSS/tmp/testPreGraphPrep1.csv ];then
rm ~/GGOSS/tmp/testPreGraphPrep1.csv
fi
if [ -f ~/GGOSS/tmp/SamplePercentTrack.csv ];then
rm ~/GGOSS/tmp/SamplePercentTrack.csv
fi
if [ -f ~/GGOSS/tmp/sumColumns.csv ];then
rm ~/GGOSS/tmp/sumColumns.csv
fi
if [ -f ~/GGOSS/tmp/sumRows.csv ];then
rm ~/GGOSS/tmp/sumRows.csv
fi
if [ -f ~/GGOSS/tmp/tmp2.txt ];then
rm ~/GGOSS/tmp/tmp2.txt
fi

```

10.9.3.6.4.2 Taxonomy group pooling

```

#!/bin/bash

tail -n +2 ~/TaxaFile.csv | awk -F '|' '{print $2}' > ~/TaxaFiletmp.csv

#replace column 1 with column made - ##
#take first 2 columns
tail -n +2 ~/TaxaFile.csv | awk -F '|' '{print $1,$2}' > ~/TaxaFileFirst2.csv

#add new column in
paste -d' ' ~/TaxaFileFirst2.csv ~/TaxaFiletmp.csv > ~/TaxaFile3WithNewTaxa.csv

#take 4th to end column

```

```

tail -n +2 ~/TaxaFile.csv | cut -d ',' -f 4- > ~/TaxaFileCol4on.csv
#put the two files together
paste -d ' ' ~/TaxaFile3WithNewTaxa.csv ~/TaxaFileCol4on.csv > ~/TaxaFileNewTaxa.csv

#sort file by column 3
sort -t' ' -k 3,3 ~/TaxaFileNewTaxa.csv > ~/TaxaFileNewTaxaSorted.csv

#make a file for each family
#number of unique names

UniqueNamesList=$(awk -F ' ' '{print $3}' ~/TaxaFileNewTaxaSorted.csv | uniq)
NumberOfUniqueNames=$(echo "$UniqueNamesList" | grep -c -v "ThisIsMyAntiMatch")
touch ~/TaxaFileTaxaGroupedAndSummed.csv
line=1
for i in $(seq $NumberOfUniqueNames)
do
UniqueName=$(echo "$UniqueNamesList" | awk -v x=$line 'NR==x {print}')

ListOfAllForGivenName=$(awk -F ' ' -v x=$UniqueName '$3 == x { print $0 }'
~/TaxaFileNewTaxaSorted.csv)

#sum all in ListOfAllForGivenName
line2=1

#awk part only calculating the first column
SumLine=$(echo "$ListOfAllForGivenName" | cut -d ' ' -f 4- | awk -F ',' 'BEGIN { ORS=" " } {for
(i=1;i<=NF;i++) sum[i]+=$i;}; END{for (i in sum) print sum[i];}')

#add name back to the sum line
paste <(echo "$UniqueName") <(echo "$SumLine") --delimiters ' ' >>
~/TaxaFileTaxaGroupedAndSummed.csv

line=$(( $line + 1 ))
done

#at the end add the title back on
TopLine=$(head -n +1 ~/TaxaFile.csv | cut -d ',' -f 3-)
sed -i "1i $TopLine" ~/TaxaFileTaxaGroupedAndSummed.csv

```

10.9.3.6.5 Data plotting for community analysis

10.9.3.6.5.1 Table creation for rarefaction

```

#For rarefaction

echo "Commencing combining and transposing the .shared and .taxonomy files"

#print column 1 and 3 from cons taxonomy (OTU and Taxonomy)

awk -F $'\t' '{ print $1 $3 }' ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt >
~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy1.txt

rm ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt

```

```

mv ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy1.txt
~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt

#rm column 1 and column 3 from shared
cut -f 2,4- ~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt >
~/GGOSS_InputOutput/Mothur/rarefaction.shared1.txt

rm ~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt

#remove 1st row from rarefaction.shared1.txt - OTU names
sed '1 d' ~/GGOSS_InputOutput/Mothur/rarefaction.shared1.txt >
~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt

#####-----#####

#### ----- ## Cut OTU's that have a selectively low count

#####-----#####

#cut off the sample names
cut -f 2- ~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt >
~/GGOSS_InputOutput/Mothur/rarefaction.sharedCalcPrepOTUremoval.txt

NumberOfColumns=$(awk -F $'\t' '{print NF; exit}'
~/GGOSS_InputOutput/Mothur/rarefaction.sharedCalcPrepOTUremoval.txt)
echo "Number of columns: $NumberOfColumns"

ColumnRemovePrep=1
Column=1
for i in $(seq 1 $NumberOfColumns)
do
#take column one by one and #sum the column
awk -F $'\t' -v x=$Column '{print $x}'
~/GGOSS_InputOutput/Mothur/rarefaction.sharedCalcPrepOTUremoval.txt | paste -sd+ - | bc >
~/GGOSS_InputOutput/Mothur/rarefaction.sharedCalcPrepOTUremoval1.txt

#line 1 is sum of column 2 in original file so..

ColumnSum=$(awk 'NR==1 { print }'
~/GGOSS_InputOutput/Mothur/rarefaction.sharedCalcPrepOTUremoval1.txt)

echo "Column number: $Column
Column Sum: $ColumnSum"

#above assumes (which is generally the case) that OTU's are in size order, therefore at the first 2
remove all columns after - this makes it a lot quicker!

if (( "$ColumnSum" < 3 ))
then
echo "Column $Column sum is less than 3 - Columns here after are less than 3"
ColumnStart=1
#take from 1st column to the column of the first less than 3

```

```

cut -f${ColumnStart}-${Column} ~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt >
~/GGOSS_InputOutput/Mothur/rarefaction.shared4.txt

mv ~/GGOSS_InputOutput/Mothur/rarefaction.shared4.txt
~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt

AmountOfOTUsRemoved=$(echo "$NumberOfColumns - $Column" | bc)

break

fi
Column=$(( $Column + 1 ))
done

#remove the excess OTU's from the taxonomy file

sed -n "1,${Column}p" ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt >
~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy4.txt

mv ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy4.txt
~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt

echo "Number of OTU's removed: $AmountOfOTUsRemoved"

cp ~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt
~/GGOSS_InputOutput/Mothur/CheckOTURemovalWorkedrarefaction.shared.txt

#line 1 in column 2 in original file

#####-----#####

#-----End of OTU cutting section

#####-----#####

#####-----#####

#-----Transpose shared file section

#####-----#####

NumberOfLines=$(grep -c -v "ThisIsMyAntiMatch"
~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt)
echo "Number of lines: $NumberOfLines"

NumberOfColumns=$(awk -F '\t' '{print NF; exit}'
~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt)
echo "Number of columns: $NumberOfColumns"

line=1
for i in $(seq 1 $NumberOfLines)
do

#checks the file that the column loop builds a single new column from is empty
if [ -f ~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt ]

```

```

then
rm ~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt
fi

touch ~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt

#if [ -f ~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt ]
#then
#rm ~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt
#fi

echo "There are $NumberOfColumns lines to build for each column
Currently up to building column ${line}, out of $NumberOfLines columns"

TenPercentOfColumnMade=$(echo "$NumberOfColumns / 10" | bc)
TwentyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 2" | bc)
ThirtyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 3" | bc)
FortyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 4" | bc)
FiftyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 5" | bc)
SixtyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 6" | bc)
SeventyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 7" | bc)
EightyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 8" | bc)
NintyPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 9" | bc)
HundredPercentOfColumnMade=$(echo "$TenPercentOfColumnMade * 10" | bc)

Col=1
for i in $(seq 1 $NumberOfColumns);do

if (( "$Col" == "$TenPercentOfColumnMade" ));then
echo "column ${line}, 10% complete"
fi

if (( "$Col" == "$TwentyPercentOfColumnMade" ));then
echo "column ${line}, 20% complete"
fi

if (( "$Col" == "$ThirtyPercentOfColumnMade" ));then
echo "column ${line}, 30% complete"
fi

if (( "$Col" == "$FortyPercentOfColumnMade" ));then
echo "column ${line}, 40% complete"
fi

if (( "$Col" == "$FiftyPercentOfColumnMade" ));then
echo "column ${line}, 50% complete"
fi

if (( "$Col" == "$SixtyPercentOfColumnMade" ));then
echo "column ${line}, 60% complete"
fi

if (( "$Col" == "$SeventyPercentOfColumnMade" ));then
echo "column ${line}, 70% complete"
fi

```

```

if (( "$Col" == "$EightyPercentOfColumnMade" ));then
echo "column ${line}, 80% complete"
fi

if (( "$Col" == "$NintyPercentOfColumnMade" ));then
echo "column ${line}, 90% complete"
fi

if (( "$Col" == "$HundredPercentOfColumnMade" ));then
echo "column ${line}, 100% complete"
fi

#gets every column from $line and makes one column with as many lines as there had been columns
awk -F '$\t' -v x=$Col -v y=$line 'NR==y { print $x }'
~/GGOSS_InputOutput/Mothur/rarefaction.shared.txt >>
~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt

Col=$(( $Col + 1 ))
done

if [ -f ~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt ];then

    if [ -f ~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt ];then
    echo "Line: $line1 building template for transposing..."
    else
    touch ~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt
    fi

#paste the above file together with it
paste ~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt
~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt >>
~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt

mv ~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt
~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt

else

#rename the above file as the final files name that will have everything pasted to it
mv ~/GGOSS_InputOutput/Mothur/rarefaction.shared2.txt
~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt

fi

line=$(( $line + 1 ))
done

mv ~/GGOSS_InputOutput/Mothur/Pretranspose_rarefaction.shared.txt
~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt

#####-----#####

#----- End of Transpose shared file section

```

```
#####-----#####
```

```
#convert to tab delimited .csv file
```

```
tr '\t' < ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.txt >  
~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.csv  
tr '\t' < ~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.txt >  
~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.csv
```

```
paste ~/GGOSS_InputOutput/Mothur/rarefaction_cons.taxonomy.csv  
~/GGOSS_InputOutput/Mothur/transposed_rarefaction.shared.csv >  
~/GGOSS_InputOutput/Mothur/PreRarefaction.csv
```

```
rm ~/GGOSS_InputOutput/Mothur/rarefaction.shared1.txt
```

10.9.3.6.5.2 BASH run script for rarefaction

```
#!/bin/sh
```

```
yad --title="GILES -- Mothur - Rarefaction" --width=400 --center --sticky --on-top --no-buttons --no-  
escape --text-align=center --text="          Running Rarefaction
```

```
" & R < ~/GGOSS/Scripts/R_Scripts/Rarefaction.R --no-save
```

```
#need to close window  
pkill yad
```

```
mv ~/GGOSS/Scripts/Rplots.pdf ~/GGOSS_InputOutput/Mothur/Rarefaction.pdf
```

```
gnome-open ~/GGOSS_InputOutput/Mothur/Rarefaction.pdf
```

```
~/GGOSS/Buttons/ButonPostMothurAnalysis.sh
```

```
wmctrl -r 'Rarefaction.pdf — R Graphics Output' -e 0,1150,35,740,600
```

10.9.3.6.5.3 R script for rarefaction

```
#rarefaction:
```

```
rm(list=ls())  
data=read.csv("~/AdultStool/AdultStool_TimeGrouped.csv" , header=T, row.name=1, check.names =  
F)  
attach(data)  
data = t(data)  
library(vegan)  
library(BiodiversityR)
```

```
#sample = number want to sub sample to - set to lowest read count (total OTU count per sample)  
unless thats lower than user settings for lowest acceptable read count - in which case use the users  
minimum
```

```
Rarefy.data = rarefy(data, sample =25000, se=T)
```

```
Rarefy.data
```

```
#change step number in accordance to average read count as the lower the count the lower the step
number
Rarefy.curve = rarecurve(data, sample =25000, step=200, xlab ="Sequence Reads", col ="red")
rrarefy(data,25000)
```

10.9.3.6.5.4 BASH run script for MDS plot

```
#!/bin/sh
yad --title="GILES -- Mothur - MDS" --width=400 --center --sticky --on-top --no-buttons --no-escape
--text-align=center --text="          Creating MDS plot

" & R < ~/GGOSS/Scripts/R_Scripts/MDS.R --no-save

pkill yad

mv ~/GGOSS/Scripts/Rplots.pdf ~/GGOSS_InputOutput/Mothur/MDS.pdf

gnome-open ~/GGOSS_InputOutput/Mothur/MDS.pdf

echo | ~/GGOSS/Buttons/ButtonPostMothurAnalysis.sh
```

10.9.3.6.5.5 R script for MDS plot

```
#MDS

rm(list=ls())
library(cluster)
library(vegan)
data=read.delim("~/GGOSS_InputOutput/Mothur/PreRarefaction.txt" , sep=" ", header=T,
row.name=1)
attach(data)
data = t(data)
#Create dissimilarity matrix per sample (euclidean dist.)
dm = daisy(data, metric = c("euclidean"))
dm.1 = as.matrix(dm)
dm.1

#Run the call for non-metric scaling
data_MDS=metaMDS(dm.1)

#Can then view items produced by 'metaMDS' function by running following commands:
names(data_MDS)
#Or just view the results of the MDS:
data_MDS

#plot the metaMDS data using 'plot' function
x=data_MDS$points[,1]
y=data_MDS$points[,2]
plot(x,y,xlab="MDS 1", ylab="MDS 2", xlim=range(data_MDS$points[,1])*1.2, type="n",)

#Use 'text' command to label data points per sample.
text(x, y, labels=row.names(data), cex=0.5)
```

10.9.3.6.5.6 BASH script for PCA

```
#!/bin/sh

yad --title="GILES -- Mothur - PCA" --width=400 --center --sticky --on-top --no-buttons --no-escape
--text-align=center --text="          Running PCA

" & R < ~/GGOSS/Scripts/R_Scripts/PCA_Script.R --no-save

#need to close window
pkill yad

mv ~/GGOSS/Scripts/Rplots.pdf ~/GGOSS_InputOutput/Mothur/PCA.pdf

gnome-open ~/GGOSS_InputOutput/Mothur/PCA.pdf

echo | ~/GGOSS/Buttons/ButtonPostMothurAnalysis.sh

wmctrl -r 'PCA.pdf — R Graphics Output' -e 0,1150,35,740,600
```

10.9.3.6.5.7 R script for PCA

```
# PCA script
#INPUT data needs to be whole integers, so round them up first
rm(list=ls())
data=read.csv("~/AdultStool/AdultStool_SampleGroupedAndPooledfamily.csv" , header=T,
row.name=1, check.names = F)
attach(data)
data = t(data)
library(vegan)
library(BiodiversityR)

#sample = number want to sub sample to - set to lowest read count (total OTU count per sample)
unless thats lower than user settings for lowest acceptable read count - in which case use the users
minimum
Rarefy.data = rarefy(data, sample =30000, se=T)
Rarefy.data
#change step number in accordance to average read count as the lower the count the lower the step
number (step number should be roughly 1% and round up)
Rarefy.curve = rarecurve(data, sample =30000, step=200, xlab ="Sequence Reads", col ="red")
rrarefy(data,30000)

library(devtools)
#install_github("vqv/ggbiplot")
library(ggbiplot)
library(lattice)

# Look at the correlations
library(gclus)
my.abs <- abs(cor(data[,-1])) ##Can change to view correlations of other datasets
```

```

my.abs
my.colors <- dmat.color(my.abs)
my.colors
my.ordered <- order.single(cor(data[,-1]))
my.ordered

pdf(file="~/AdultStool/PreSampleAndFamilyPooledPCA_1.pdf", width=50, height=50)
cpairs(data, my.ordered, panel.colors=my.colors, gap=0.5)
dev.off()

# Do the PCA

#Make sure all values are centred and scaled accordingly by "=T"
my.prc <- prcomp(data[,-1], center=TRUE, scale=TRUE) ##Can change to view correlations of other
datasets

#Apply Kaiser criterion (variances of >1.0) to select components

#Can also use Scree plots to decide which components to use.
#Select components until change between variances drops to <1
screeplot(my.prc, main="Scree Plot", xlab="Components")
#Sometimes easier to see with line
pdf(file="~/AdultStool/SampleAndFamilyPooledPCA.all.scree.pdf", width=7, height=7)
screeplot(my.prc, main="SampleAndFamily Scree Plot for PCA", type="line" )
dev.off()
# DotChart PC1

#calculate loading values for dotplot
load <- my.prc$rotation
load
sorted.loadings<- load[order(load[, 1]), 1]
sorted.loadings
#Create labels for dotchart of 1st principle component
myTitle <- "Loadings Plot for PC1"
myXlab <- "Variable Loadings"
dotchart(sorted.loadings, main=myTitle, xlab=myXlab, cex=0.3, col="red")
#dotchart shows which variables have most significant effect on each PC

# DotChart PC2

#calculate loading values for dotplot
sorted.loadings <- load[order(load[, 2]), 2]
#Create labels for dotchart of 2nd principle component
myTitle <- "Loadings Plot for PC2"
myXlab <- "Variable Loadings"
dotchart(sorted.loadings, main=myTitle, xlab=myXlab, cex=0.3, col="red")
#dotchart shows which variables have most significant effect on each PC

#if you want to color groups
Groups=read.csv ("~/GGOSS_InputOutput/Mothur/OTUGroups.csv", header=T)

# Now draw the BiPlot
#alpha changes the opacity so can see all the points easier if they're layered on top of each other
ggbiplot(my.prc, choices=1:2, var.scale=1, obs.scale=1, var.axes=T, varname.size=4, labels.size=6,
pch=0.8, alpha = .7, groups=Groups$Group, col= Groups$Group, labels = rownames(data))

```

10.9.3.7 General GGOSS function necessities

10.9.3.7.1 File import

```
#!/bin/bash

if [ -f ~/GGOSS/tmp/dnd.txt ];then
rm ~/GGOSS/tmp/dnd.txt
rm "/tmp/GGOSSdnd.log"
rm "/tmp/GGOSSdnd2.log"
fi

#create fifo file to displaying text in --text-info pane
mkfifo "/tmp/GGOSSdnd.log"
exec 3<> "/tmp/GGOSSdnd.log"

RemoveFile=1

#creating the key id for box and plugs
id=$(echo $[( $RANDOM % ( $[10000 - 32000] + 1) ) + 10000] )

#the first pane is dnd box
#the second is --text-info from fifo file
yad --plug="$id" --tabnum=1 --dnd | while read line2
do
echo "$line2" >&3
echo "$line2" >> "/tmp/GGOSSdnd2.log"
done &

yad --plug="$id" --tabnum=2 --text-info --tail <&3>> "/tmp/GGOSSdnd2.log" &

yad --title="Import files" --center --paned --key="$id" --text="Drag and drop your files into the grey
section of the window to import them" --width="800" --height="500" --splitter="150" --button="gtk-
quit:1" --button="gtk-ok:0"
#out of the script if close buttons are clicked
case $? in
1)
rm "/tmp/GGOSSdnd.log" "/tmp/GGOSSdnd2.log"
exit;;
252)
rm "/tmp/GGOSSdnd.log" "/tmp/GGOSSdnd2.log"
exit;;
esac

cat /tmp/GGOSSdnd2.log > ~/GGOSS/tmp/dnd.txt
NumberOfFilesToCopy=$(grep -v -c "ThisIsMyAntiMatch" ~/GGOSS/tmp/dnd.txt)
PercentCompleteWorthOfEachFile=$( echo "scale=2; 100 / $NumberOfFilesToCopy" | bc )
(if [ -f ~/GGOSS/tmp/dnd.txt ];then
line=1
TotallingPercentage=0
for i in $(seq $NumberOfFilesToCopy);do

file=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/dnd.txt | sed 's/.*\//')
```

```

path=$(awk -v x=$line 'NR==x {print}' ~/GGOSS/tmp/dnd.txt | sed 's/file:\|\\|/')

cp "$path" ~/GGOSS_InputOutput/FastqFiles/"$file"
echo $TotallingPercentage
echo "#Importing files    ${TotallingPercentage}% Complete"
TotallingPercentage=$( echo "$TotallingPercentage + $PercentCompleteWorthOfEachFile" | bc )
line=$(( $line + 1 ))
done

rm "/tmp/GGOSSdnd.log" "/tmp/GGOSSdnd2.log"
if [ -f ~/GGOSS/tmp/dnd.txt ];then
rm ~/GGOSS/tmp/dnd.txt
fi

fi

) | yad --progress --auto-close --auto-kill --center --width=700 --image=$ICON --image-on-top --
title="Importing files into GGOSS: |G|ui for |G|enomic analysis incorporating |O|pen |S|ource
|S|oftware" \
--percentage=0

FilesToRemove=$(ls ~/GGOSS_InputOutput/FastqFiles/)

echo "$FilesToRemove" | yad --title="Imported Files" --list --column="Files present for analysis" --
multiple --width=800 --height=600 --center --align=center --button="Remove" --button="Continue":2
--separator=" > ~/GGOSS/tmp/AllImportedFilesToRemove.txt

mode="$?"
case $mode in
0)RemoveFile=2 ;;
2)~/GGOSS/GenomicsProgram.sh ;;
esac

#Check they want to remove
if (( "$RemoveFile" == 2 ));then

ListOfFileToRemove=$(cat ~/GGOSS/tmp/AllImportedFilesToRemove.txt)
echo "$ListOfFileToRemove" | yad --title="Remove file" --list --column="Are you sure you wish to
remove the following file/s?" --width=370 --height=300 --center --align=center --button="Yes":0 --
button="No":1

mode="$?"
case $mode in
0)~/GGOSS/GenomicsProgram.sh & RemoveFile=3 ;;
1)~/GGOSS/GenomicsProgram.sh & RemoveFile=4 ;;
esac
fi

# for removing imported files
if (( "$RemoveFile" == 3 ));then

echo "File/s to remove:
"
cat ~/GGOSS/tmp/AllImportedFilesToRemove.txt

```

```

    NumberOfFilesToRemove=$(grep -v -c "ThisIsMyAntiMatch"
~/GGOSS/tmp/AllImportedFilesToRemove.txt)
    echo "number of files to remove: $NumberOfFilesToRemove"
    FileToRemove=1
    for i in $(seq 1 "$NumberOfFilesToRemove");do
    echo "FileToRemove: $FileToRemove"

    RemoveThisFile=$(awk -v x=$FileToRemove 'NR==x {print}'
~/GGOSS/tmp/AllImportedFilesToRemove.txt)

    echo "RemoveThisFile: $RemoveThisFile"

    #uses -- and expands the variable with "" to allow for file names with spaces
    rm -- ~/GGOSS_InputOutput/FastqFiles/"$RemoveThisFile"

    FileToRemove=$(( $FileToRemove + 1 ))
    done
fi

```