Fuzzy Clustering Models for Gene Expression Data Analysis

Yu Wang

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

Research undertaken in the faculty of Engineering and Environment

2014

Declaration

I declare that the work contained in this thesis has not been submitted for any other award. To the best of my knowledge and belief, this work fully acknowledges opinions, ideas and contributions from the work of others.

Name: Yu Wang

Signature:

Date:

Acknowledgements

First of all, I would like to express my deep gratitude to my principal supervisor Maia Angelova. Without her encouragement and guidance, I cannot complete my Ph.D study. I would like to thank Mathematical Modeling Lab for excellent academic environment. In addition, my sincere appreciation to my first supervisor Akhtar Ali, he gives me many valuable suggestions.

I would appreciate my parents support. I also owe special thanks to my wife for her patience and encouragement.

Finally, I would like to appreciate Northumbria University and China Scholarship Council (CSC) to offer me a valuable opportunity studying in the UK.

Publications

Yu Wang, Maia Angelova, and Yang Zhang (2013) **A Framework for Density Weighted Kernel Fuzzy c-Means on Gene Expression Data.** *Advances in Intelligent Systems and Computing* Volume 212, pp 453-461

Yu Wang, Maia Angelova and Akhtar Ali (2013) **Fuzzy clustering of time series gene expression data with cubic-spline**. *Journal of Biosciences and Medicines*, Volume **1**, pp16-21.

Yu Wang, Maia Angelova **Weighted kernel fuzzy c-means method for gene expression analysis**, *2012 Spring Congress on Computational Biology and Bioinformatics (CBB-S),* Xi'an China, May. 2012

Yu Wang, Maia Angelova, and Akhtar Ali **An automatic parameter selection in density weighted kernel fuzzy clustering for gene expression data analysis** *Bioinformatics* (preparation)

Abstract

With the advent of microarray technology, it is possible to monitor gene expression of tens of thousands of genes in parallel. In order to gain useful biological knowledge, it is necessary to study the data and identify the underlying patterns, which challenges the conventional mathematical models. Clustering has been extensively used for gene expression data analysis to detect groups of related genes. The assumption in clustering gene expression data is that co-expression indicates co-regulation, thus clustering should identify genes that share similar functions.

Microarray data contains plenty of uncertain and imprecise information. Fuzzy c-means (FCM) is an efficient model to deal with this type of data. However, it treats samples equally and cannot differentiate noise and meaningful data. In this thesis, motivated by the preservation of local structure, a local weighted FCM is proposed which concentrate on the samples in neighborhood. Experiments show that the proposed method is not only robust to the noise, but also identifies clusters with biological significance.

Due to FCM is sensitive to the initialization and the choice of parameters, clustering result lacks stability and biological interpretability. In this thesis, a new clustering approach is proposed, which computes genes similarity in kernel space. It not only finds nonlinear relationship between gene expression profiles, but also identifies arbitrary shape of clusters. In addition, an initialization scheme is presented based on Parzen density estimation. The objective function is modified by adding a new weighted parameter, which accentuates the samples in high density

areas. Furthermore, a parameters selection algorithm is incorporated with the proposed approach which can automatically find the optimal values for the parameters in the clustering process. Experiments on synthetic data and real gene expression data show that the proposed method substantially outperforms conventional models in term of stability and biological significance.

Time series gene expression is a special kind of microarray data. FCM rarely consider the characteristics of the time series. In this work, a fuzzy clustering approach (FCMS) is proposed by using splines to smooth time-series expression profiles to minimize the noise and random variation, by which the general trend of expression can be identified. In addition, FCMS introduces a new geometry term of radius of curvature to capture the trend information between splines. Results demonstrate that the new method has substantial advantages over FCM for time-series expression data.

Contents

List of Figures

List of Tables

List of Abbreviations

ARI	Adjusted Rand Index
BHI	Biological Homogeneity Index
BP	Biological Process
CC	Cellular Component
cDNA	Complementary DNA
CLICK	Cluster Identification via Connectivity Kernels
DNA	Deoxyribonucleic Acid
EM	Expectation Maximization
FCM	Fuzzy c-mean
FOM	Figure of Merit
KFCM	Kernel Fuzzy c-means
KNN	K-Nearest Neighbours
LFCM	Local fuzzy c-means
MF	Molecular Function
mRNA	Messenger ribonucleic acid
NGMs	Neuro-Glial markers
NTRs	Neuro-transmitter receptors
PEPS	Peptide signaling family
PDF	Parzen Density Function
RSS	Residual Sum of Squares
SOM	Self Organizing Map
SVM	Support Vector Machine

Chapter 1 Introduction

1.1 Overview

Bioinformatics is a new application of computers, mathematical and statistics models to analyses of biological data. There are two important research fields in bioinformatics: genomic analysis and proteomic analysis. Genomic analysis aims to extract information from large amounts of gene data, while proteomic analysis has an objective to determine protein functions from protein databases. The high-throughput technologies can rapidly sequence and analyze the whole genome, which supplies an opportunity to understand the complex cellular interactions. Although the sequencing of genomes has delivered many insights into their composition, it has offered a static view of genes in various organisms. Questions about the interaction of genes and the impact of environmental conditions on genetic networks remain difficult to study using sequence data only (Andreas and Francis, 2005).

Microarray techniques can simultaneously measure the expression of thousands of genes across a collection of related experiments or during biological process, which investigates the dynamic behavior of genes. Interactions in gene networks and responses to environmental changes can be monitored systematically (Andreas and Francis, 2005). One of the greatest challenges posed by microarray technologies is the analysis of the large amounts of data. Finding meaningful structures and useful information in microarray experiments is a formidable task and demands new approaches of data processing and analysis. These approaches have

to be exploratory and should not be model dependent, since only a fragment of the underlying data-producing mechanisms is known. Although biological experiments provide a wealth of information on genes and proteins, these experiments are expensive and time-consuming. Hence computational prediction methods are needed to provide valuable information for large DNA microarray data whose structures or functions cannot be determined from biological experiments.

1.2 Problem definition

The potential applications of microarray data are numerous. Functionally related genes can be detected by clustering of gene based on expression values (Page and Coulibaly, 2008). Medical applications of microarray data analysis seeks to identify genes involved in disease by comparing gene expression values between tissues of healthy and diseased individuals (Andreas and Francis, 2005). This is often accomplished by supervised learning techniques for class comparison and class prediction. Moreover, patterns of genes specifically induced in pathological tissues may be identified using clustering techniques (Andreas and Francis, 2005). Finding genes that are common to specific groups of tumors may prove useful. Such findings could offer medical researchers a starting place in their quest to improve the reliability of cancer diagnosis and treatment effectiveness. The possibility of gene targeted treatment requires one to more accurately understand the underlying genetic and environmental factors which contribute to the development of cancer (Andreas and Francis, 2005). Cluster analysis is one tool in a growing arsenal of research weapons for better understanding these relationships. In recent years, clustering methods have been used extensively in analyzing biological data, especially for DNA microarrays data. Clustering is an important technique by identifying interesting patterns in the data. A key step in the clustering process is the identification of a group of genes that manifest similar expression patterns over several conditions into clusters, thus revealing relations among genes and their functions. A cluster of genes can be defined as a set of biologically relevant genes which are similar based on a proximity measure.

Developing an effective clustering algorithm frequently involves three steps (Figure 1.1). The first step is to select effective features by identifying a subset of the original data. Irrelevant and redundant genes or conditions are excluded for further analysis. The second step is the clustering process, which utilizes a strategy to find the optimal or sub-optimal groups in the dataset. The strategy is usually based on two components: proximity measure and clustering criterion. A proximity measure quantifies the similarity between two observations, while the clustering criterion is based on the expected distribution of underlying data (such as intra homogeneity and inter separateness). The final step is cluster validation, which assesses the quality of the clusters. "Good" cluster for gene expression analysis is the one that can be biologically interpreted.

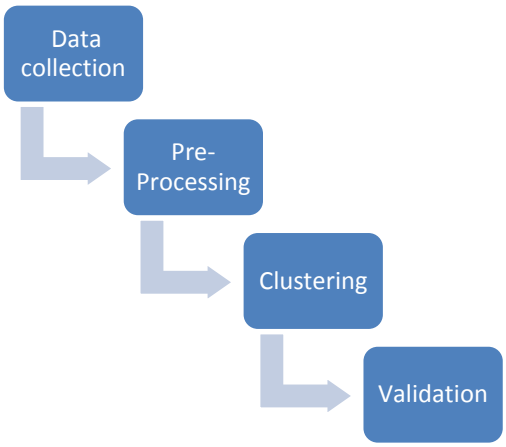


Figure 1. 1 Framework of clustering for gene expression data analysis

Data clustering analysis is a useful tool and has been extensively applied to extract information from gene expression profiles obtained by DNA microarrays. However, existing clustering approaches are mainly developed in computer science for image processing and pattern recognition, which neglects the specific characteristics of gene expression data or the particular requirements from the biological domain. Therefore, clustering result lacks of reliability and biological interpretation. Moreover, although numerous algorithms have been developed to address the problem of data clustering, these algorithms have their limitations such as determining the number of clusters, selecting the proximity measure. Alt-

hough some cluster indices address this problem, they still have the drawback of model over-fitting. Alternative approaches, based on statistics with the log-likelihood estimator and a model parameter penalty mechanism, can reduce over-fitting, but are still limited by assumptions regarding models of data distribution and by a slow convergence with model parameter estimation. Even when the number of clusters is known a priori, different clustering algorithms may provide different solutions because of their dependence on the initialization parameters. Because most algorithms use an iterative process to estimate the model parameters while searching for optimal solutions, a solution that is a global optimum is not guaranteed.

1.3 Aims and objectives

This thesis aims to investigate the performance of existing clustering techniques and thus to contribute to the development of new clustering techniques for gene expression data. The main objectives are as follows:

1. In order to evaluate the clustering performance for gene expression data, this study reviews a range of clustering techniques and evaluates theirs advantages and disadvantages.
2. Traditional FCM is sensitive to the noise. However, gene expression data involves a large component of noise, which limits its application. The local structure includes a lot of useful information which can be utilized to accentuate the meaningful data and minimize the noise influence. By preserving the local structure, a weighted FCM is proposed.
3. FCM is sensitive to the initialization. In order to solve this sensitivity and avoid FCM trapping into local minimum, an initialization scheme is proposed. Moreover, FCM utilizes Euclidean distance to calculate gene similarity. It is only effective finding spherical and equal sized clusters, which makes the results lack of biological interpretation. In order to identify general clusters, a density weighted KFCM is proposed.

4. Time series is a special kind of microarray data. However, conventional clustering methods rarely consider the characteristics of time series. In order to find useful information from this type of data, the characteristics of time series is studied and a new clustering approach is proposed which uses spline to smooth gene expressions. It not only eliminates random variable and noise, but also preserves the general trend of the expression files.

1.4 Thesis contribution

This dissertation focuses on construction of a machine learning and data mining framework for discovery cluster structure with biological significance in gene expression data. Three novel algorithms have been developed:

1. A local weighted FCM method (LFCM) is proposed, which renders LFCM immune to noise by utilizing local structure information. Experiments on artificial data and real gene expression data show that the proposed method outperforms the conventional ones.
2. A density weighted KFCM methods (DKFCM) is developed, which incorporates an automatic parameter selection to find the optimal values in the clustering process. This method detects arbitrary shapes of clusters and the clusters are of biological significance in gene expression data analysis.
3. A FCMS method is developed for time series gene expression data, which utilizes spline to smooth gene expression data, and adopts a new proximity measure to compute genes similarity. Experiments show that the proposed method can identify distinct and accurate patterns, which offers biologists an efficient way to understanding the data.

1.5 Thesis structure

The thesis is organized as follows:

Chapter 1 gives an overview of the thesis. The problem definition, research objectives and thesis contribution are all addressed.

Chapter 2 discusses the biological foundation of research background, such as gene theory, microarray technology etc. Literatures on the clustering techniques and validation measures for gene expression analysis are also reviewed.

Chapter 3 describes the dataset used in this research. Data preprocessing, such as missing values estimation, filtering and standardization are discussed.

Chapter 4 gives an introduction to the FCM and KFCM. KFCM not only finds the nonlinear relationship between genes, but also detects arbitrary shapes of clusters. A full comparison of FCM and KFCM with other popular algorithms is run using artificial data and real gene expression data.

Chapter 5 reveals the limitations of FCM, which assigns equal weights to genes without consideration of their contributions to the clustering process. This treatment makes the results lack accurate and biological interpretation. A local FCM is proposed by assigning different weights to genes according to their contribution to the clustering. Experiments show that the proposed method achieves better performance than the conventional ones.

Chapter 6 proposes a density weighted KFCM approach. An initialization method is presented based on Parzen density estimation. In addition, the objective function is amended by adding a new weighted parameter to accentuate the objects in high density area. Furthermore, a parameter optimization is presented which automatically finds the optimal values in the clustering process. Experiments on synthetic data and real gene expression data show that proposed method substantially outperforms conventional models.

Chapter 7 describes the characteristics of time series gene expression data. In or-

der to minimize the noise influence and identify the trend change, cubic spline is used to smooth gene expression. FCM is then used to cluster the splines based on radius of curvature. Experiments results show that the proposed method has better performance than conventional FCM.

Chapter 8 draws a conclusion of the thesis. Some of the major challenges laying ahead in the analysis of gene expression data are discussed.

Chapter 2 Research Background and Literatures Review

2.1 Microarray and gene expression data

Proteins are the major active elements of cells. They perform many key functions of biological systems and they are the structural building blocks of cells and tissues (Andreas and Francis, 2005). The information for producing the proteins required in a cell under a particular condition is contained in the deoxyribonucleic acid (DNA), and the complete DNA sequence of a living organism, the genome, is organized into chromosomes and genes. The production of protein from DNA is divided into two main steps. In step one, known as transcription, single stranded messenger ribonucleic acid (mRNA) is copied from the DNA, and in the second step, known as translation, proteins are produced based on information from the mRNA. This is illustrated as,

DNA → mRNA→ Protein

Gene expression analysis is the study of mRNA levels transcribed from DNA. In contrast to DNA which is static over the life-time and cells of a living organism, mRNA level varies over time and between cell types. It also varies within cells under different conditions (Andreas and Francis, 2005). For example, the amount of mRNA transcribed from a gene in a healthy organism can differ from the amount of mRNA transcribed from the same gene in the corresponding cell type of a sick organism. Therefore, this gene is differentially expressed between the two conditions healthy and sick.

Microarray is considered as an important tool for advancing the understanding of the DNA information, molecular mechanisms, and pathophysiology of critical illness. By microarray, the expression of thousands of genes can be assessed and complex pathways can be more fully evaluated in a single experiment. Microarrays are based on the fundamental principle of base-pair complementarity of nucleic acids. Since the binding of different nucleotide strands occurs independently, base-pair complementarity allows parallel probing of complex mixtures of gene transcripts (Andreas and Francis, 2005). There are two major platforms on which microarray experiments are performed: Affymetrix and complementary DNA (cDNA). The primary difference between these designs is that the cDNA approach uses a single long stretch of DNA for each gene while the Affymetrix approach uses several short oligonucleotides to probe for each gene (Andreas and Francis, 2005). The cDNA technology measures the relative gene abundance from two samples while the Affymetrix technology measures the absolute gene abundance for a single sample. In the oligonucleotide arrays, each gene is represented by multiple probes of length 20 bp (Andreas and Francis, 2005). These probes are synthesized base by base and are placed in hundreds of thousands of different positions on a glass plate, using photolithography. The arrays are then scanned and the quantitative fluorescence image along with the known position of the probes is used to assess whether a gene is present and its abundance. In the oligonucleotide arrays the fluorescence image is an absolute measure of the abundance of mRNA of a sample. Using solid surfaces to attach cDNAs or oligonucleotides, whole genomes can be studied with a single array. Parallel measurement of gene activities overcomes the limitations of the traditional gene-by-gene approach as whole networks of interacting genes can be readily studied.

For the cDNA microarray, the DNA from thousands of genes is spotted onto a small glass slide in a regular pattern. Each spot or probe interrogates for a specific gene. Probes are generated by amplifying genomic DNA with gene specific primers. The probes are spotted onto the slide automatically by a robot (Andreas and

Francis, 2005). mRNA from the samples is purified and reverse transcribed to cDNA with fluorescent labeled nucleotides. If two samples are used (*e.g.* control and treatment), they are labeled separately with the fluorescent dyes Cyanine-3 (Cy3) and Cyanine-5 (Cy5), which emit light in different spectrums. The spectrums are assigned the colors green (Cy3) and red (Cy5) for convenience. The labeled cDNA is mixed in equal amounts and hybridized to the array (Figure 2.1). Unbound cDNA is washed away and the array is scanned twice with a laser, generating one red and one green image (Andreas and Francis, 2005).

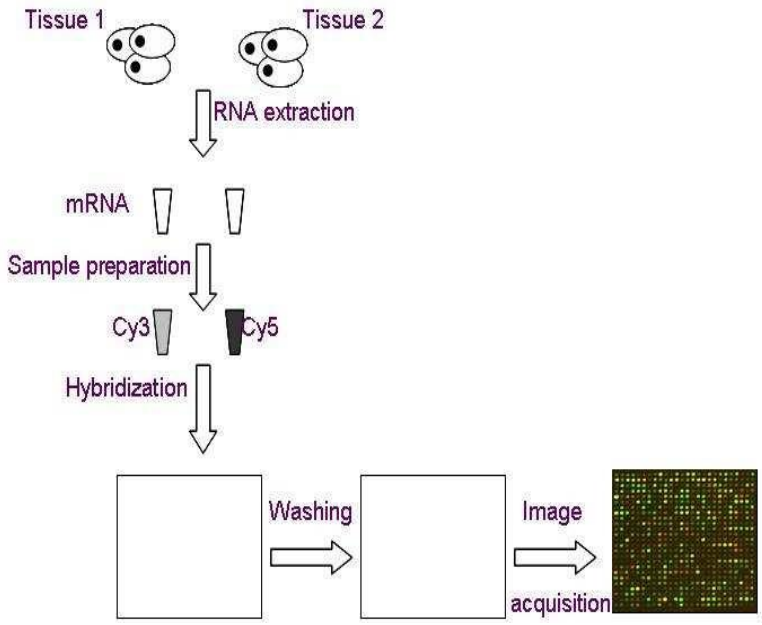


Figure 2. 1 Steps in a microarray experiment
The Cy3 and Cy5 in the diagram refer to the mRNAs dyed using the two fluorescent dyes of Cy3 and Cy5.

Once the images are overlaid, spots hybridized with equal amounts of control and treatment cDNA are yellow, while spots for genes that ar differentially expressed are different shades of red or green (Andreas and Francis, 2005). The cDNA microarray image is illustrated in Figure 2.2. Various image analysis techniques are employed to identify the red and green intensities in the spots along with the surrounding background. Since the spot size and hybridization properties change for different nucleotide sequences, the measured fluorescence intensity cannot be translated to an absolute level of mRNA. The ratio between the amounts of gene specific mRNA in the two samples is called a fold difference, which is often in-

terpreted as evidence that the gene is differentially expressed.

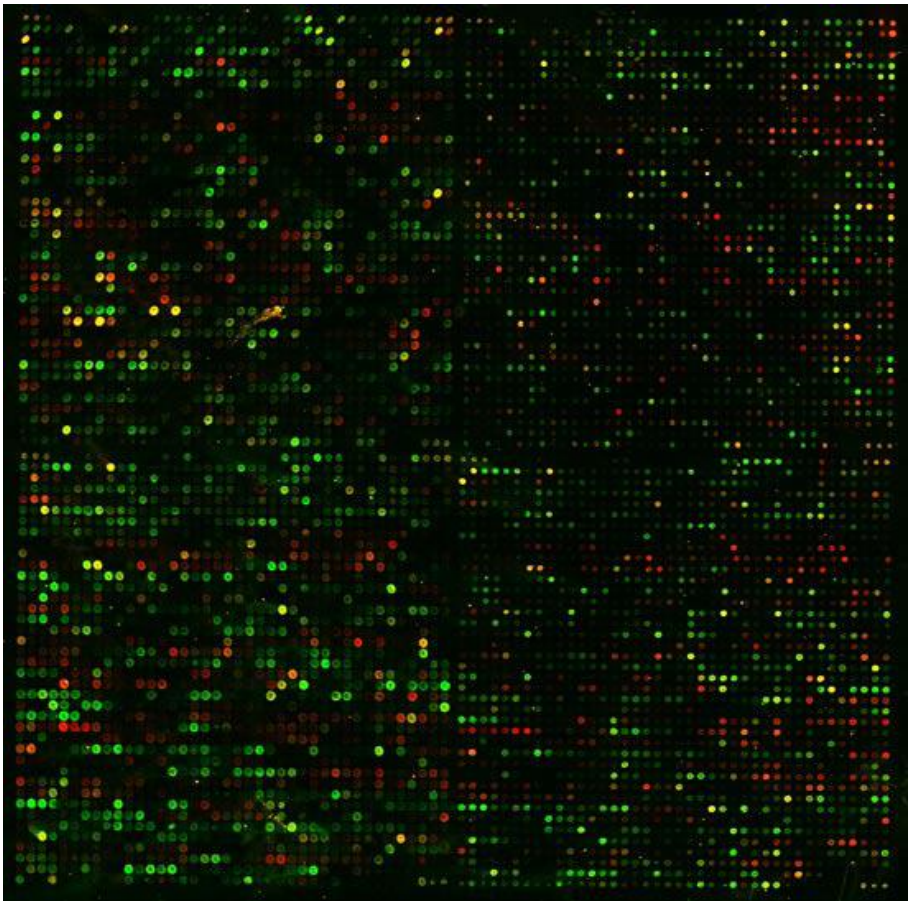


Figure 2. 2 Scan of a cDNA microarray containing the whole yeast genome (Ben-Dor et al.,1999)

2.2 Clustering Algorithms

Generally, clustering has two main applications for gene expression analysis: gene based clustering and sample based clustering. In gene-based clustering, the genes are treated as the objects, while the samples are the features. While, the samples based clustering regards the samples as the objects and the genes as the features, it partitions samples into homogeneous groups. Each group may correspond to some particular macroscopic phenotype, such as clinical syndromes or cancer types (Golub *et al*. 1999). The distinction of gene based clustering and sample based clustering is based on different characteristics of clustering tasks for gene expression data. In this research, only gene-based clustering is considered.

DeRisi *et al*. (1996) initially revealed expression patterns when they studied the gene expression data of *Yeast cell cycle*. In order to infer the function of novel genes, they employed clustering analysis by grouping them with genes of well-known functionality. This is based on the observation that genes showing similar expression patterns (co-expressed genes) are often functionally related and are controlled by the same regulatory mechanisms (co-regulated genes). Expression clusters are frequently enriched by genes of certain functions *e.g. DNA replication*, or *protein synthesis*. If a gene of unknown function falls into such a cluster, it is likely to serve the same functions as other members of the cluster. This method enables assigning possible functions to a large number of genes by clustering of co-expressed genes (Chu *et al*. 1998). Analysis of cluster structure can further identify the underlying mechanisms of metabolic and regulatory networks in the cell. It is especially valuable for organism and cell types where little previous knowledge about their biology exists.

Sample based clustering takes samples as objects and genes as features. It helps to understand gene regulation, metabolic and signaling pathways, the genetic mechanisms of disease, and the response to drug treatments. For instance, if overexpression of certain genes is correlated with a certain cancer, it is promising to explore which other conditions affect the expression of these genes and which other genes have similar expression profiles. It is also valuable to investigate compounds (potential drugs) lower the expression level of these genes. Alizadeh *et al*. (2000) applied a clustering algorithm to large B-cell lymphoma using 96 samples of normal and malignant lymphocytes and found that there is diversity in gene expression among the tumors of diffuse large B-cell lymphoma patients. They identified two molecularly distinct forms of diffuse large B-cell lymphoma, which had gene expression patterns indicative of different stages of B-cell differentiation. Interestingly, these two groups correlated well with patient survival rates, thus confirming that the clusters are meaningful. Ayano *et al*. (2013) employed clustering to differentiate genetic lineages of undifferentiated-type gastric carci-

nomas analysed of genomic DNA microarray data. The goal of sample based clustering is to find the phenotype structures or substructures of the sample. Although the conventional clustering methods, such as k-means, SOM, hierarchical clustering can be directly applied to cluster samples using all the genes as features, the irrelevant genes may seriously degrade the quality and reliability of clustering results (Xing and Karp, 2001). Thus, particular methods should be applied to identify informative genes and reduce gene dimensionality for clustering samples to detect their phenotypes. In this study, sample based clustering is out of the research scope.

Recently, many methods for cluster analysis have been proposed, such as k-means, hierarchical clustering, self-organizing maps, and graph theoretic approaches. These algorithms are also applied to analysis of microarrays.

(1) k-means

The k-means is the most widely used clustering method, which partitions *n* objects into *k* clusters, and each objects belongs to its nearest cluster centres (Steinley, 2006). The objective function is,

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} d_{ij}^2(x_j, v_i) \tag{2.1}$$

where *C* and *N* denote the number of clusters and objects respectively, x_j in the *j* th objects. $d_{ij}^2(x_j, v_i)$ is the distance between vector x_j and prototype v_i.

$$d_{ij}^2(x_j, v_i) = \arg\min\left\{d_{ij}^2(x_j, v_i)\right\} \tag{2.2}$$

The k-means aims at minimizing the intra-cluster distance, which starts with selection of *k* cluster centres. Then, each object in the dataset is assigned to the closest cluster. After that, the cluster centres are recalculated according to the associated objects. This process is repeated until convergence is achieved. Figure 2.3 is an illustration of the process of k-means, in the first iteration; three initial cluster centres are randomly selected with symbols "+", and it converges to the

minimum in the sixth iteration.

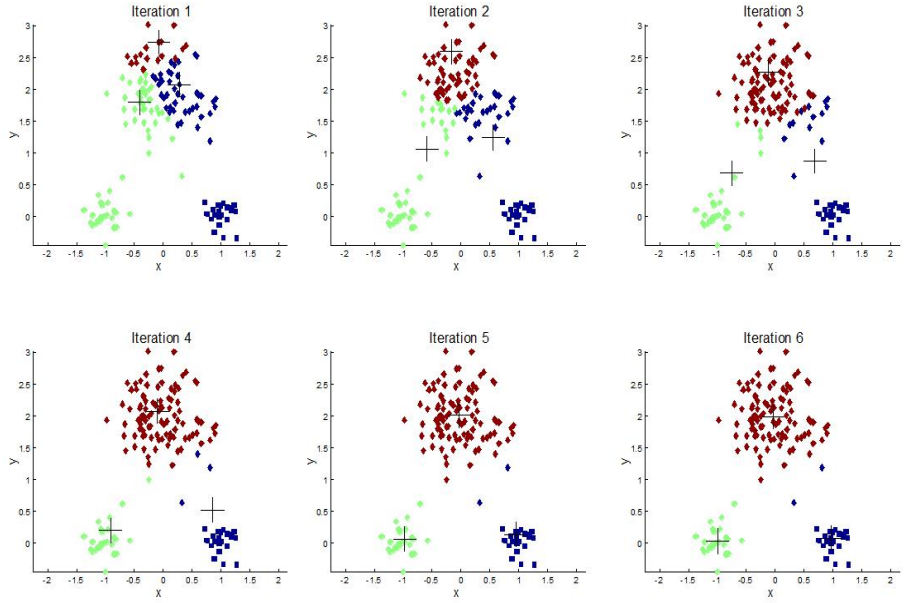


Figure 2. 3 The illustration of the k-means process with six iterations steps

The k-means algorithm is easy to implement and its time complexity is suitable for large datasets. However, k-means need user to specify the number of clusters which is usually unknown in advance. In order to detect the optimal number of clusters, users usually run the algorithms repeatedly with different values of *k* and compare the clustering results. For large gene expression dataset which contains thousands of genes, this extensive parameter fine-tuning process may not be practical. Moreover, k-means forces each gene into a cluster, this compulsive strategy may cause the algorithm to be sensitive to noise (Steinley, 2006). Finally, k-means leads to local minimum of the objective function, thus the clustering result depends on the initiation. To reduce the influence of the initial partition on the clustering result, one can run the algorithm multiple times then choose the result that has the minimal cost function.

Recently, many advanced clustering algorithms based on k-means have been proposed to overcome the drawbacks. Steinley (2003) gave a deep discussion on the *k*-means local optima and proposed a solution to avoid the algorithm trapping in local optima. However, the qualities of clusters in gene expression datasets vary widely. Thus, it is difficult to choose the appropriate globally-constraining param-

eters. Similarly, Tseng (2007) proposed a penalized and weighted k-means for clustering with scattered objects, which have been applied in gene expression data, and can find tightly meaning clusters. Genetic algorithm was combined with k-means for clustering large-scale microarray data (Wu, 2008). Compared with the original approach, the new method is able to capture clusters with complex and high-dimensional structures accurately. Iam-On and Boongoen (2012) presented a new weighted k-means for microarray data by learning the local structure, which can effectively identify the cancer relating genes and aid the biologists to find the subcategories of cancer.

(2) Hierarchical clustering

There are two basic types of hierarchical clustering methods: agglomerative and divisive clustering. The agglomerative method starts with taking each object as a cluster and merges objects into groups according to their similarities. In the first iteration, the most similar objects are grouped together and merged. In the final iteration, all of the objects are contained in a single large cluster. In each iteration, the method fuses the objects which are the most similar. Figure 2.4 shows the agglomerative clustering process and the tree structure. According to the similarities between clusters, agglomerative method can be divided into: *single linkage*, *complete linkage* and *average linkage*. The *single linkage* method considers the shortest pairwise distance between objects in two different clusters as the distance between the two clusters (Guess and Wilson, 2002). While the *complete linkage* method defined the distance as the most distant pair of objects (Guess and Wilson, 2002). As the *average linkage* clustering, the average of the pairwise distances between all pairs of objects coming from each of the two clusters is taken as the distance between two clusters (Guess and Wilson, 2002). On the contrary, divisive method starts with all of the objects contained in one large cluster. In each iteration, the groups are subdivided or kept in the same cluster based on how close the individuals within clusters are in terms of their similarity measures. Eventually, there are as many clusters as individuals (Guess and Wilson, 2002).

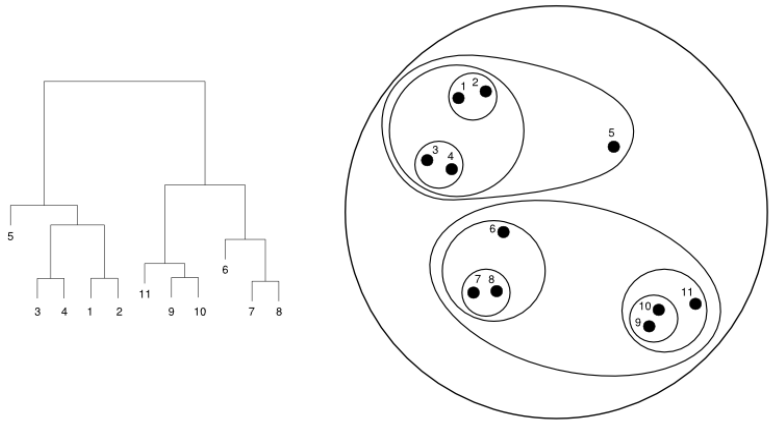


Figure 2. 4 An illustration of Hierarchical clustering process

Compared with *k*-means, hierarchical approach is not sensitive to initialization and is robust to the noise (Guess and Wilson, 2002). In addition, the number of clusters needs not to be specified a priori and it can graphically represent the dataset, which provides the user a thorough impression of the data distribution. However, the clustering procedure is static, patterns assigned to a cluster cannot move to another cluster. It fails to separate overlapping clusters due to a lack of information about the global shape or size of the clusters (Boratyn *et al*., 2006).

Lewis, and Noble (2003) describe a generalization of the hierarchical clustering algorithm that efficiently incorporates high-order features by using a kernel function to map the data into a high dimensional feature space. Makretsov *et al.* (2004) used a hierarchical clustering in tissue microarray immunostaining data and successfully identifies prognostically significant groups of breast cancer. Inamura *et al*. (2005) combines non-negative matrix factorization and hierarchical clustering to find two subclasses of lung squamous cell carcinoma with different gene expression profiles and prognosis. Boratyn *et al.* (2006) used certain biological knowledge to supervise the hierarchical clustering, and produced clusters with more biological significance, which outperformed the conventional hierarchical clustering algorithms substantially. Trevino *et al.* (2012) presented a hierarchical clustering for *Escherichia coli* data and identify distinct gene expression patterns. Badsha et al. (2013) proposed a robust hierarchical clustering method with respect

to the noise by maximizing the β-likelihood function for a gene expression data.

(3) Self-organizing map

Self-organizing map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space (Kohonen, 1984). Initially SOM constructs geometry of nodes (a 7 x 9 grid in Figure 2.5). Each node is associated with a reference vector, and the input vectors are mapped to the node with the closest reference vector. The location of the nodes is iteratively adjusted by moving in the direction to the dense areas of the input vector space (Torkkola *et al.*, 2001). Once the algorithm has proceeded through a user defined number of iterations, it terminates with similar objects grouped around a specific node.

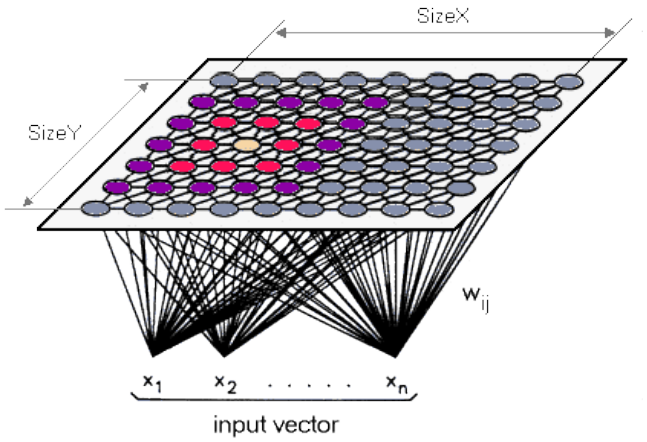


Figure 2. 5 Self-organizing map (Kohonen, 1984)

Torkkola *et al*. (2001) applied SOM to exploratory analysis of microarray data and found SOM not only enabled quick selection of the gene families identified in previous work, but also facilitated the identification of additional genes with similar expression patterns. Covell *et al.*(2003) used SOM analysis of microarray data for molecular classification of cancer. Similarly, Hautaniemi *et al*. (2003) applied SOM to analysis and visualization of gene expression microarray data of human cancer and found a set of potential predictor genes for classification purposes. Comparison and visualization of the effects of different drugs is straightforward with SOM (Dragomir et al., 2004). Dragomir *et al* (2004) used SOM to explore

the microarray data by incorporating independent component analysis (ICA) to reduce data features. Wu *et al*. (2005) proposed a hybrid SOM-SVM approach for the zebrafish gene expression analysis by utilizing a small portion labeled genes to train the model, and then clustering pattern, results showed that this method is capable of finding certain biologically meaningful clusters.

(4) Graph-theoretical algorithm

Given a dataset *X*, a proximity matrix *P* can be constructed where $P(x, y) = proximity(x, y)$ and a weighted graph $G(V, E)$, where each data corresponds to a vertex. For clustering methods, each pair of objects is connected by an edge with weight assigned according to the proximity value between the objects. Graph-theoretical clustering techniques are explicitly presented in terms of a graph, thus converting the problem of clustering a dataset into such graph theoretical problems as finding minimum cut or maximal cliques in the proximity graph *G*. Cluster Identification via Connectivity Kernels (CLICK) is a representive graph-theoretical algorithm which has been successfully used for gene expression analysis and produced high quality clusters (Roded *et al.*, 2003). CLICK seeks to identify highly connected components in the proximity graph as clusters. CLICK makes the probabilistic assumption that after standardization, pairwise similarity values between elements are normally distributed. Under this assumption, the weight ω_{ij} of an edge is defined as the probability that vertices *i* and *j* are in the same cluster. The clustering process of CLICK iteratively finds the minimum cut in the proximity graph and recursively splits the data set into a set of connected components from the minimum cut. CLICK also takes two post-pruning steps to refine the cluster results. The adoption step handles the remaining singletons and updates the current clusters, while the merging step iteratively merges two clusters with similarity exceeding a predefined threshold (Roded *et al.*, 2003). Clusters obtained by CLICK demonstrated better quality in terms of homogeneity and separation. However, CLICK has little guarantee of not going astray and generating highly unbalanced partitions, e.g., a partition that only separates a few noises from the remaining data objects (Roded *et al.*, 2003). Furthermore, in gene expression

data, two clusters of co-expressed genes may be highly intersected with each other. In such situations, CLICK is unable to identify the two clusters and reported as one highly connected component (Roded *et al.*, 2003).

(5) Model-based clustering

The mixture models approach assumes that the data are from a mixture of a specified number of groups in various proportions. By assuming a parametric form for the density function in each group, a likelihood function can be formed in terms of a mixture density (Dempster *et al.*, 1977). The unknown parameters of the distribution can be estimated by the method of maximum likelihood. This process leads to estimates of cluster specific parameters as well as the proportion of observations falling in each cluster and the posterior probability of each observation falling in a specific cluster. Clustering proceeds by assigning each object to a group based on the relative value of the estimated posterior probability of belonging to that group compared with the posterior probabilities of belonging to the other groups.

The Expectation/Maximization (EM) algorithm is a two-stage iterative algorithm, which consists of expectation and maximization steps (Dempster *et al.*, 1977). The expectation step estimates the data by calculating expected values conditional on the observed data. Once the data are estimated in the expectation step, the maximum likelihood estimates of the parameters are calculated in the maximization step. The EM algorithm requires starting values for the parameter estimates to be input for the first expectation step (Dempster *et al*., 1977). One of the advantages of the mixture models for analyzing microarray data is that they provide a statistical criterion for assessing the number of clusters present in the data. A strong assumption made in fitting mixture models to microarray data is that the genes are independent and identically distributed according to the mixture density. However, the EM algorithm converges slowly, particularly at regions where clusters overlap and requires the data distribution to follow some specific distribution model (Yeung *et al*. 2001a). Moreover, gene expression data is likely contain

overlapping clusters and do not always follow standard distributions, (e.g., Gaussian), which is inappropriate for such kind of data.

Given various clustering methods, however, there is no one clustering algorithm that performs significantly better than the others when tested across multiple datasets. This is due to microarray data tends to have complex biological system and diverse structures (Andreas and Francis, 2005).

2.3 Validation measures

Different methods frequently yield different clustering results. Thus, a fair comparison between alternative clustering methods is necessary. However, there is no benchmark for the critical assessment of any clustering approach in the field of gene expression data analysis. In order to ensure the quality of the clustering algorithm and compare the clustering performance, many validation methods have been proposed based on statistical models for comparison of clustering approaches with a controlled number of clusters. Generally, they can be classified into three categories: internal measure, external measure, and biological measure.

2.3.1 Internal measure

Internal measures assess the quality of the clustering approaches using intrinsic information of the dataset. Silhouette index and Figure of merit are selected as the internal validation in this work.

Silhouette index

Silhouette index is a measure of tightness and separation of clusters, which is used to assess the level of statistical significance of clusters. For a given cluster X_j ($j = 1, ..., c$), this method assigns each sample X_j a quality measure, $s(i)$ ($i = 1, ..., m$), known as the Silhouette width. The Silhouette width is a confidence

indicator on the membership of the $i\,th$ sample in cluster X_j. The Silhouette width for the $i\,th$ sample in cluster X_j is defined as,

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.3}$$

where $a(i)$ is the average distance between the $i\,th$ object and all of the objects included in X_j, and $b(i)$ is the minimum average distance between the $i\,th$ sample and all of the samples clustered in X_k $(k = 1, ..., c; k \neq j)$, and this formula follows that $-1 \leq s(i) \leq 1$. $s(i)$ closing to 1 indicates that the $i\,th$ object has been well clustered, *i.e.* it was assigned to an appropriate cluster. $s(i)$ closing to zero suggests that the i th sample could be assigned to the neighboring cluster. If $s(i)$ is close to –1, one may argue that a object has been misclassified. Thus, for a given cluster, X_j $(j = 1, ..., c)$, it is possible to calculate a cluster Silhouette S_j, which characterizes the heterogeneity and isolation properties of such a cluster,

$$S_j = \frac{1}{m}\sum_{i=1}^{m} s(i) \tag{2.4}$$

where m is the number of samples in S_j. It has been shown that for any partition $U : X_1 \cup ....X_i \cup ....X_c$, a Global Silhouette value, GS_u, can be used as an effective validity index,

$$GS_u = \frac{1}{c}\sum_{j=1}^{c} S_j \tag{2.5}$$

The partition with the higher GS_u is taken as the optimal partition. It is should be note that the internal validation has a limitation that clusters are validated using the intrinsic information from the data.

Figure of merit

Figure of merit (FOM) reveals the reliability of the resulting clusters, which indicates the probability of the clusters is not formed by chance. It is based on the

concept that if a clustering result reflects true cluster structure, then a predictor based on the resulting clusters should accurately estimate the cluster labels for new test samples. For gene expression data, extra data objects are rarely used as test samples, since the number of available samples is limited. Rather, a cross-validation method is applied. The generated clusters are assessed by repeatedly measuring the prediction strength with one or a few of the data objects left out in turn as "test samples" while the remaining data objects are used for clustering. Intuitively, genes within the same clusters are expected to have similar expression levels, while genes in disjoint clusters are expected to be relatively far apart from each other. Therefore, FOM is the ratio of the within-cluster dispersion to the between-cluster separation. The FOM is defined as follows,

$$FOM(e,k) = \frac{\frac{1}{n}\sum_{i=1}^{k}\sum_{x\in C_i}\left|R(x,e) - \mu C_i(e)\right|}{\frac{1}{k-1}\left(\mu_{C_i}^{max}(e) - \mu_{C_i}^{min}(e)\right)} \qquad (2.6)$$

where n is the number of genes, k is the number of clusters, $R(x,e)$ is the expression level of gene x under condition e and $\mu C_i(e)$ is the average expression level in condition e of genes in cluster C_i. The FOM measures the mean deviation of the expression levels of genes in e relative to their corresponding cluster means. Thus, a small value of FOM indicates strong prediction strength, and therefore a high level reliability of the resulting clusters.

2.3.2 External validation

External validation assesses the degree of consensus between the clustering result and the class labels. Rand index (RI) is a measure of agreement between two partitions: one is the clustering result and the other is the standard partition (Given by external information). The RI computes the proportion of the total observation pairs that agree (Everitt *et al.*, 2001). Agreement means that either both of the observations in the pair fall into the same cluster according to both partitions or both observations fall into different clusters according to both partitions. Suppose T is

the true clustering of a gene expression data based on biological knowledge and *C* is a clustering result given by certain clustering algorithm. Let *a* denote the number of gene pairs belonging to the same cluster in both *T* and *C*, *b* is the number of pairs belonging to the same cluster in *T* but to different clusters in *C*, *c* is the number of pairs belonging to different clusters in *T* but to the same cluster in *C* and *d* is the number of pairs belonging to different clusters in both *T* and *C*. The Rand index (RI) is computed by,

$$RI(T,C) = \frac{a+d}{a+b+c+d} \tag{2.7}$$

The RI has an expected value slightly greater than 0. If the partitions agree perfectly, the RI is 1. The value of RI varies from 0 to 1 and higher value indicates that the clustering result is more similar to the standard partitions. However, a major problem with the RI is that the expected value of two random partitions does not take a constant value.

Adjusted Rand index (ARI) is proposed by Hubert and Arabie (1985) and it is more sensitive than the RI. ARI assumes the generalized hyper geometric distribution as the model of randomness. The general form of ARI is,

$$ARI(T,C) = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)} \tag{2.8}$$

ARI has an expected value of zero and ranges from -1 to 1. where negative values indicate poor clusters, while positive values means significant clustering methods. When ARI attains 1.0, the clustering method is the perfect. The RI and ARI are frequently used to assess the quality of clusters for microarray data (Yeung *et al*., 2001; Yeung and Ruzzo, 2001).

The following example illustrates the computation of RI and ARI. Given a dataset including 5 items and 2 clustering partitions:

Clustering A: 1, 2, 2, 1, 1

Clustering B: 2, 1, 2, 1, 1

In order to calculate the parameters *a*, *b*, *c*, and *d*, all possible pairs are listed corresponding to the elements of the two clustering. In total, there are $C_5^2 = 10$ possible pairs:[1; 2]; [1; 3]; [1; 4]; [1; 5]; [2; 3]; [2; 4]; [2; 5]; [3; 4]; [3; 5]; [4; 5]. For example, the term of [2; 3] is correspond to the pair (2; 2) in clustering A and (1; 2) in clustering B. The parameters are computed by,

$$\begin{cases}\text{CA, [1,2] -> (1,2) -> } 1 \neq 2 \\ \text{CB, [1,2] -> (2,1) -> } 2 \neq 1\end{cases} \Rightarrow \text{d = + 1} \quad \begin{cases}\text{CA, [1,3] -> (1,2) -> } 1 \neq 2 \\ \text{CB, [1,3] -> (2,2) -> } 2 = 2\end{cases} \Rightarrow \text{b = + 1}$$

$$\begin{cases}\text{CA, [1,4] -> (1,1) -> } 1 = 1 \\ \text{CB, [1,4] -> (2,1) -> } 2 \neq 1\end{cases} \Rightarrow \text{c = + 1} \quad \begin{cases}\text{CA, [1,5] -> (1,1) -> } 1 \\ \text{CB, [1,5] -> (2,1) -> } 2\end{cases} \Rightarrow \text{c = +}$$

$$\begin{cases}\text{CA, [2,3] -> (2,2) -> } 2 = 2 \\ \text{CB, [2,3] -> (1,2) -> } 1 \neq 2\end{cases} \Rightarrow \text{c = + 1} \quad \begin{cases}\text{CA, [2,4] -> (2,1) -> } 2 \neq 1 \\ \text{CB, [2,4] -> (1,1) -> } 1 = 1\end{cases} \Rightarrow \text{b = + 1}$$

$$\begin{cases}\text{CA, [2,5] -> (2,1) -> } 2 \neq 1 \\ \text{CB, [2,5] -> (1,1) -> } 1 = 1\end{cases} \Rightarrow \text{b = + 1} \quad \begin{cases}\text{CA, [3,4] -> (2,1) -> } 2 \neq 1 \\ \text{CB, [3,4] -> (2,1) -> } 2 \neq 1\end{cases} \Rightarrow \text{d = + 1}$$

$$\begin{cases}\text{CA, [3,5] -> (2,1) -> } 2 \\ \text{CB, [3,5] -> (2,1) -> } 2\end{cases} \Rightarrow \text{d =} + \quad \begin{cases}\text{CA, [4,5] -> (1,1) -> } 1 = 1 \\ \text{CB, [4,5] -> (1,1) -> } 1 = 1\end{cases} \Rightarrow \text{a = + 1}$$

According to equation (2.7) and (2.8), RI and ARI can be computed by,

$$RI = \frac{a+b}{a+b+c+d} = \frac{1+3}{1+3+3+3} = 0.4 \tag{2.9}$$

$$ARI = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} = \frac{2\times(1\times 3 - 3\times 3)}{(1+3)(3+3) + (1+3)(3+3)} = -0.2 \tag{2.10}$$

2.3.3 Biological validation

Biological validation evaluates the ability of a clustering algorithm to produce biologically meaningful clusters. Biological homogeneity index (BHI) measures how homogeneous the clusters are biologically (Datta and Datta, 2006). Consider $B = \{B_1, B_2, B_3 \ldots\ldots B_F\}$ be a set of *F* functional classes, not necessarily disjoint, and *B(i)* be the functional class containing gene *i* (with possibly more than one functional class containing *i*). Similarly, *B(j)* is the function class containing gene

j, if *B(i)* and *B(j)* match (any one match is sufficient in the case of membership to multiple functional classes), it indicates that the two genes have similar biological function by assigning an indicator $I(B(i)=B(j))=1$ if else, $I(B(i)=B(j))=0$. Intuitively, it is expected that genes placed in the same statistical cluster also belong to the same functional classes. Then, for a given clustering partition $C=\{C_1,C_2,C_3.....C_K\}$ and set of biological classes *B*, the *BHI* is defined as,

$$BHI(C,B)=\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k(n_k-1)}\sum_{i\neq j\in C_k} I(B(i)=B(j)) \qquad (2.11)$$

where $n_k=n(C_k\cap B)$ is the number of annotated genes in statistical cluster C_k. The BHI is in the range [0, 1], with larger values corresponding to more biological homogeneous clusters.

2.4 Pre-processing for Microarray Data

2.4.1 Data preparation

The aim of microarray experiments is to investigate the activity patterns of genes. However, microarrays do not assess gene activities directly, but by measuring the fluorescence intensities of labeled target cDNA hybridized to probes on the array. Generally, the first step in the analysis of microarray data is the transformation of the fluorescence signals into quantities for gene expression analysis. Although the ratio (intensities of two signals ratio) provides an intuitive measure of expression changes, it has the disadvantage of treating up and down regulated genes differently. For example, genes up regulated by a factor of 2 have an expression ratio of 2, while those down regulated by the same factor have an expression ratio of (–0.5). The most widely used transformation of the ratio is the logarithm base 2, which has the advantage of producing a continuous spectrum of values and treating up and down regulated genes in a similar fashion. Recall that logarithms treat numbers and their reciprocals symmetrically: $\log_2(1) = 0$, $\log_2(2) = 1$, $\log_2(1/2) = -1$, $\log_2(4) = 2$, $\log_2(1/4) = -2$, and so on. The logarithms of the expression ratios

are also treated symmetrically, so that a gene up regulated by a factor of 2 has a $\log_2$(ratio) of 1, a gene down regulated by a factor of 2 has a $\log_2$(ratio) of −1, and a gene expressed at a constant level (with a ratio of 1) has a $\log_2$(ratio) equal to zero. For the remainder of the dissertation, $\log_2$(ratio) will be used to represent expression levels.

Gene expression data from a microarray experiment can be represented by a real-valued expression matrix (Figure 2.6), where the rows represent expression patterns of genes, the column represent the expression profiles of samples, and each cell is the measured expression level of a gene in certain sample.

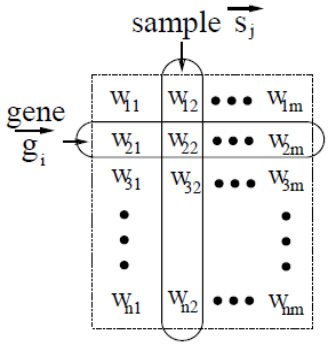


Figure 2. 6 Gene expression matrix

2.4.2 Missing values

In gene expression analysis, complete information is preferred throughout the experiment. Unfortunately, real datasets frequently have missing values, which is caused by errors or by random noise. For example, sensor failures in a control system may cause the system to miss information. The missing values can also come from the platform level, such as insufficient resolution, image corruption, spotting, scratches or dust on the slide, or hybridization failure. According to Cho *et al*. (1998), *Yeast cell cycle* data contained over 6000 missing measurement values that accounts for 6% of the total expression values. However, most clustering algorithms do not allow for missing values. In order to adapt to the clustering algorithms, two approaches have been widely used to address the problem of miss-

ing values. The simple one is to ignore the genes including missing values. This measure is adopted when the proportion of incomplete data is small, but the elimination brings a loss of information. The other one is the imputation-based approach, which supplies missing values by certain means of approximation. *K* nearest neighbors (*knn*) is a popular method to compute the missing values. A missing value of gene *i* at time point *t* is estimated by the average values for time *t* of the several nearest neighboring genes *j*. The distance was calculated by,

$$d(g_i, g_j)^2 = \frac{n}{n-m}\sum_k (g_{ik} - g_{jk})^2 \qquad (2.12)$$

where g_i is the gene expression vector for gene *i*, g_j is the gene expression vector for neighboring gene *j*, *n* is the number of arrays in the time-course experiment and *m* is the number of measurements which are missing for gene *i* or *j* or both. The sum includes only measurements for which both gene expression values (g_{ik}, g_{jk}) are present. This procedure exploits the high correlation between genes in expression data. It assumes that genes which are well correlated for existing measurements are also correlated for missing measurements. According to Troyanskaya *et al*., (2001), *knn* is an effective measure to estimate the missing values in gene expression data.

2.4.3 Filtering

In one biological process, not all genes show obvious variation according to different experiment conditions. Genes expressed at low levels or show small changes are useless for clustering. Involving these data in clustering process will not only increase the redundancy of the data, but also decrease the quality of the result. In order to identify the potential pattern effectively, most clustering approaches include a filtering step to remove these genes.

The commonly used method for gene filtering is based on the variability of gene expression values for a given gene. Genes whose expression values do not change

by more than a specified value across the samples are filtered out. The logic behind this type of filtering is that gene expression values for a gene active in a specific biological process should change at some point. Filtering based solely on variability works well in some cases.

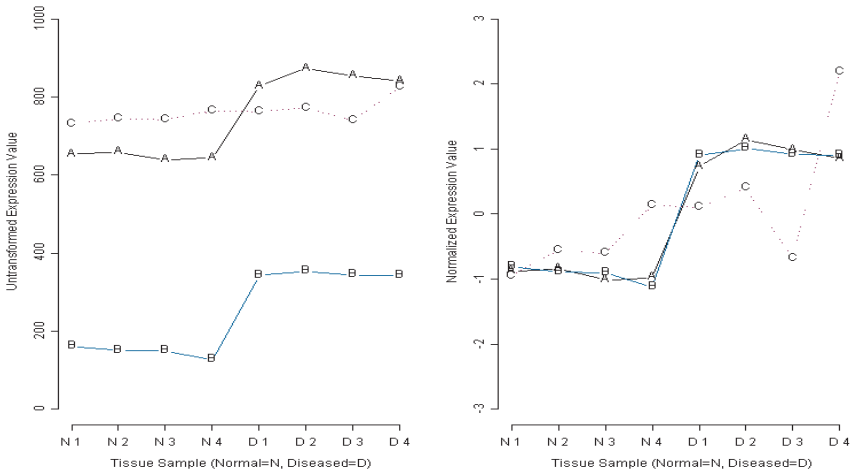
However, no consideration is given to the baseline levels of gene expression. Genes that are naturally lowly expressed will have small variances. These genes could be incorrectly removed if a small change in expression values is biologically significant. Specifying a single variance threshold for determining whether or not to keep a gene implies the assumption that all of the genes have similarly scaled variances. This may not always be true. For example, a gene with a large mean and variance for its expression level is of less interest than a gene with a small mean and the same large variance. One might consider using a filtering technique based on the coefficient of variation in this situation. The coefficient of variation gives a measure of the variability in relation to the magnitude of the estimate and is calculated by dividing the estimate by its standard error.

2.4.4 Standardization

For cluster analysis, co-expressed genes frequently show similar changes in expression but may differ in the overall expression rate. Therefore, gene expression vectors have to be standardized. Since most clustering algorithms performed in Euclidian space, co-expressed genes may thus be wrongly assigned to different clusters. Therefore, it is necessary to standardize the expression values of genes by a mean of zero and a standard deviation of one to ensure that vectors of genes with similar changes in expression are close. In standardization, the mean over all the experiments of a gene is subtracted from the expression level of the gene, and the difference is then divided by the standard deviation of the expression levels of the gene over all the experiments. Although normalization is only an intermediate step in the analysis, it has a considerable influence on the final results. The normalization equation as follows (Hoffmann *et al.*, 2002),

$$g_i = \frac{g_i - \overline{g_i}}{\sqrt{\sum (g_i - \overline{g_i})^2}} \tag{2.13}$$

For example, three hypothetical genes A, B and C are given, and their expression levels have been measured in normal tissue samples and diseased tissue samples. The results of these measurements are displayed in Figure 2.7 (a). Genes A and B are tightly co-regulated and differentially expressed across tissue types but they are expressed at different level. Gene C is not differentially expressed across tissue types but happens to have average expression levels similar to that of gene A. It is expected to find clusters that place genes A and B together but would not cluster them with gene C which is constant across all tissue samples. If clustering using the raw expression profiles, genes A and B will be separated. Figure 2.7 (b) shows the expression profiles for the same three genes after normalization across samples. In this transformed data, the expression values for genes A and B are closely aligned. In contrast, the values for gene C fluctuate randomly. This transformation results in that genes A and B being clustered in one group.



(a) Untransformed expression (b) Normalization

Figure 2. 7 Untransformed expression vs Normalization

2.5 Datasets

(1) *Yeast 384:* The full *Yeast cell cycle* dataset shows expression values of ap-

proximately 6000 genes over two cell cycles (17 time points). Tavazoie *et al*. (1999) selected 384 genes expression profiles, which peak at different time points corresponding to the five phases (G1, S, G2/M, M/G1 and S/G2) of cell cycle. The cell division contains four main phases G1, S, G2, and M shown in Figure 2.8. The division process begins at G1 phase where the cell is prepared for duplication. DNA is replicated in the phase S. In G2 phase, the cell is prepared for cell division. The final phase M is for mitosis, in which a cell is divided into two daughter cells. G2/M, M/G1 and S/G2 are the transition phase.

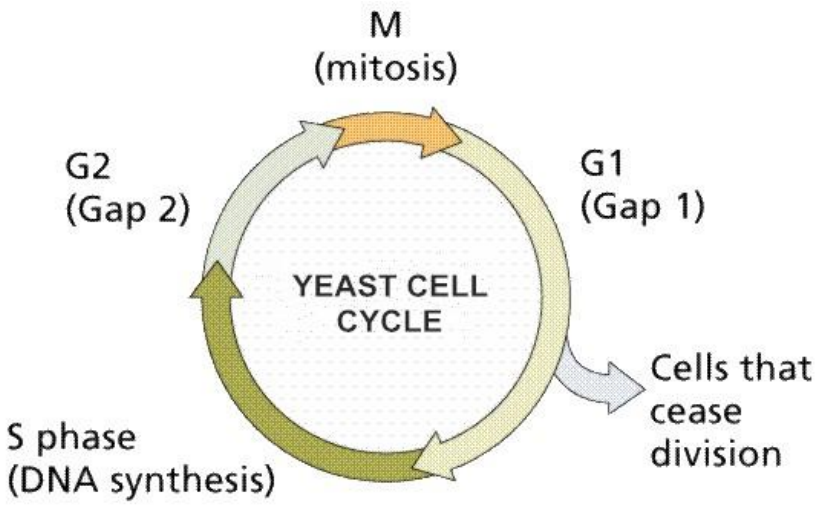


Figure 2. 8 *Yeast cell cycle* process

(2) *Yeast 2945:* Tavazoie *et al.,* (1999) selects 2945 genes by excluding values at time points 90 and 100 minutes. These data sets have already been normalized, where the average expression values are zero and the standard deviation is one.

(3) *Yeast 237:* Tavazoie *et al.,* (1999) extract 237 genes from the full *Yeast cell cycle* data which correspond to four functional classes: DNA synthesis and replication, organization of centrosome, nitrogen and sulphur metabolism, and ribosomal proteins. The resulting 237x17 data matrix is standardized.

(4) *Rat CNS:* This dataset consists of 112 genes in 9 time points. Wen *et al.* (1998) suggested that four major gene families are in this data: Neuro-Glial markers family (NGMs), Neuro-transmitter receptors family (NTRs), Peptide signaling family

(PepS) and Diverse (Div).

(5) *Serum*: This dataset contains 517 genes with 12 expression values (Iyer *et al.*, 1999). The expression of these genes varies in response to *Serum* concentration in human fibroblasts. Although there is no external criterion for this data set, Iyer *et al* (1999) suggests five various biological groups in this data.

The dataset's name, source and size are listed in Table 2.1. Optimal number of clusters in datasets are also specified.

Table 2. 1 Parameters of Gene expression data

Data name	Source	Size	Optimal number clusters
Yeast 384	*Cho et al.(1999) http://faculty.washington.edu/kayee/data.html*	384 x 17	5
Yeast 2945	*Tavazoie et al.(1999) https://tavazoielab.c2b2.columbia.edu/lab/*	2945 x 17	16
Yeast 237	*Tavazoie et al.(1999) http://faculty.washington.edu/kayee/data.html*	237 x 17	4
Rat CNS	*Wen et al. (1998) http://faculty.washington.edu/kayee/data.html*	112 x 9	4
Serum	*Iyer et al. (1999) Http://www.sciencemag.org/feature/data*	517 x 12	5

Chapter 3 Fuzzy c-means and kernel fuzzy c-means for gene expression analysis

3.1 Introduction

This chapter introduces the fuzzy c-means algorithm (FCM), by which a data item may belong to more than one cluster with different degrees of membership. Then a discussion of the parameter selection is presented. FCM is not robust to the noise and only effective finding spherical clusters, both of them limit its application for gene expression data analysis. Kernel fuzzy c-means (KFCM) is then presented which maps data onto a high dimensional feature space in order to increase the representation capability of linear machines. Experiments on artificial data and real gene expression showed that KFCM are more efficient and reliable.

3.2 Fuzzy theory

The mathematical models reviewed in Chapter 2 are crisp, deterministic, and precise in character, which means dichotomous, *yes-or-no* rather than *more-or-less*. However, the problems in the real world are not always yes-or-no type or true-or-false type. Real situations are often uncertain or vague. Due to lack of information the future state of the model might not be known completely. This type of uncertainty can be solved by fuzzy set theory (Bezdek, 1981), which describes

mathematically the imprecision or vagueness. Imprecisely defined classes play an important role, despite of this imprecision, humans still carry out sensible decisions. DNA microarray data contains uncertainty and imprecise information (Dembele and Kastner, 2003). Hard clustering methods such as *k*-means and SOM are poorly suited to the analysis of microarray data because the clusters of genes frequently overlap. Fuzzy theory has many advantages in dealing with data containing uncertainty. Fuzzy clustering approaches fits well with the fuzzy sets theory which takes this uncertainty into consideration to analyze DNA microarrays. In the fuzzy clustering, a cluster is viewed as a fuzzy set in the dataset. Thus, each feature vector in the dataset will have membership values with all clusters by indicating a degree of belonging to the cluster (Bezdek, 1981). The goal of a fuzzy clustering method is to define each cluster by finding its membership function.

3.3 Fuzzy c-means

Different cluster algorithms have been applied to the analysis of gene expression data: k-means, SOM and hierarchical clustering. All these methods have been restricted to a *one-to-one* mapping: one gene belongs to exactly one cluster (Dembele and Kastner, 2003). This principle seems reasonable in many fields of cluster analysis, it however might be limited for the study of microarray data. In biology, genes can participate in different genetic networks and are frequently coordinated by a variety of regulatory mechanisms (Andreas and Francis, 2005). For the analysis of microarray data, it is expected that single genes can belong to more than one cluster.

The most widely applied fuzzy clustering method is the fuzzy c-means (FCM) algorithm. Dembele and Kastner (2003) used FCM to analysis of microarray data and proposed a method for estimation of the fuzzy parameter *m*. In addition, FCM is applied to tumor classification and marker gene prediction by feature selection (Wang *et al*., 2003). Fu and Medico (2007) devised a cluster analysis software (GEDAS) based on FCM and the SOM algorithm, experiments and results show

that the proposed algorithm helps to discover co-expressed gene clusters. Pal *et al*. (2007) discovered biomarker from gene expression data for predicting cancer subgroups using neutral network and relational fuzzy clustering. Benjamin *et al*. (2010) proposed a fuzzy clustering approach to improve breast cancer prognostication. Maji and Paul (2013) proposed a robust rough-fuzzy c-means by integrating the merits of rough sets and fuzzy sets. The concept rough sets deals with uncertainty, vagueness, and incompleteness in cluster definition, the integration of probabilistic and possibilistic memberships of fuzzy sets enables efficient handling of overlapping partitions in noisy environment.

The FCM algorithm is an extension of the traditional hard *k*-means clustering algorithm by allowing one object belongs to more than one cluster (Bezdek, 1981). The FCM assigns a membership degree to each object. The centres of the clusters are computed based on the degree of memberships of objects. The algorithm is an iterative optimization that minimizes the cost function defined as follows:

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2}(x_j, v_i) \tag{3.1}$$

where C and N denote the number of clusters and objects respectively, v_i is the ith cluster centre, u_{ij} represent the membership of object x_j in the ith cluster, which satisfy: $0 \le u_{ij} \le 1$ and $\sum_{i=1}^{c} u_{ij} = 1$. $d_{ij}^{2}(x_j, v_i)$ is the Euclidean distance between vector x_j and prototype v_i. The original formulation of FCM uses point prototypes and inner-product induced norm metric for d_{ij}^{2} given by:

$$d_{ij}^{2}(x_j, v_i) = \| x_j - v_i \|_{A}^{2} = (x_j - v_i)^{T} A (x_j - v_i) \tag{3.2}$$

The parameter m controls the fuzziness of the resulting partition. i.e. the degree to which the membership of a gene is distributed among the clusters. For $m \to 0$, the FCM turns into hard clustering of the data. The prototypes v_i are then simply the means of the clusters j. For $m \to \infty$, the partition approaches maximal fuzzi-

ness. Besides the parameter *m*, users must choose values of the minimal change in the objective function for termination and the maximal number of iterations. The above constrained optimization problem can be solved by using Lagrange multipliers (Bezdek, 1981):

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2}(x_j, v_i) + \lambda\left(\sum_{i=1}^{C} u_{ij} - 1\right) \tag{3.3}$$

The closed form formulas for updates are derived by taking the partial derivatives with respect to both and setting them to zero. When the iteration converges, a fuzzy partition matrix and the pattern prototypes are obtained (Bezdek, 1981). The partition matrix and the cluster centre of KFCM are estimated by (3.4) and (3.5).

$$u_{ij} = \frac{d_{ij}^{2}(x_j, v_i)^{-1}}{\sum_{i=1}^{c} d_{ij}^{2}(x_j, v_i)^{-1}} \tag{3.4}$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} x_j}{\sum_{j=1}^{N} u_{ij}^{m}} \tag{3.5}$$

Based on equation (3.4) and (3.5), the following algorithm is used (Dembele and Kastner, 2003),

Fuzzy c-means (FCM). Given *N* data $X = \{x_j\}_j^N$ **and the desired number of cluster *C*, output a membership matrix** $U = \{u_{ij}\}$

1: Initialize number of clusters *C*, and fuzzy exponent parameter *m*
2: Initialize iteration counter $k = 0$**;**
3: Initialize the fuzzy partition matrix U^0**;**
4: Compute the initial prototypes v_i
5: Repeat**:**
6: (a) Update all memberships U^0 **with Equation 3.4;**
7: (b) Update all prototypes v_i **with Equation 3.5;**
8: Until **(prototype parameters stabilize)**

Figure 3. 1 Fuzzy c-means algorithm

In FCM, data are bound to each cluster by means of a membership function, which represents the fuzzy behavior of this algorithm. It builds an appropriate matrix *U* with matrix elements ranging between 0 and 1, which represents the degree of membership between data and centres of clusters. Figure 3.2 is a mono-dimensional example, a one dimension dataset is distributed on *x* axis.

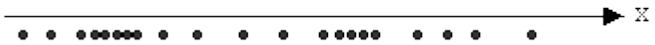


Figure 3. 2 Mono-dimensional data distribution

For *k*-means algorithm, it associates each datum to a specific centroid. Two clusters (referring as A and B) can be identified in proximity of the two data concentrations. The membership degree *m* can be seen in Figure 3.3.

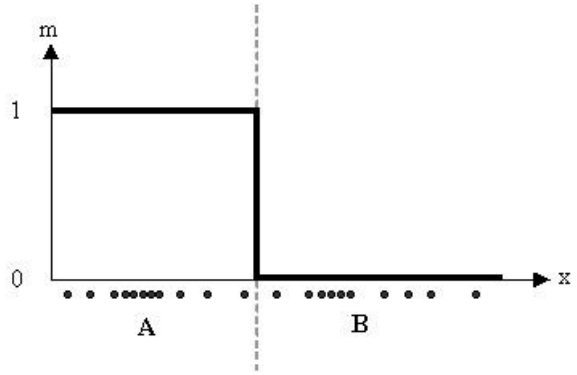


Figure 3. 3 *k*-means membership degree

For FCM, the datum does not belong to one cluster exclusively. Instead, the datum belongs to all clusters with a membership coefficient (Figure 3.4).

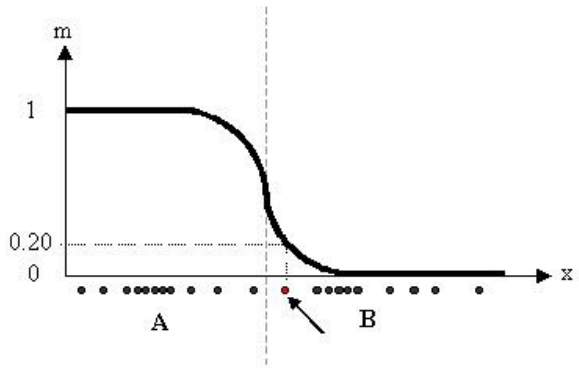


Figure 3. 4 Fuzzy c-means membership degree

Specifically, the datum shown as a red marked spot belongs more to the B cluster rather than the A cluster. The value 0.2 indicates the degree of membership to A

for this datum.

$$U_{Mc} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{Mc} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

(a) k-means membership matrix (b) FCM membership matrix

Figure 3. 5 Membership matrix

The FCM introduces a matrix *U* to represent the membership of each sample belonging to all cluster centres. The number of rows and columns are determined by the number of data *N* and the number of clusters *C* respectively. Figure 3.5 shows the comparison of membership function of k-means and FCM. Figure 3.5 (a) is k-means membership for each datum, the coefficients are always unitary by indicating that each datum can belong only to one cluster. Figure 3.5 (b) is the membership function of FCM, by which each datum can belong to every cluster.

3.3.1 Initialization

For FCM, different initializations can lead to various results because it only converges to local minima. In order to avoid FCM trapping in local minima, the conventional method is to run FCM with different initializations and choose the one with the smallest value of the objective function. However, this approach is time consuming and instable. Recently, FCM has been integrated with optimization algorithms, such as, the Genetic Algorithm, Particle Swarm Optimization, and Ant Colony Optimization (Mehdizadeh *et al*., 2008; Halder *et al.*, 2011; Ghosh *et al.*, 2011; Lianjiang *et al.*, 2010). Alternatively, a mountain clustering method has been used with FCM, which can find the optimal cluster centres (Yang *et al*. 2010). The mountain clustering estimates the cluster centres based on the density function. It consists of the following steps:

(1) Firstly, a grid will be formed based on the data space. All the grid points of the data space, shall initially be considered as possible cluster centres.

(2) Construction of the mountain function which denotes the data's density. The mountain function of grid point N_i is given by

$$M_1(N_i) = \sum_{j=1}^{N} e^{-\alpha d(x_j, N_i)} \tag{3.6}$$

where, $d(x_j, N_i) = \left\| x_j - N_i \right\|^2$, x_j $(j = 1, ...n)$ is the j th point, α is a positive constant according to the dataset. The formula indicates that every point x_j contributes to the value of the mountain function, and the contribution is inversing to the distance between the point x_j and the grid point N_i. The mountain function value, $M(N_i)$, tends to higher values while the number of samples close by N_i increases, and the mountain function value tend to decrease when the number of samples close by N_i decreases. So the mountain function can be regarded as an index of data's density. The parameter α is important in mountain cluster method. It not only identifies the high density value but also the smoothness of the mountain function.

(3) The third step involves the identification of cluster centres by subsequent destruction of the mountain peaks (Yang *et al*. 2010). In this step, the grid point which has the largest mountain function value is selected as the first cluster centre. Let N_1^* is the first cluster centre, it is found with

$$M_1(N_1^*) = \max\{M_1(N_i)\} \tag{3.7}$$

To find other cluster centres, the identified cluster centre will be eliminated. A value inversely proportional to the distance of the grid point from the found centres is subtracted from the previous mountain function. This process is carried out using the equation:

$$M_k(N_i) = M_{k-1}(N_i) - M_{k-1}(N_{k-1}^*) \cdot e^{\gamma d(N_{k-1}^*, N_i)} \tag{3.8}$$

where

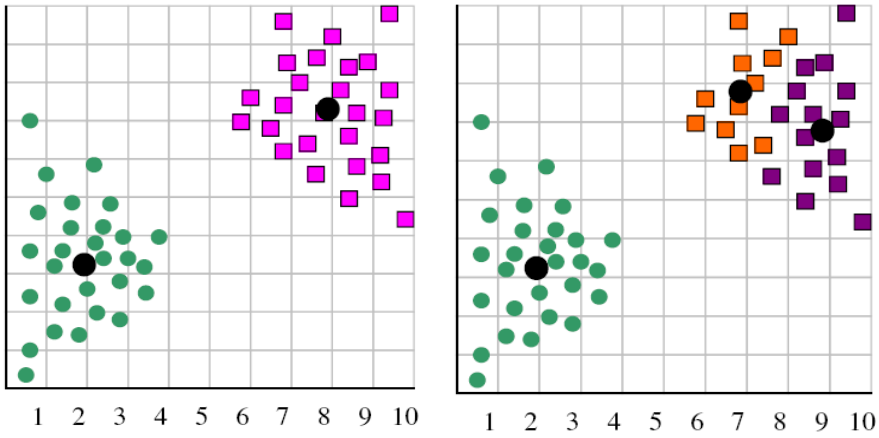
$$M_{k-1}(N_{k-1}^{*}) = \max\{M_{k-1}(N_i)\} \tag{3.9}$$

$M_{k-1}(N_{k-1}^{*}) \bullet e^{-\gamma d(N_{k-1}^{*}, N_i)}$ and N_i are in direct proportion with $M_{k-1}(N_1^{*})$, but inversely proportional to the identified cluster centre N_1^{*}. The new mountain function value $M_k(N_1^{*}) = 0$, and then the grid point that has the largest value of new mountain function $M_k(N_i)$ is selected as the second cluster centre. The identification process continues until enough cluster centres are identified.

Although mountain clustering can overcome the problem of initialization, it requires a priori specification of the parameters: the grid resolution and the mountain peak. The clustering performance of the mountain method strongly depends on the grid resolution, with finer grids giving better performance. As the grid resolution increases, however, its computation grows exponentially with the dimension. Most mountain clustering methods use constant values for these parameters. However, different datasets have different data distributions, and these values need to be adjusted accordingly (Loquin and Strauss, 2008).

3.3.2 Number of clusters

When clustering data without any priori information of the data structure, one usually has to make assumptions about the number of clusters. Figure 3.6 is an example, the given data can be grouped into two clusters (Figure 3.6 (a)), whereas it also reasonable to cluster the data into three groups (Figure 3.6 (b)). The optimal number of cluster is a crucial for the clustering result, various number of clusters will results in different explanation.



(a) Identified two clusters (b) Identified three clusters

Figure 4. 6 Optimal number of clusters

Due to the diversity and uncertainty in dataset, the clustering algorithm frequently searches for a range number of clusters, regardless of whether they are really present in the data or not. The most widely used method is locating the "knee" of an error curve (Foss and Za ïane, 2002). It plots the evaluation metrics versus the number of clusters. The aim is to find the point where increasing the number of clusters does not add much information anymore. The evaluation metrics can be computed based on the sum of all pairwise distances between data in each cluster, the sum of distance between clusters etc.

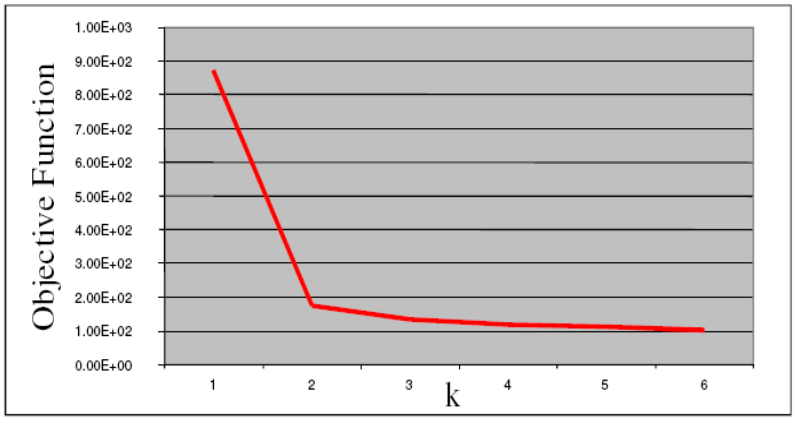


Figure 3. 7 The intra distance vs number of cluster

Figure 3.7 shows an evaluation graph of a hierarchical clustering algorithm on the

dataset of Figure 3.6. The y-axis values are the sum of intra distances of all clusters. The curve has three distinctive areas: a sharply-sloping region to the left, a curved transition area in the middle and a flat region to the right. Starting from the left, the sum of intra distance distances decrease rapidly. This rapid decrease in distance indicates that similar objects are being merged together, and that the quality of the clusters is becoming good because objects are internally homogeneous. Another interesting area of the graph is the flat region, where the clustering process begins at the initial fine grain clustering, there are many very similar clusters to be merged and the trend continues to the right in a rather straight line for some time. In this region, many clusters are similar to each other and should be merged. A reasonable number of clusters is therefore in the curved area, or the "knee" of the graph. This knee region is between the quickly decreasing region on the left side, and the low distance merges that form a nearly straight line on the right side of the graph. Clustering in this knee region contain a balance of clusters that are both highly homogeneous, and also dissimilar to each other. Determining the number of clusters where this knee region exists will therefore give a reasonable number of clusters to return. However, locating the exact knee point is problematic if the knee is a smooth curve. In such an instance, the knee could be anywhere on this smooth curve, and thus the number of clusters to be returned seems imprecise. Such an evaluation graph would be produced by a dataset with clusters that are overlapping or not very well separated. In such instances, there is no single 'correct' answer and all of the values along the knee region are likely to be reasonable estimates of the number of clusters. Thus, an ambiguous knee indicates that there probably is no single optimum answer, but rather a range of acceptable answers (Foss and Za ̈ane, 2002).

3.3.3 Fuzziness exponent

Fuzziness exponent *m* is a crucial parameter since it determines the influence of noise on the cluster analysis (Bezdek, 1981). For *m=1*, FCM becomes hard clustering, and the FCM algorithm is then equivalent to the k-means clustering. The

membership values are either one or zero. All genes of a cluster are treated equally for the calculation of the cluster centre. Increasing the parameter *m* reduces the influence of genes with low membership values. Gene expression vectors with large noise content generally have a low membership value, since the corresponding genes are not well represented by a single cluster, but rather are partially assigned to several clusters. When *m* goes to infinity, all memberships $u_{ij} = 1/j$, the FCM will become the fuzziest and all clusters will be melt.

In the FCM literatures (Ghosh et al., 2011; Graves and Pedrycz, 2010), *m*=2 is frequently used, which however is not suitable for gene expression data (Dembele and Kastner, 2003). For example, FCM is used to partition the *Yeast* microarray data by setting *m=2*, it can be observed that all the membership values were similar in Figure 4.8 (a), this indicates that FCM failed to extract any clustering structure. On the other hand, if setting *m=1.17* (Dembele and Kastner, 2003), the membership values have a non-uniform distribution(Figure 4.8 (b)), by which distinct clusters can be found.

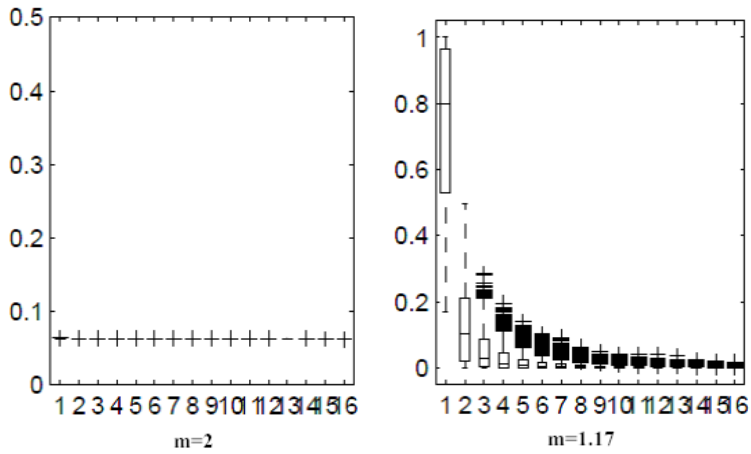


Figure 3. 8 Influence of the fuzziness parameter *m*
(Horizontal axis represents number of clusters, vertical axis represents membership values. Boxplot represents of sorted membership values of FCM. For fixed values of *m*, the *C* membership values of each gene are sorted in decreasing order. For a point in each plot, horizontal segments are 99 centile, third quartile, median, first quartile and first centile values respectively; isolated segments represent noises.)

Estimation the value for fuzziness parameter *m* is a significant issue in applying the FCM method to microarray data analysis. The optimal values for *m* vary from one dataset to another. Although some researchers have already given some

methods for choosing the values of *m*, these methods usually are time-consuming (Yang *et al*., 2007). Empirically, FCM algorithm can obtain the best clustering results by minimizing the objective function,

$$\begin{aligned}\frac{\partial J}{\partial m} &= \sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^{m}\ln(\mu_{ij})d_{ij}^{2}(x_j;v_i) \\ &= \sum_{i=1}^{C}\sum_{j=1}^{N}[\mu_{ij}\lg(\mu_{ij})][\mu_{ij}^{m-1}d_{ij}^{2}(x_j;v_i)] < 0\end{aligned} \tag{3.10}$$

According to the formula (3.10), the objective function monotonically decreases with the increase of *m*. It is reasonable to select the value of *m* when the clustering result attains its minimum. The objective function has a minimum point to the partial derivative of the objective function with respect to the parameter *m* (Dembele and Kastner, 2003).

$$m^{*} = \{m \left| (\frac{\partial J}{\partial m}) = 0\right.\} \tag{3.11}$$

As for the fuzzy clustering, the inflection point of the objective function just corresponds to the minimal value of its derivative. The optimal weighted index m^{*} can be selected by using the following formula:

$$m^{*} = \arg\{\min\{\frac{\partial J}{\partial m}\}\} \tag{3.12}$$

According to Kim (2006) and Dembele and Kastner (2003), the fuzzy exponent *m* for the datasets used in this thesis is empirically chosen following equation in table 3.1,

Table 3.1 Fuzzy exponent for datasets

Dataset	*m*
Yeast 384	1.34
Yeast 237	1.34
Yeast 2945	1.68
Rat CNS	1.21
Serum	1.25

3.3.4 Proximity measurement

Clustering methods usually require the definition of distance or similarity between data, to identify genes or samples that have similar expression profiles. Similarity measure is crucial for the clustering results, which relies on the unique characteristics of the specific data structure (Yang *et al.*, 2003). The commonly used distance metrics for gene expression data analysis are Euclidean distance and Pearson Correlation coefficients.

Euclidean distance computes the difference based on the absolute expression value (Yang *et al.*, 2003), which is given by

$$E(x,y)=\sqrt{\sum_{i=1}^{n}\frac{(x_i-y_i)^2}{n}} \tag{3.14}$$

In Equation (3.14), the expression profiles x_i and y_i are subtracted directly from each other. Therefore, it needs to ensure that the expression data are properly normalized when using the Euclidean distance (Yang *et al.*, 2003), for example by converting the measured gene expression levels to log-ratios., which may identify similar or identical regulation. For gene expression data, the overall shapes of gene expression patterns (or profiles) are of greater interest than the individual magnitudes of each feature. Euclidean distance does not score well for shifting or scaled patterns (Wang *et al*., 2002).

Pearson correlation is based on the Pearson correlation coefficient (PCC), which describes the similarity of objects as

$$PCC(x,y)=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{3.15}$$

where $\bar{x}$, $\bar{y}$ denote the mean expression value. *PCC* has a value from −1 to 1, where *PCC*=1 when *x* and *y* are identical, *PCC*=0 when they are unrelated, and

PCC=−1 when they are anti-correlated (Yang *et al.*, 2003). The Pearson's correlation distance is then defined as,

$$P(x, y) = 1 - PCC(x, y) \quad (3.16)$$

The value of $P(x, y)$ lies in [0, 2], $P(x, y) = 1$ implies that *x* and *y* have no correlation, and $P(x, y) = 0$ and $P(x, y) = 2$ imply that *x* and *y* are totally different and identical respectively. The Pearson correlation measures the similarity between the shapes of two expression patterns, which is invariant under any scalar transformation of the data (Yang *et al.*, 2003).

However, both the Pearson correlation and the Euclidean distance are sensitive to noise and outliers (Yang *et al.*, 2003). A single noise could transform the Euclidean distance to an unbounded value, while transforming the Pearson correlation to any value between -1 and 1. Both measures are easily distorted when the expression levels are not uniformly distributed across the expression pattern (Yang *et al.*, 2003). For example, two expression patterns with one high measured value at the same condition will obtain a high correlation coefficient score, regardless of the expression values of the other cellular conditions. Similarly, a large difference in a single expression level at the same cellular will lead to a high Euclidean distance, regardless of the other expression levels.

3.4 Kernel based Clustering

The use of kernels has received considerable attention in pattern recognition, because kernels make it possible to map data onto a high dimensional feature space and increase the representation capability of linear machines. Girolami (2002) generalized the approach for a wider variety of clusters when he proposed kernel-based clustering. Chiang and Hao (2003) proposed a multiple spheres support vector clustering algorithm based on the adaptive cell growing model which maps data points to a high dimensional feature space using the desired kernel function. Camastra and Verri (2005) presented a kernel based clustering algorithm inspired by the k-means algorithm that iteratively refines results using a one-class support

vector machine. Tzortzis and Likas (2009) proposed a deterministic and incremental algorithm to overcome the cluster initialization problem: their algorithm maps data points from the input space to a higher dimensional feature space through the use of a kernel function and optimizes the clustering error. Filippone *et al*. (2008) contributed a survey of kernel and spectral clustering methods, in which kernel clustering methods are taken as the kernel versions of classical clustering algorithms such as k-means and SOM.

3.4.1 Kernel

Kernels are a generalization of a known inner product. Instead of working directly on the given data, the kernel methods discover more intricate information by mapping it to feature space *H* (Shawe-Taylor and Crisianini, 2004). The mapping *F* between the original space and feature space should benefit the comparisons for similarity measures between the original data points. For example, Figure 3.9 shows a feature mapping, $\phi: (x_1, x_2) \to (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, two types of data are not linear separable in Figure 3.9 (a) (e.g. $\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1$). By a feature mapping, the data can be represented in another space as shown in Figure 3.9 (b), where the data become linearly separable (e.g. $\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1 \Rightarrow \frac{1}{a^2} z_1 + 0 \cdot z_2 + \frac{1}{a^2} z_3 = 1$).

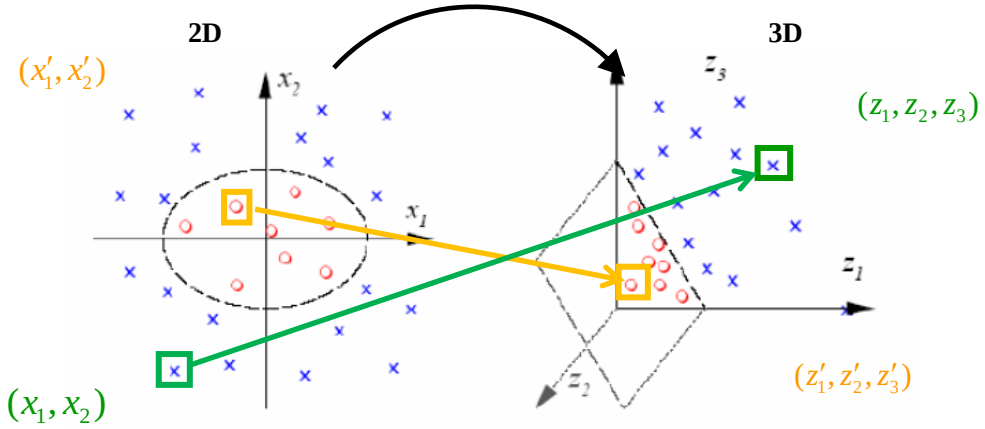


Figure 3. 9 Kernel mapping

The inner product between any pair of the data points in the feature space *H* is

calculated by kernel function in terms of their inner product in the original space.

$$\begin{aligned} <\phi(x_1,x_2),\phi(x_1',x_2')> &= <(z_1,z_2,z_3),(z_1',z_2',z_3')> \\ &= <(x_1^2,\sqrt{2}x_1x_2,x_2^2),(x_1'^2,\sqrt{2}x_1'x_2',x_2'^2)> \\ &= x_1^2x_1'^2+2x_1x_2x_1'x_2'+x_2^2x_2'^2=(x_1x_1'+x_2x_2')^2 \\ &= (<x,x'>)^2=k(x,x') \end{aligned} \tag{3.17}$$

Thus, $k(x,x')=<x,x'>^2$ is a valid kernel function. The correspondence of kernels to feature spaces is one to many. For example, the same kernel computes the inner product for the four dimensional map

$$\phi: x=(x_1,x_2)\Rightarrow\phi(x)=(x_1^2,x_2^2,x_1x_2,x_2x_1)\in F=R^4$$

The example is a special case for an entire family of kernels, the polynomial kernel: $k(x,z)=<x,z>^d=(\sum_{i=1}^{n}x_iz_i)^d$. Each member of this family maps to the feature space spanned by all the polynomials of order *d* (Filippone *et al.*, 2008*)*.

The inner products of the mapped data points are able to define the similarity between the original data points. Hence, in order to compare data points in terms of their similarities, it is unnecessary to know the explicit mapping from the original dataset to the feature space. It is possible to compute distances in feature space without knowing explicitly. Thus, the similarity between (x_1,x_2) and (x_1',x_2') is defined as the one between $\phi(x_1,x_2)$ and $\phi(x_1',x_2')$, and finally represented in terms of the inner product in *H*,

$$\begin{aligned} \|\phi(x)-\phi(x')\|^2 &= (\phi(x)-\phi(x'))^T(\phi(x)-\phi(x')) \\ &= \phi(x)^T\phi(x)-2\phi(x)^T\phi(x')+\phi(x')^T\phi(x') \\ &= <\phi(x),\phi(x)>-2<\phi(x),\phi(x')>+<\phi(x'),\phi(x')> \end{aligned} \tag{3.18}$$

Another application is to compute the angle between vectors in the feature space,

$$\begin{aligned} &<\phi(x),\phi(x')>=\|\phi(x)\|\cdot\|\phi(x')\|\cos\theta \\ &\Rightarrow\cos\theta=\frac{<\phi(x),\phi(x')>}{\|\phi(x)\|\cdot\|\phi(x')\|}=\frac{<\phi(x),\phi(x')>}{\sqrt{<\phi(x),\phi(x)>}\sqrt{<\phi(x'),\phi(x')>}} \end{aligned} \tag{3.19}$$

Given a set of vector, $\{x_1, x_2, ...x_N\}$ all kernel related methods construct the kernel matrix, which gives all the information about the relations between the vectors (John and Nello, 2004). If the kernel is valid, *K* is symmetric definite-positive,

$$K = \begin{bmatrix} <\phi(x_1),\phi(x_1)> & \cdots & <\phi(x_1),\phi(x_N)> \\ \vdots & \ddots & \vdots \\ <\phi(x_N),\phi(x_1)> & \cdots & <\phi(x_N),\phi(x_N)> \end{bmatrix} = \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_N) \\ \vdots & \ddots & \vdots \\ k(x_N,x_1) & \cdots & k(x_N,x_N) \end{bmatrix} \tag{3.20}$$

where $k(x_i, x_j) =< x_i, x_j >^2$ is a kernel function.

Different kernels will induce different metric measures for the original space with new clustering algorithms. Some robust kernels are listed in Table 3.2.

Table 3. 2 Kernel functions

Name of Kernel	Kernel function
Log	$\log(1+\beta\|x-y\|^2)$
Sigmoid	$\tanh(\frac{\beta\|x-y\|^2}{2})$
Cauchy	$\frac{1}{1+\beta\|x-y\|^2}$
Gaussian	$\exp(-\frac{\|x-y\|^2}{2\sigma^2})$

3.4.2 Kernel based FCM clustering

Since FCM uses the squared-norm as proximity measure, it is effective finding spherical clusters. In order to identify more general shape of clusters, many improvements have been made. Zhang and Chen (2003) proposed the kernel-based fuzzy c-means (KFCM) algorithm which allows for incomplete data. Shen *et al.* (2006) addressed the same problem using weighted KFCM for better feature selection. As mentioned by Graves and Pedrycz (2010), KFCM is divided into two

categories. In the first category, prototypes reside in the original space and are implicitly mapped to the kernel space through the use of a kernel function, whereas in the second category, prototypes are directly constructed in the kernel space, which allows more freedom for prototypes in the feature space (Zhang and Chen, 2003). KFCM adopts a new kernel-induced metric in the data space to replace the original Euclidean norm metric in FCM. By replacing the inner product with an appropriate kernel function, one can implicitly perform a nonlinear mapping to a high dimensional feature space without increasing the number of parameters. According to Wu and Yang (2002), Gaussian kernel is more robust than other kernels and it has been successfully applied into many learning systems, such as Support Vector Machines (SVM), kernel principal component (John and Nello, 2004). Therefore, Gaussian is adopted as the kernel in this work.

Consider the dataset X, and F is the transformed feature space with higher or even infinite dimension. KFCM is based on kernelization of the metric, which computes centroids in input space and the distances between patterns in kernels. The method minimizes the following objective function,

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N} \mu_{ij}^{m} \left\| \phi(x_j) - \phi(v_i) \right\|^2 \tag{3.21}$$

where $\left\| \phi(x_j) - \phi(v_i) \right\|^2 = k(x_j, x_j) + k(v_i, v_i) - 2k(x_j, v_i)$.

According to (3.21), the partition matrix and the cluster centres of KFCM are estimated by (3.22) and (3.23).

$$u_{ij} = \frac{(1/(1-k(x_j, v_i)))^{1/(m-1)}}{\sum_{i=1}^{C}(1/(1-k(x_j, v_i)))^{1/(m-1)}} \tag{3.22}$$

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} k(x_j, v_i)\, x_j}{\sum_{j=1}^{N} u_{ij}^{m} k(x_j, v_i)} \tag{3.23}$$

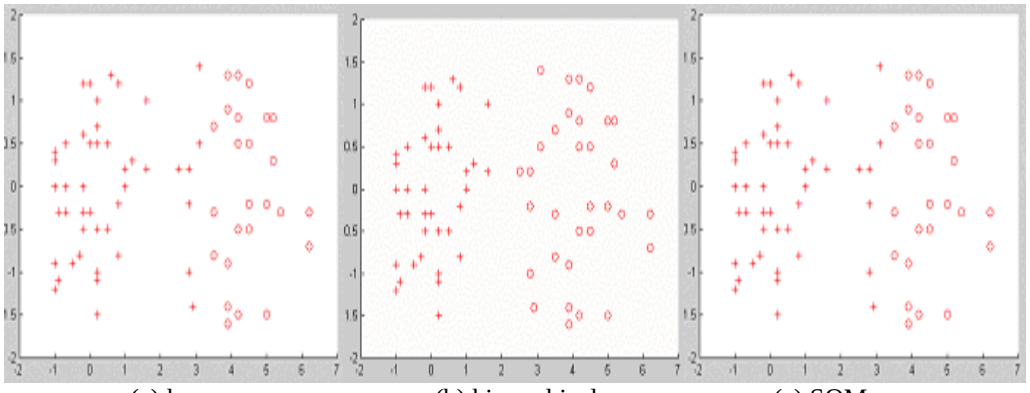
3.5 Evaluation of performance

In order to evaluate the performances of the clustering algorithms, both artificial data and gene expression data are used to assess the quality of the clusters. In KFCM, σ in the kernel function is set 150 empirically. Agglomerative hierarchical clustering with *average linkage* is selected for comparison.

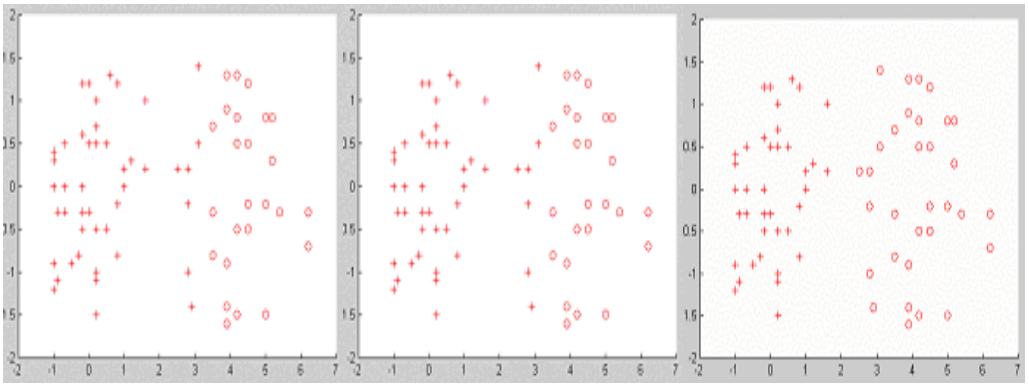
3.5.1 Artificial data

Experiments 1: Two-cluster dataset

In order to examine these clustering algorithms' performance on finding arbitrary cluster, a two-cluster dataset is produced with elliptical distribution. The cluster results of *k*-means, Hierarchical, SOM, EM, FCM and KFCM are shown in Figure 3.10(a)-(f) respectively, where two clusters from the clustering algorithms are with symbols "+" and "o". Figure 3.10 shows that Hierarchical and KFCM clustering method can identify the correct structure, while k-means and FCM fail to find the underlying patterns correctly because incorporating Euclidean distance is only effective finding spherical clusters thereby lacking the ability to capture no spherical clusters. EM assumes data is fitted in Gaussian mixture models, which is not suitable for this dataset. SOM incorporating Euclidean distance computes the similarity between the input vector and the map's node's weight vector, which is not good at finding arbitrary shapes of clusters.



(a) k-means (b) hierarchical (c) SOM

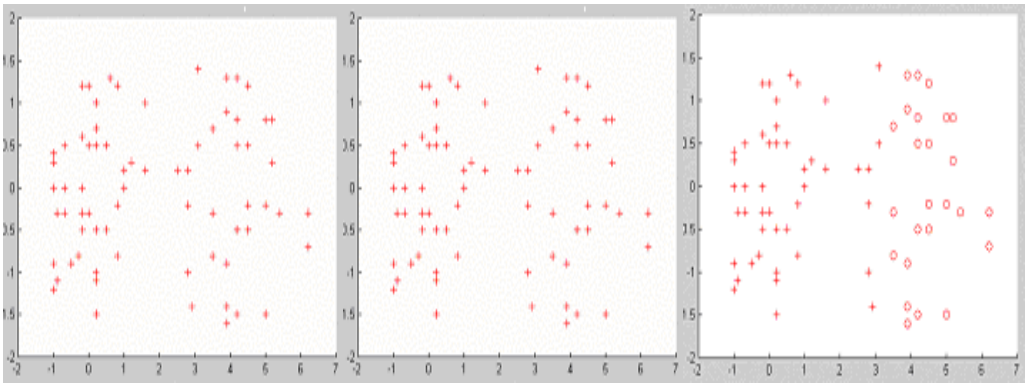


(d) EM (e) FCM (f) KFCM

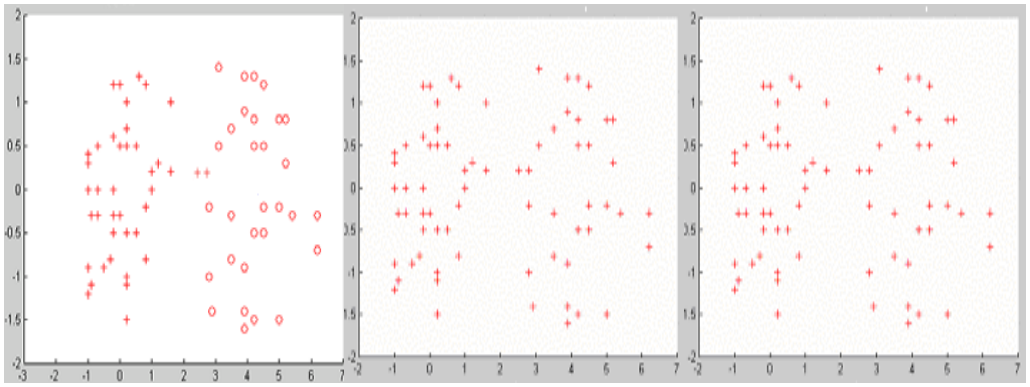
Figure 3.10 Clustering result for two-cluster data

Experiments 2: Two-cluster dataset with an outlier

In order to demonstrate the robustness, an outlier (20, 0) is added to the above dataset. The cluster results are shown in Figure 3.11(a)-(f). It can be seen that all of these algorithms are heavily affected by the outlier. Specifically, the partitional clustering (k-means, FCM and KFCM) and Hierarchical clustering algorithm have poor performance, which takes the two clusters as a whole one. The best performance is achieved by EM algorithm, and only 2 objects are grouped mistake. For EM, it assumes that data are generated by a mixture of Gaussian distributions with certain probability, therefore, it is less impacted by the outlier.



(a) k-means (b) hierarchical (c) SOM

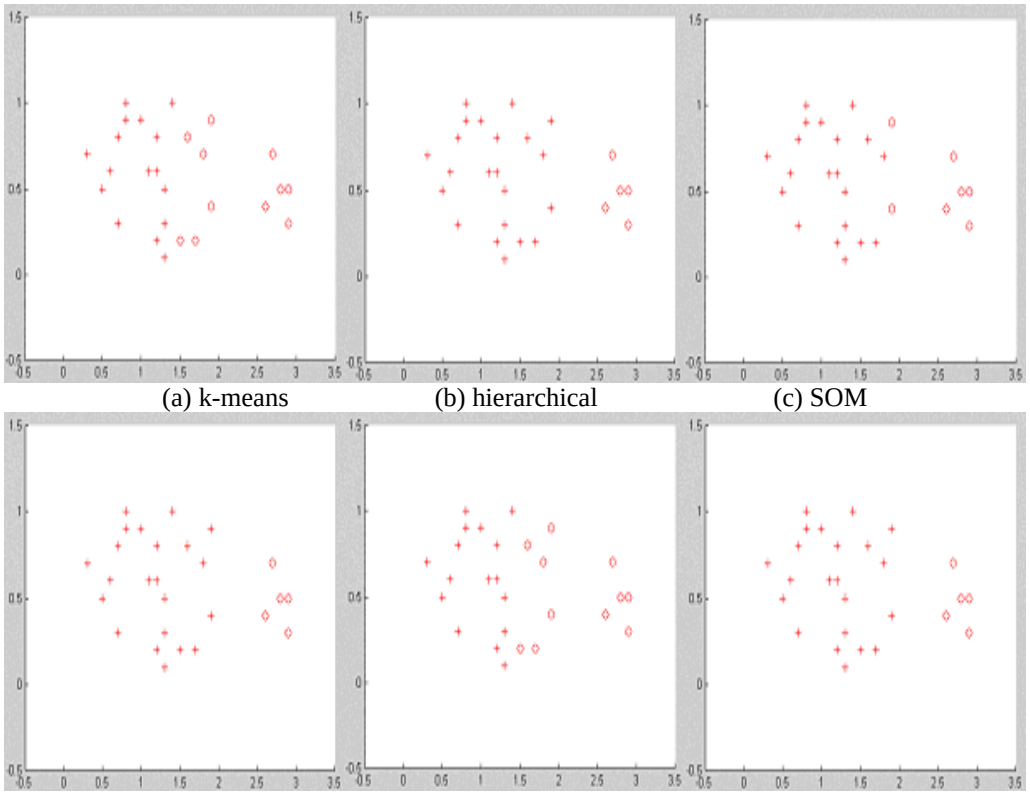


(d) EM (e) FCM (f) KFCM

Figure 3. 11 Clustering result for two-cluster data with noise

Experiment 3 Unbalance clusters

Two clusters are produced with different volume. Figures 3.12 (a)-(f) show the clusters produced by *k*-means, Hierarchical, SOM, EM, FCM and KFCM respectively. It can be seen that *k*-means, SOM and FCM cannot detect the correct clusters. The other methods can detect the patterns correctly.



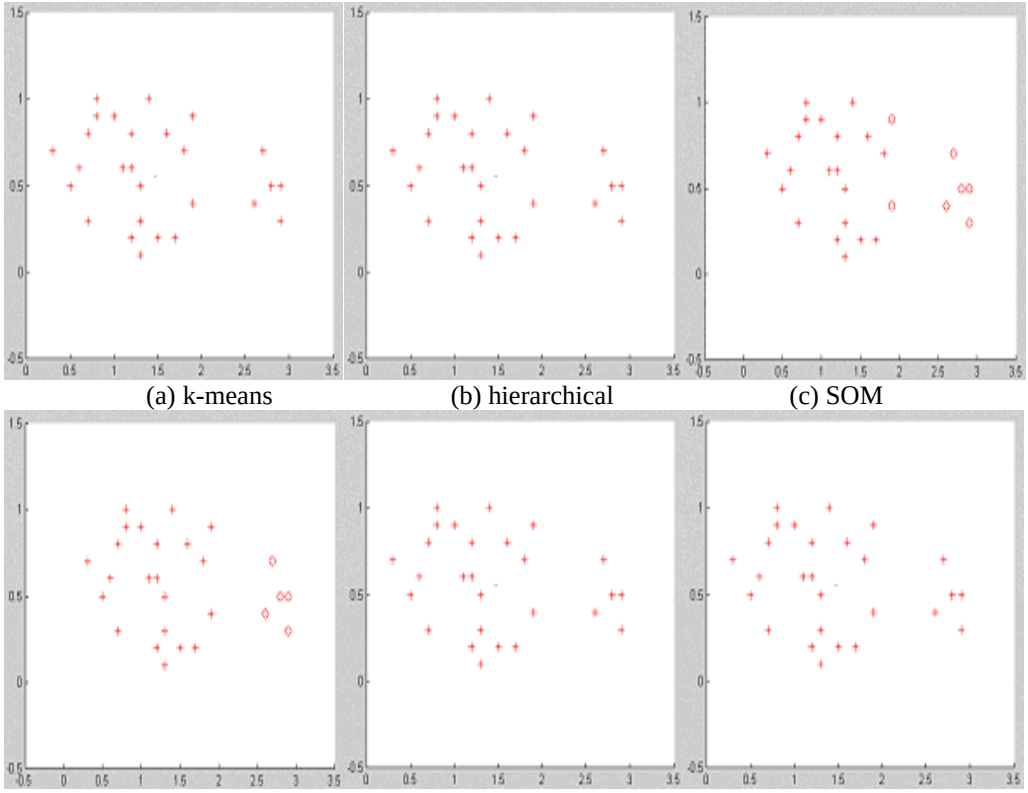
(a) k-means (b) hierarchical (c) SOM

(d) EM (e) FCM (f) KFCM

Figure 3. 12 Clustering result for unbalance data

Experiment 4: Unbalance cluster with an outlier

An outlier (20, 0) is added to the unbalance clusters. The partitional clustering and hierarchical clustering algorithms are heavily affected by the outlier and takes all data as a cluster and the noise a singleton. For EM and SOM, they are not sensitive to the outlier relatively, especially EM can identify the correct patterns.



(a) k-means (b) hierarchical (c) SOM

(d) EM (e) FCM (f) KFCM

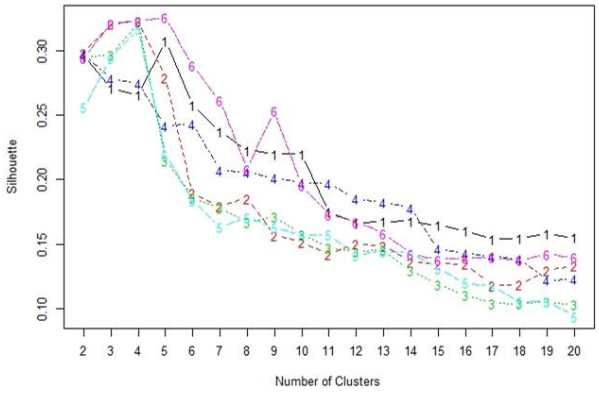
Figure 3. 13 Clustering result for unbalance data with noise

3.5.2 Gene expression data

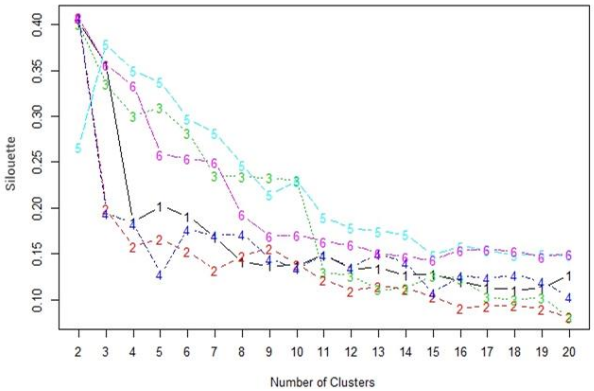
In order to examine the performance of clustering algorithms for real gene expression data, three dataset are selected: *Yeast 384*, *Yeast 237* and *Rat CNS*. For assure the optimal clusters, all methods run a range of number of clusters. The software R was used for experiments, in addition, "clValid", "cluster", R packages, are used for assess the quality of produced clusters in three validation measure, Silhouette, FOM and BHI. Moreover, the package, "mclust", is used to assess the clusters quality in term of ARI (Asyali and Alci, 2005). Before experiments, the

data was log2 transformed to make symmetry between negative and positive fold change and normalized to obtain a mean expression value of one for each gene. This ensures that genes which share the same expression pattern have similar gene expression vectors. For FCM, the fuzziness exponent is empirically set to 1.34 1.34 and 1.21 for *Yeast 384*, *Yeast 237* and *Rat CNS* respectively.

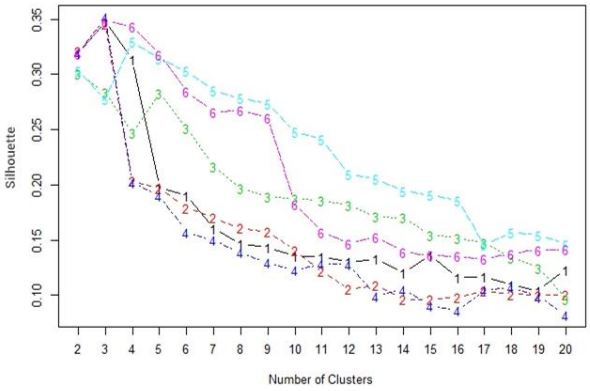
In term of Silhouette, Figure 3.14 (a) shows that KFCM achieves better performance than the other methods on *Yeast 384* in the proceeding number of clusters (2-10), this indicates that the clusters generated by KFCM have better intra homogeneity and inter separateness. Due to using Euclidean distance, FCM cannot identify arbitrary shapes of cluster and shows poor performance. However, FCM outperforms the other methods for a range number of clusters (3-15) for *Yeast 237*. For *Rat CNS*, FCM and KFCM have similar performances on the number of cluster (4-9). Detail comparison for optimal number of clusters can be found in Table 3.3, where FCM and KFCM achieve better performances than the other methods.



(a) Silhouette index for *Yeast 384*



(b)Silhouette index for *Yeast 237*



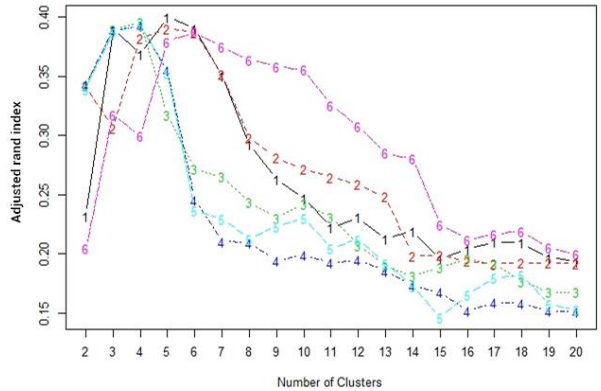
(c) Silhouette index for Rat CNS

Figure 3. 14 Silhouette index for three datasets
(Line 1-6 represent k-means, hierarchical clustering, SOM, EM, FCM and KFCM respectively)

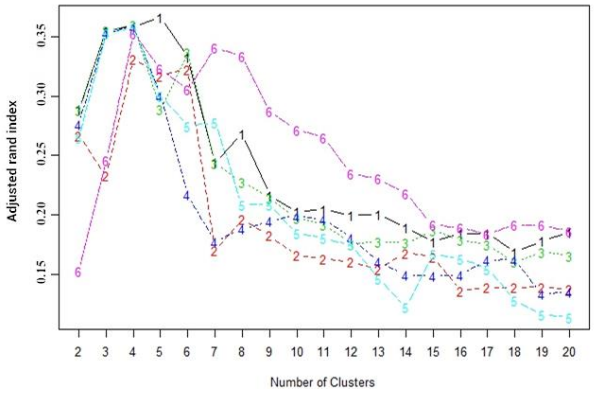
Table 3. 3 Sillouette index for optimal number of clusters

	k-means	Hierarchical	SOM	MODEL	FCM	KFCM
Yeast 384	**0.309**	**0.275**	**0.211**	**0.246**	**0.215**	**0.341**
Yeast 237	**0.178**	**0.161**	**0.306**	**0.177**	**0.355**	**0.343**
Rat CNS	**0.314**	**0.202**	**0.252**	**0.201**	**0.336**	**0.344**

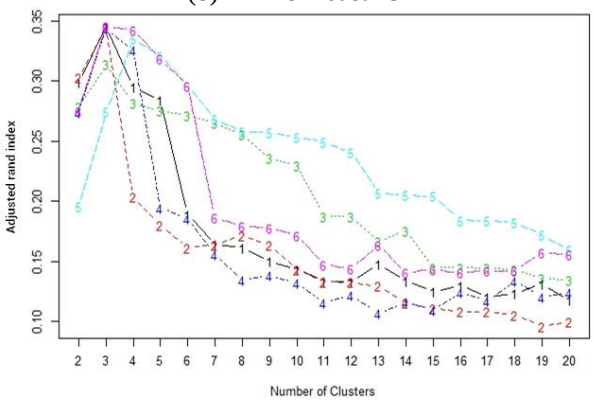
In term of ARI, KFCM outperforms the other methods on *Yeast 384* and *Yeast 237*. For *Rat CNS*, KFCM perform better on the range number of clusters (2-7), with the increasing number of clusters, the cluster accuracy produced by KFCM decreased. Meanwhile, FCM performs better on most number of clusters. This indicates that FCM and KFCM can produce better clustered in according to the external labels. Detail comparison for optimal number of clusters can be found in Table 3.3, where FCM and KFCM achieve better performances as well.



(a) ARI for *Yeast 384*



(b) ARI for *Yeast 237*



(c) ARI for *Rat CNS*

Figure 3. 14 ARI for three datasets
(Line 1-6 represent k-means, hierarchical clustering, SOM, EM, FCM and KFCM respectively)

Table 3. 4 ARI for optimal number of clusters

	k-means	Hierarchical	SOM	MODEL	FCM	KFCM
Yeast 384	**0.396**	**0.390**	**0.309**	**0.354**	**0.353**	**0.382**
Yeast 237	**0.355**	**0.334**	**0.355**	**0.355**	**0.355**	**0.353**
Rat CNS	**0.294**	**0.202**	**0.281**	**0.330**	**0.312**	**0.346**

Setting a high value for cutoff, clusters become distinct and genes in each cluster will have highly correlated expression patterns in all of the experiments that will be closely related in terms of function and regulation. As the cutoff decreases, clusters become fuzzy and additional genes will be assigned to each cluster groups (Audrey and Michael, 2002). In order to demonstrate the fuzzy attribute of FCM, the membership cutoff (0.04, 0.06, 0.08, 0.10) is used to assign genes to all of the clusters for the three datasets. Table 3.5 shows the number of genes that were assigned to more than one cluster. When the membership cutoff decreases from 0.10

to 0.04, genes placed in more than one group increase for the three datasets. This experiment also demonstrates that gene expression data are frequently connected and clusters are often highly intersected with each other. Setting appropriate cutoff is crucial to understanding gene function (Gasch and Eisen, 2002). Cutoff value in Table 3.5 will be used in the thesis.

Table 3. 5 Fuzzy assignment of genes to clusters for three gene expression data

Membership Cutoff	Number and percentage of genes assigned to > 1 cluster		
	Yeast 384	*Yeast 237*	*Rat CNS*
0.10	**25 (6.5%)**	**10 (4.1%)**	**9 (4.6%)**
0.08	**35 (9.1%)**	**15 (6.3%)**	**17 (8.1%)**
0.06	**43 (11.3%)**	**36 (15.4%)**	**19 (9.3%)**
0.04	**190 (49.6%)**	**101 (42.8%)**	**52 (25.3%)**

3.6 Discussion and Research Motivation

k-means, FCM and KFCM belong to partitional clustering algorithms, all of them partitions the data into groups and each group represents one cluster. However, in *k*-means, each object belongs to exactly one cluster, the partition is crisp; otherwise, in FCM and KFCM, the partition is marked as fuzzy and one object can be classified into more than one groups. Gene expression data is likely to contain overlapping clusters and not always follows standard distributions, which results in the crisp clustering methods not satisfactory.

In term of FCM, it incorporates Euclidean distance to calculate the object's similarities, which in only effective find spherical clusters. FCM assigns memberships to different objects, and it treats all objects equally in the clustering process. However, due to the non-uniformly and asymmetrically distribution, different samples play different roles in the clustering process. Hence, it is very useful to give an appropriate weight to the objects in cluster analysis. Local structure is an important concept and it offers the local information in the clustering process, by which, Local FCM can differentiates the contribution of different objects and

makes the clustering result more accurate in Chapter 4.

In order to find arbitrary shapes of clusters, kernel method is introduced to FCM to increase the linear representation ability. Although KFCM is good at finding more various shapes of clusters, it is not robust to the noise and outliers. The large component of noise in gene expression data makes KFCM lose its effectiveness. Moreover, KFCM needs user to specify the parameters which are usually unknown in advance, such as: the initial cluster centres, σ in the kernel. In order to avoid KFCM depend on the prior knowledge or trapping into local minimum, partial knowledge (DKFCM) is utilized in Chapter 5 to guide the clustering process. Experiments on gene expression data show that the proposed method substantially outperforms conventional models in term of stability and cluster quality.

Time series microarray is a special category of gene expression data, which is characterized by time dependency. Most previous works (FCM, KFCM *etc*) analyzing this type of data are developed originally for static data by neglecting the time series characteristics. In order to make FCM more effective, the time series characteristic is investigated, gene expression data is smoothed by cubic spline to minimize the influence of noise and random variation. By tuning the smoothing parameter, it can be smoothed with statistical consideration. Results in Chapter 6 demonstrate that the proposed method has substantial advantages over FCM for time-series gene expression data.

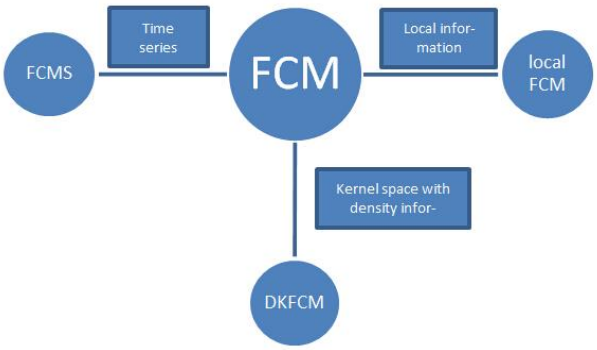


Figure 3. 16 Relationship between the proposed methods

3.7 Conclusion

Microarray data contains plenty of uncertain and imprecise information. FCM is an efficient model to deal with this type of data. By analyzing the performance of different clustering methods for artificial and gene expression data, FCM algorithm has been proven an effective method of clustering for Microarray data by provides a more stable clustering result. For datasets composed of hyper spherically shape separated clusters, FCM discovers these clusters accurately. However, due to the diversity structure and nonlinear relationship, the conventional FCM which incorporate Euclidean distance to compute the similarity between genes is not reliable in dealing this issue. Kernel metric is introduced to overcome the limitation and find more meaningful clusters. KFCM finds clusters in the feature space with higher even infinite dimension, in which the nonlinear relationship could be classified by a hyper plane. KFCM is more effective than FCM because it is not only finds spherical clusters, but also detect nonlinear relationship between gene profiles. Experiment shows that KFCM has the better performance when the data includes more diversity inherent structure, e.g. non-spherical clusters, noise etc. Therefore, KFCM is more reliable than FCM for gene expression data analysis. Finally, a discussion is given to address the limitations of the clustering methods, and motivations for the proposed methods are presented.

Chapter 4 Local weighted FCM for Microarray data analysis

4.1 Introduction

This chapter focuses on the improvement of FCM based on its limitation. Although FCM assigns memberships to different samples, it treats all samples equally in the clustering process. However, due to the non-uniformly and asymmetrically distribution of samples, different samples play different roles in the clustering process. Moreover, a sample may contribute to the clustering results differently in different processes. Hence, it is very useful to give an appropriate sample weight in cluster analysis. For this purpose, sample weighting clustering algorithms have been proposed in literature (Krinidis and Chatzis, 2010; Nock and Nielsen, 2004; Van and Kim, 2009). In sample weighted clustering, the weight determines the impact of the sample on the clustering process. Conditional fuzzy C-means (Kim and Ryu, 2002) and generalized fuzzy C-means clustering (Van Lung and Kim, 2009) consider various contributions of different samples and take account of sample weighting in the clustering process. However, the applications of the above algorithms are limited because they need users to weight samples. To overcome the problem, Krinidis and Chatzis (2010) proposed a formalized clustering framework, which offers weights by penalizing solutions on the samples and the sample weight can be automatically determined during the process of clustering.

Local structure is a popular technique prevailing in pattern recognition and image processing. It accentuates the neighborhood and captures details information to learn (Richard *et al*., 2001). Noordam *et al*. (2000) proposed a geometrically guided FCM algorithm for image segmentation, where a geometrical condition is used by taking into account the local neighborhood of each pixel. Wang *et al.* (2010) effectively utilize the structure information by building a graph incorporating neighborhood information of the dataset for pattern recognition. However, to the best of researcher's knowledge, there is no clustering method for gene expression analysis utilizing local structure information. In this chapter, motivated by the idea of preserving the neighborhood structure, a local weighting scheme for clustering gene expression data is proposed. Local FCM (LFCM) accentuates the objects in the neighborhood by assigning proper weights, so that LFCM can mainly describe the neighborhood structure of the data (Wang and Angelova, 2012). The advantage of this method is that it produces quality clusters and can handle noisy datasets.

4.2 Local weighted FCM

Due to the variability in the measurement or experimental error, gene expression data often contains a huge amount of noise. Clustering algorithms for this data should be capable of extracting useful information from a high level of background noise. However, FCM cannot differentiate the noise and meaningful data. In order to overcome the drawbacks and make it more robust, LFCM is proposed with the following objective function:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} w_{ij} u_{ij}^{m} \left\| x_j - v_i \right\|^2 \qquad (4.1)$$

where w_{ij} is the weight parameter, and it describe the importance of sample x_j to centres v_i. m is the fuzzy exponent which determines the amount of fuzziness of the resulting classification. In order to preserve the neighborhood structure, the weighting function is defined as:

$$w_{ij} = e^{-(\|x_j - v\|)\eta} \tag{4.2}$$

where η_i is a scaling parameter. when $\eta_i \to 0$, the weight $w_{ij} \to 0$, all samples have the same weight and the clustering will produce poor clusters. On the other hand, when $\eta_i \to \infty$, the weight matrix has all entries equal to 1, and thus the weighted clustering is reduced to non-weighted clustering. In order to choose appropriate values for the weights, a local scale η_i can be computed by:

$$\eta_i = \begin{cases} \sigma_i^2 & x_j \in N_{ik} \\ (\frac{1}{c}\sum_{i=1}^{c}\sigma_i)^2 & otherwise \end{cases} \tag{4.3}$$

$$\sigma_i = (1/k \sum_{j=1}^{k} \| x_j - v_i \|^2 \tag{4.4}$$

where *k* is the number of neighbours of the *i* th cluster centre. N_{ik} is the *k*-nearest neighbours of the *i* th cluster. Figure 4.1 illustrate that the *k*-nearest samples around two cluster centres. The distance variance σ_i^2 of the cluster centre *i* represents the degree of aggregation around the clusters centres. The small value of variance indicates that the clusters are compact and well separated. If the dataset with distinct clusters, σ_i^2 should be as small as possible. However, if the dataset with fuzzy or undistinguished clusters, σ_i^2 should be given a large value to suppress the noises.

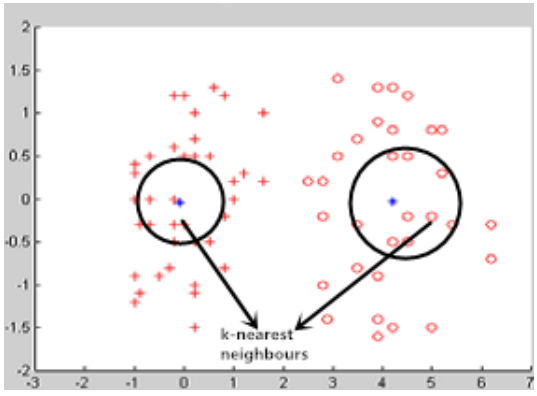


Figure 4. 1 k-nearest neighbours with more influence to clustering

Equation 4.3 shows that the scale can automatically adapt to the local structure. By definition, each sample x_j satisfies the constraint that $\sum_{i=1}^{C} u_{ij} = 1$. In order to obtain the solution of the LFCM, the objective function can be minimized by:

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N} w_{ij} u_{ij}^{m} \left\| x_j - v_i \right\|^2 + \sum_{j=1}^{N} \lambda_j \left(\sum_{i=1}^{C} u_{ij} - 1\right) \tag{4.5}$$

Suppose that $\frac{\partial J}{\partial v_i} = 0$, the cluster centres can be obtained by,

$$v_i = \frac{\sum_{j=1}^{n} w_{ij} u_{ij}^{m} \left\| x_j - v_i \right\|^2 x_j}{\sum_{j=1}^{n} w_{ij} u_{ij}^{m} \left\| x_j - v_i \right\|^2} \tag{4.6}$$

In order to get the optimization membership, setting $\frac{\partial J}{\partial u_{ij}} = 0$

$$\frac{\partial J}{\partial u_{ij}} = m u_{ij}^{m-1} w_{ij} \left\| x_j - v_i \right\|^2 + \lambda_j \tag{4.7}$$

then

$$u_{ij} = \left(\frac{-\lambda_j}{m w_{ij} \left\| x_j - v_i \right\|^2}\right)^{1/(m-1)} \tag{4.8}$$

According to $\sum_{i=1}^{C} u_{ij} = 1$ and equation (4.8),

$$(-\lambda_j)^{1/(m-1)} = \left(\sum_{k=1}^{C} \frac{1}{m w_{ij} \left\| x_j - v_i \right\|^2}\right)^{1/(m-1)} \tag{4.9}$$

Equation (4.9) can be re-formulated,

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{w_{ij} \left\| x_j - v_i \right\|^2}{w_{ik} \left\| x_k - v_i \right\|^2}\right)^{1/(m-1)}} \tag{4.10}$$

Based on the equation (4.6) and equation (4.10), the following algorithm is proposed,

Local Fuzzy c-means (LFCM). Given *N* data $X = \{x_j\}_j^N$ and the desired number of cluster *C*, output a membership matrix $U = \{u_{ij}\}$

1: Initialize number of clusters *C*, and fuzzy exponent parameter *m*
2: Initialize iteration counter $k = 0$;
3: Initialize the local fuzzy *C* partition matrix U^0;
4: Compute the initial prototypes v_i
5: Repeat**:**
6: (a) Update all memberships U^0 with Equation 4.10;
7: (b) Update all prototypes v_i with Equation 4.6;
8: Until **(prototype parameters stabilize)**

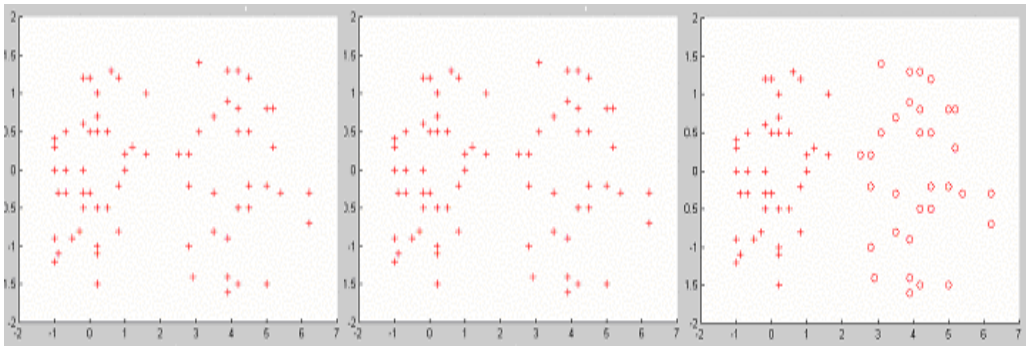
4.3 Experiments and results

In order to evaluate the proposed method, experiments are carried out to compare the performance with FCM and KFCM for artificial data and gene expression data. In KFCM, σ in the kernel function is set 150 empirically.

4.3.1 Artificial data

Experiment 1: Two-cluster dataset with an outlier

Noise in gene expression data usually emerge as outliers which leads the traditional algorithms lacking robustness. In order to evaluate the robustness of the proposed algorithm, a two-cluster dataset is generated by adding an outlier (20, 0). The clustering results of FCM, KFCM and LFCM are shown in Figure 4.2 (a)-(c) respectively. It can be seen that FCM and KFCM are heavily affected by the outlier and all data are merged in one cluster. Figure 4.2(c) shows that LFCM can detect two clusters correctly by indicating that the proposed method is immune to the outlier.

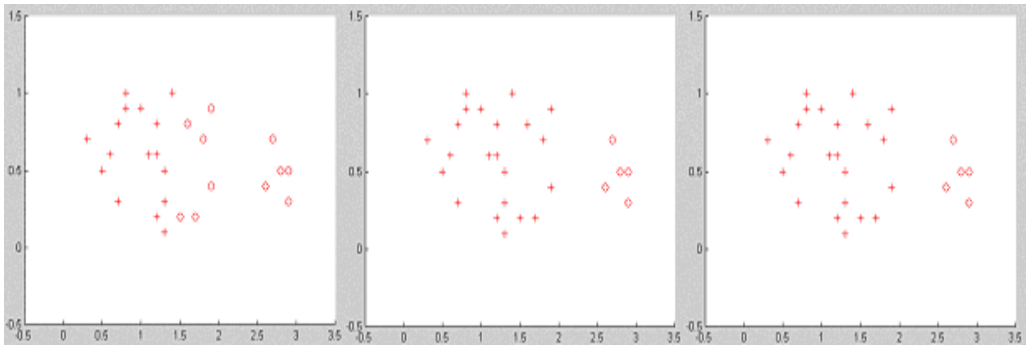


(a) FCM (b) KFCM (c) LFCM

Figure 4. 2 Clustering result for two-cluster data with noise

Experiment 2: Unbalance clusters

This dataset contains two clusters with various volumes. Figure 4.3 (a) shows that FCM cannot detect the unequal two-cluster correctly, while LFCM and KFCM exhibit similar performance by detecting the clusters correctly. It indicates that LFCM can identify clusters with different sizes.

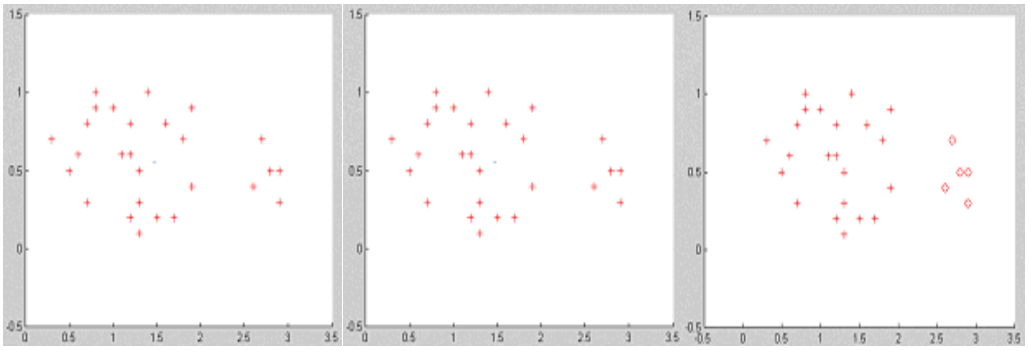


(a) FCM (b) KFCM (c) LFCM

Figure 4. 3 Clustering results for unbalance cluster data

Experiment 3: Unbalance clusters with an outlier

In this experiment, an outlier (10, 0) is added to the unbalance dataset. Experimental results are shown in Figure 4.4 (a)-(c). It can be seen that both FCM and KFCM cannot detect the correct patterns. However, LFCM can identify the correct patterns in the dataset and is not affected by the outlier. It accentuates of the samples in neighborhood and assigns small weights to the samples outside the neighborhood.

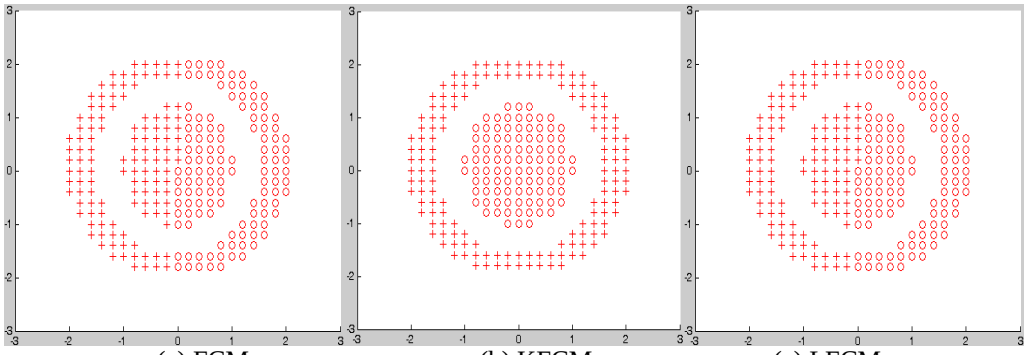


(a) FCM (b) KFCM (c) LFCM

Figure 4. 4 Clustering result for unbalance data with noise

Experiment 4: Ring data

Ring data (Chiang and Hao, 2003) is used to evaluate LFCM ability on identifying the arbitrary shapes of clusters. Figure 4.5 shows that FCM and LFCM incorporating Euclidean distance cannot cope with this data, while the incorporation of kernel metrics renders KFCM immune to unreliable feature and identify the correct structure.



(a) FCM (b) KFCM (c) LFCM

Figure 4. 5 Clustering result for Ring data

The number of neighbours *k* is crucial for clustering result. In these experiments, *k* is set to 20 empirically for artificial data. In order to show its influence, ARI is used to assess the clustering result for different number of neighbours for the Two-cluster dataset. Figure 4.6 shows that when *k* is in the range (16, 18), LFCM achieves the best performance. If selecting small number of neighbors, LFCM shows poor performs due to lack of information describing the local structure, while large number of neighbours involves the data belonging to other clusters

and leads the algorithm lose effectiveness. The selection of optimal number of nearest neighbours will be further studied.

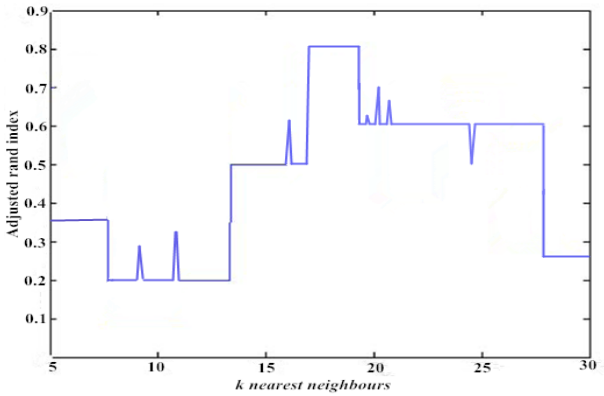
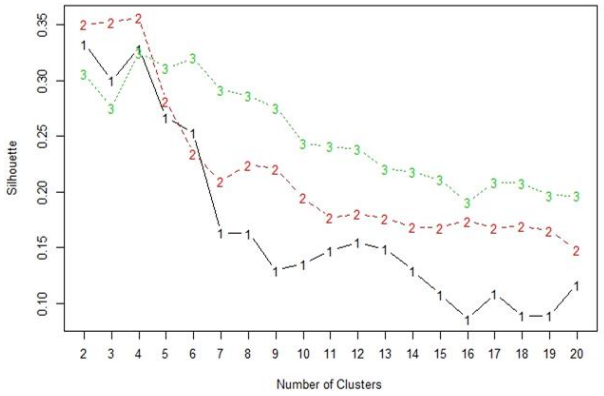


Figure 4. 6 *k* neighbours *vs* adjusted rand index

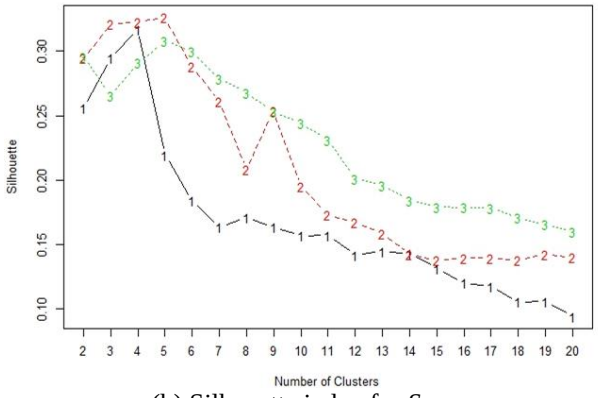
4.3.2 Gene expression data

In order to evaluate the performance on gene expression data, two gene expression datasets are selected: *Yeast 384 and Serum*. Silhouette index and ARI are used to assess the quality of the clusters. BHI and FOM are used to assess the cluster's biological significance and stability of the algorithms. The numbers of neighbours are set 26 and 40 for *Yeast 384 and Serum* respectively.

For silhouette index, Figure 4.7 (a) shows that LFCM performs better than the other two methods for *Yeast 384* on a range number of clusters (5, 20), which indicates that the clusters produced by LFCM are more intra compact and inter separated. Similar result can be found in Figure 4.7(b) for *Serum*.



(a) Silhouette index for *Yeast 384*



(b) Silhouette index for *Serum*

Figure 4. 7 Silhouette index for two sets of gene expression data
(Line 1,2,3 represent FCM, KFCM and LFCM respectively.)

ARI is used to assess the cluster accuracy of the three algorithms. By preserving the local structure, LFCM performs better than the other two methods (Figure 4.8). Although it does not generate higher values in the initial number of clusters (2-7), it peaks at the optimum number of clusters 5. ARI cannot be used for *Serum*, because there is no external criterion for this data.

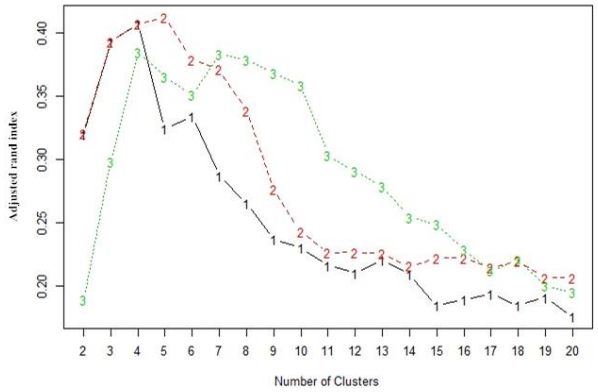
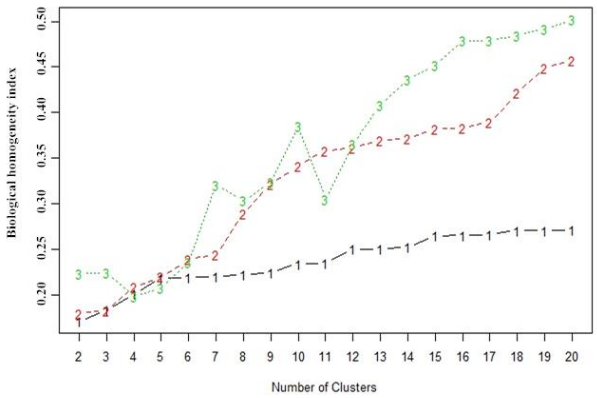
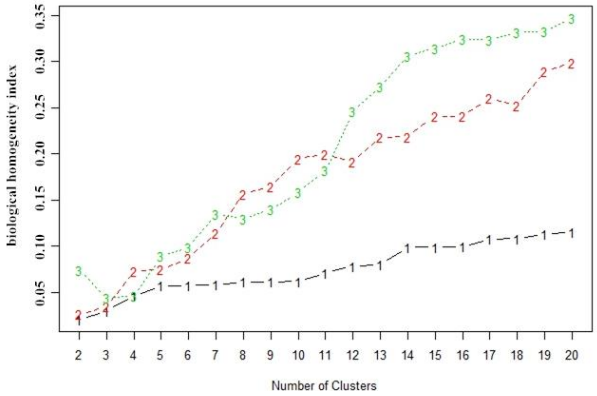


Figure 4. 8 ARI for *Yeast 384*
(Line 1,2,3 represent FCM, KFCM and LFCM respectively.)

In terms of biological significance, BHI is used to assess the produced cluster as it explicitly specifies the functional clustering of the genes. *R* package FatiGO (Al-Shahrour *et al.*, 2004) is used to annotate the functional classes of the genes. The functional categorization of the genes in the dataset were previously determined by Cho *et al*.(1999) and Iyer *et al* (1999), so these will be used initially to define the functional classes. It can be seen from Figure 4.9 (a) that LFCM outperforms the other two methods for *Yeast 384* for most number of clusters, similar performance can be found in Figure 4.9 (b) for *Serum*.



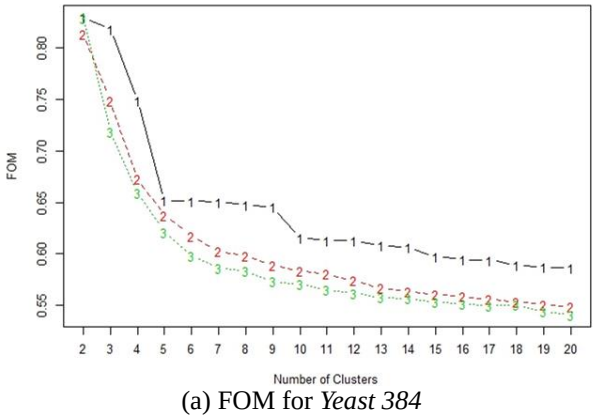
(a) BHI for *Yeast 384*



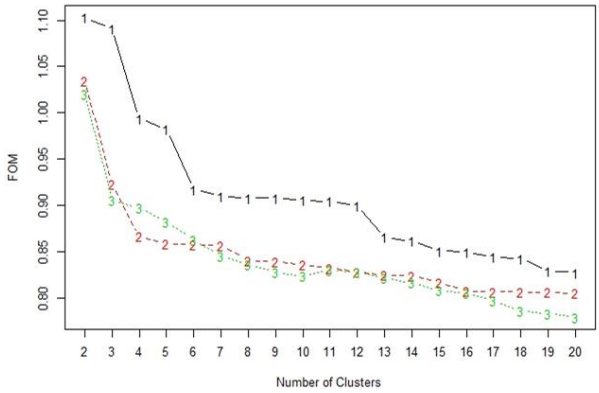
(b) BHI for *Serum*

Figure 4. 9 BHI for two sets of gene expression data
(Line 1,2,3 represent FCM, KFCM and LFCM respectively.)

In term of stability, it can be seen from Figure 4.10 (a)-(b) that LFCM shows the best performance for the two datasets, which indicates that LFCM has a high level reliability. In Figure 4.10 (a) the value of FOM for the three algorithms decreases steeply until the number of clusters reaches 5, which suggests that the three algorithms perform optimally for 5 clusters and any additional clusters produced will not add much predictive value to the algorithms.



(a) FOM for *Yeast 384*



(b) FOM for *Serum*

Figure 4. 10 FOM for two sets of gene expression data
(Line 1,2,3 represent FCM, KFCM and LFCM respectively.)

4.4 Conclusion

Gene expression data is a one-time expression of hundreds of thousands of genes obtained with microarray technology, which includes a large component of noise (Andreas and Francis, 2005). However, FCM and KFCM assigning equal weights to noises and meaningful data makes the results lack of biological significance. In this research, a local FCM is proposed by accentuating the objects in neighborhood. Experiments on artificial data and gene expression data show that the proposed method is not only robust to the noise, but also identifies the unbalance clusters. In addition, clustering results for gene expression data show that LFCM can produce stable clusters which have better agreement with the biological interpretation.

Chapter 5 Density weighted kernel fuzzy c-means on gene expression analysis

5.1 Introduction

In FCM algorithm, each object has the same influence to data classification, which however is in not correct in practical classification process, especially in gene expression analysis (Chuang *et al*., 2006). For instance, one gene has a tendency for typical genes to consider a great influence to classification of the data, and contrarily, for ambiguous data to consider little influence to classification of data set. In order to differentiate the various objects importance, many variations of FCM have been proposed in the past years, Fuzzy J-Means that applies variable neighborhood searching to avoid cluster solution being trapped in local minima (Belacel *et al.*, 2004). A Fuzzy-SOM approach is developed to improve FCM by arraying the cluster centroids into a regular grid (Pascual-Marqui *et al*., 2001). Asyali and Alci (2005) employ normal mixture modeling to fit microarray data and then use FCM to identify the clusters. Fu and Medico (2007) proposes a novel fuzzy clustering method (FLAME) for the analysis of DNA microarray data, it can captures non-linear relationship and non-globular clusters. Pal *et al*. (2007) uses neural networks and relational fuzzy clustering for discovering biomarkers from gene expression data for predicting cancer subgroups. Veit and Ole (2010) proposed a simple and fast method to determine the parameters for FCM analysis of

gene expression data. Wang *et al*. (2013) proposed a fuzzy clustering approach by fitting expression data with cubic spline. Although these algorithms improve the clustering performance such as the producing internally homogeneous clusters and finding diverse structures, these methods need user to specify the parameters which are usually unknown in advance, such as: neighborhood of each object should be defined and archetype feature need to be identified (Fu and Medico, 2007), smoothing parameter is needed to be specified (Wang *et al*., 2013).

In this chapter, a new fuzzy clustering approach (DKFCM) is proposed which computes gene similarity in the kernel space. In addition, an initialization method is proposed by employing Parzen density estimation. Based on the FCM framework, the objective function is modified by adding a new weighted parameter. Furthermore, this approach incorporates a parameters selection process which automatically finds optimal values for parameters in the clustering process. Experiments on artificial data and real gene expression show that the proposed method outperforms the conventional methods substantially.

5.2 Density weighted kernel FCM

The proposed approach contains two integrated processes: initialization by Parzen density function and density weighted kernel FCM. In addition, this approach gives a parameters selection scheme, which not only obtains the optimum values for the clustering process, but also avoids the algorithm trapping into local minimum.

5.2.1 Initialization by Parzen density function

FCM is sensitive to the initialization. Figure 5.1 shows 200 different initializations of FCM for *Yeast 384* (Yeung *et al.*, 2001). ARI is used to examine the clustering results. It fluctuates with various initializations, where the maximal ARI is 53.14% and the minimal one is 26.81%. To address this instability, the

conventional method is usually to repeat the algorithm and select the initialization when the objective function converges to the smallest value. However, this approach is time consuming and computationally expensive.

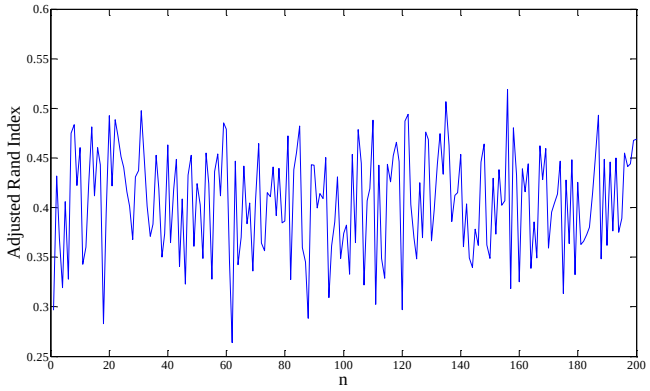
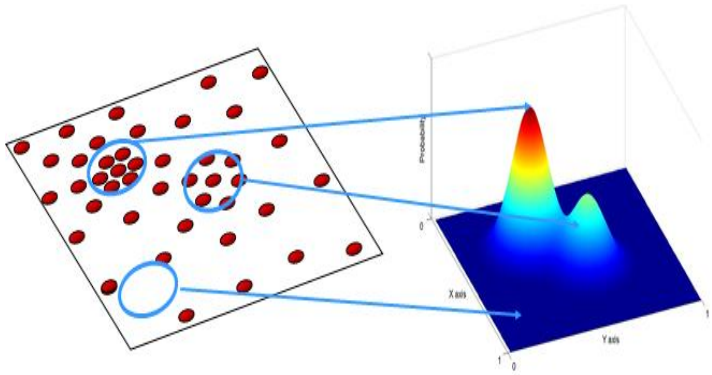


Figure 5. 1 ARI vs random initial cluster centre

Usually, the high dense area is considered as a cluster, where data objects are attracted with each other. At the core part of the dense area, object are crowded closely with each other, and thus have high density. Objects at the peripheral area of the cluster are relatively sparsely distributed, and are attracted to the core part of the dense area (Jiang and Zhang, 2003). Therefore, it is assumed that 'good' cluster centres should be with high density values. Based on this assumption, an initialization method is proposed by utilizing Parzen density estimation (John and Nello, 2004). The objects with highest density values are chosen as the initial cluster centres. Figure 5.2 shows that given a dataset (in Figure 5.2 (a)), objects with high density values estimated by Parzen density function (PDF) are chosen as the initial cluster centres.



(a) Data distribution (b) PDF estimation

Figure 5. 2 Parzen density estimation

Parzen density function is a nonparametric estimation technique which estimates data density without any prior assumption on the data distribution (John and Nello, 2004). Suppose $X=\{x_1, x_2,\ldots, x_N\}$ is a dataset including N data in d conditions, the cluster number is C. the density of x is estimated by $p(x)$,

$$p(x)=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\varphi(u) \tag{5.1}$$

where N is the number of data, V is the volume of the hypercube that centres at x, radius is h, $V=h^d$. $\varphi(u)$ is the window function. Many standard windows have been adopted in pattern recognition and machine learning.

Rectangular window

$$\varphi(u)=\begin{cases}1 & |u|\le\frac{1}{2}\\ 0 & others\end{cases} \tag{5.2}$$

Normal window

$$\varphi(u)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\|u\|^2\right) \tag{5.3}$$

Exponential window

$$\varphi(u)=\exp(-|u|) \tag{5.4}$$

The parameter u in the Parzen function can be calculated by,

$$u=\frac{x-x_i}{h} \tag{5.5}$$

Considering the Normal window as an example，$\varphi(u)$ is a hypercube centered at original point. When x_i falls into the hypercube which centres at x, $\varphi(u)=1$, else $\varphi(u)=0$. Therefore, the number of the samples in the hypercube is：

$$k = \sum_{i=1}^{N} \varphi\left(\frac{x - x_i}{h}\right) \tag{5.6}$$

The selection of initial cluster centres is basically a selection of C local extreme density values. Due to kernel method is more appropriate for gene expression data as shown in chapter 4, the density is computed in kernel space. Let ϕ is the kernel function, Equation (5.5) is written：

$$u = \frac{\phi(x) - \phi(x_i)}{h} \tag{5.7}$$

where

$$\begin{aligned} \|u\|^2 &= \frac{1}{h^2}\left(\phi(x) - \phi(x_i)\right)^T \left(\phi(x) - \phi(x_i)\right) \\ &= \frac{1}{h^2}\left(k(x,x) - 2k(x_i,x) + k(x_i,x_i)\right) \end{aligned} \tag{5.8}$$

Given Gaussian kernel，$k(x,x)=1$, Equation (5.8) is simplifies to：

$$\|u\|^2 = \frac{2}{h^2}\left(1 - k(x_i,x)\right) \tag{5.9}$$

The density estimation of each x_i in the Gaussian kernel space is：

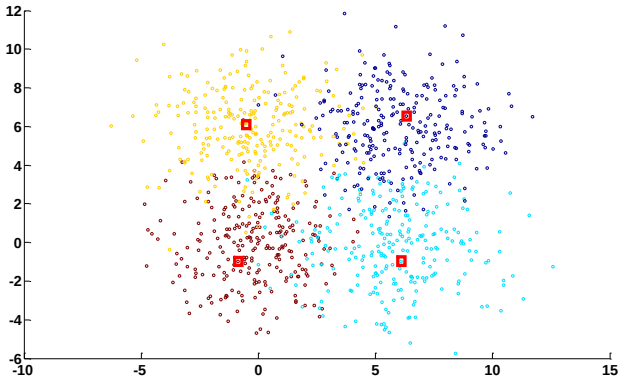
$$\begin{aligned} p(x_i) &= \frac{1}{NV}\sum_{j}^{N} \varphi(u) \\ &= \frac{1}{NV\sqrt{2\pi}} \sum_{j=1}^{N}\left(\exp\left(-\frac{1}{2}\|u\|^2\right)\right) \\ &= \frac{1}{NV\sqrt{2\pi}} \sum_{j=1}^{N}\left(\exp\left(-\frac{1}{h^2}\left(1 - k(x_j,x_i)\right)\right)\right) \end{aligned} \tag{5.10}$$

One object has a high density value if many objects are in its neighborhood. The first cluster centre C_1 is chosen as the sample having the extreme density value P_i . Next, the density of each sample x_i is revised as,

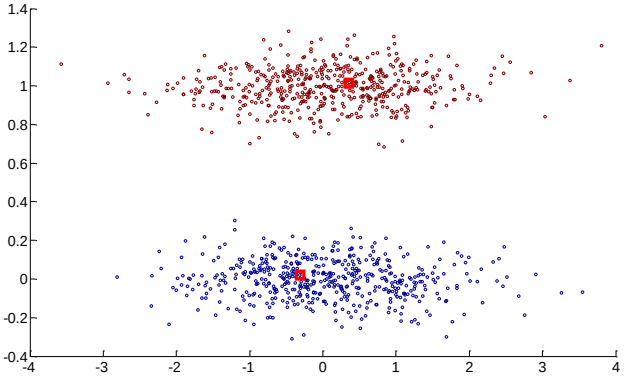
$$P_i = P_i - P_{C_1}\frac{1}{1 + \left(\phi(x_i) - \phi(x_k)\right)}, \quad k = 1, 2, \ldots \tag{5.11}$$

In order to evaluate the performance of the initialization method, two artificial

datasets with different distribution are produced (Figure 5.3 (a) and (b)). Figure 5.3 (a) includes four clusters with overlap area, while Figure 5.3 (b) includes two elliptical clusters. The proposed algorithm can detect the optimal cluster centres for the two datasets. (Red squares mark the detected cluster centres).



(a) Four clusters with overlap area



(b) Two independent elliptical clusters

Figure 5. 3 Detection of cluster centres
(The detected cluster centres are marked by square)

5.2.2 Weighted kernel fuzzy c-means

Recall in chapter 3, KFCM is sensitive to noise. The membership values in KFCM give equal weights to noise and meaningful data. In order to increase the robustness of the algorithm, a weight algorithm is proposed which can differentiate noise from meaningful data. Specifically, each sample will be assigned a weight based on its density values $p(x_j)$, which describes the spatial characteristic of samples in feature space. Samples with higher density values will have

greater influence to the clustering process. The weight w_j of sample j is computed by,

$$w_j = p(x_j) \Big/ \sum_{i=1}^{N} p(x_i) \tag{5.12}$$

The objective function is written,

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} w_j \mu_{ij}^m \left\| \phi(x_j) - \phi(v_i) \right\|^2 \tag{5.13}$$

By utilizing Lagrange multipliers, the optimal fuzzy partition matrix $\boldsymbol{U}$ and the optimal cluster centre matrix $\boldsymbol{V}$ are obtained by equation 5.14 and 5.15 respectively.

$$u_{ij} = \frac{(1/(1-K(x_j, v_i)))^{1/(m-1)}}{\sum_{i=1}^{c} (1/(1-K(x_j, v_i)))^{1/(m-1)}} \tag{5.14}$$

$$v_i = \frac{\sum_{j=1}^{n} w_j u_{ij}^m K(x_j, v_i)}{\sum_{j=1}^{n} w_j u_{ij}^m K(x_j, v_i)} \tag{5.15}$$

5.3 Parameter selection

The smoothing parameter h in Parzen density function and σ in Gaussian kernel function are discussed in this section.

5.3.1 Selection of the smoothing parameter *h*

In Parzen density estimation, h is a bandwidth parameter which exhibits a strong influence on the resulting estimation. Given some random samples from the standard normal distribution in Figure 5.4, where the grey curve is the true density (with mean 0 and variance 1). In comparison, the red curve is under smoothed by using a bandwidth of h =0.05 which includes many spurious data artifacts because

the bandwidth is small. The green curve is over smoothed and neglects a lot of detailed information since using the bandwidth $h = 2$. The black curve using a bandwidth of $h = 0.337$ is considered to be optimally smoothed since its density estimate is close to the true density.

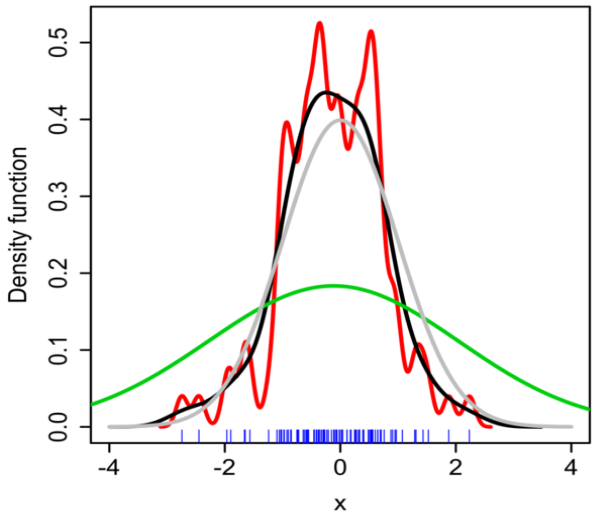


Figure 5. 4 Density function with different h

Intuitively one wants to give h as small value as the data allows because it will detect more information of the data distribution. However, there is a trade-off between the bias of the estimator and its variance. In gene expression data analysis, if choosing a small h, noise will be involved in the estimation process that consequently makes the estimation lacks of reliability. On the contrary, large h makes the density function too smooth to get the detail information. *Yeast 384* is used to test the influence of h on the clustering accuracy, where σ is set for 150.

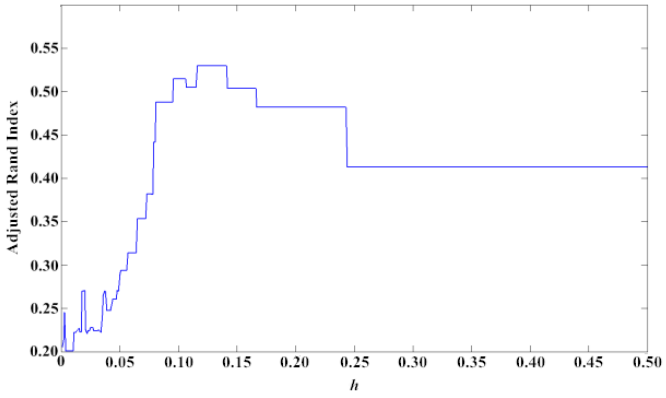


Figure 5. 5 ARI vs h for *Yeast 384*

Figure 5.5 shows that when *h* falls in the region [0.10, 0.15], ARI achieves better performance than choosing other values. Setting *h* appropriately can detect optimal initial cluster centres. In the study, a selection of *h* is given as follows,

---

1：**Initialize σ =150;**

2：**Calculate the distance of each pair of data points in kernel space using;**

$$\begin{aligned} D_{ij} &= D(x_i, x_j) = K(x_i, x_i) - 2K(x_i, x_j) + K(x_j, x_j) \\ &= 2 - 2 * K(x_i, x_j) \\ &= 2 - 2 * \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2} \right) \quad x_i, x_j \in R^d \end{aligned} \tag{5.16}$$

3**: Sort the distance according to increasing order;**

4**: Use the sum of first *k* minimal distance as the value of *h*.**

$$h = \sum_{m=1}^{k} D_{ij}^{m} \tag{5.17}$$

where $D_{ij}^{m}, m = 1, 2, ..., k$ **are the *k* minimal distance.**

5.3.2 Selection of the Gaussian parameter σ

The parameter σ determines the width of the Gaussian,

$$k(x, z, \sigma) = \exp(-\frac{\left\| x - z \right\|^2}{2\sigma^2}) \tag{5.18}$$

where *x*, *z* are samples, σ the standard deviation in Gaussian probability density function. Gaussian can be considered as an aperture function of some observation, where σ is the scale and $\sigma > 0$. The integral over the exponential function is not unity: $\int_{-\infty}^{\infty} e^{-x^2/2\sigma^2} dx = \sqrt{2\pi}\sigma$. With the normalization constant this Gaussian kernel is a *normalized* kernel, which means that increasing the σ of the kernel reduces the amplitude substantially. Figure 5.6 shows the normalized kernels for $\sigma^2 = 0.2$, $\sigma^2 = 0.5$, $\sigma^2 = 1.0$ and $\sigma^2 = 5.0$ plotted on the same axes.

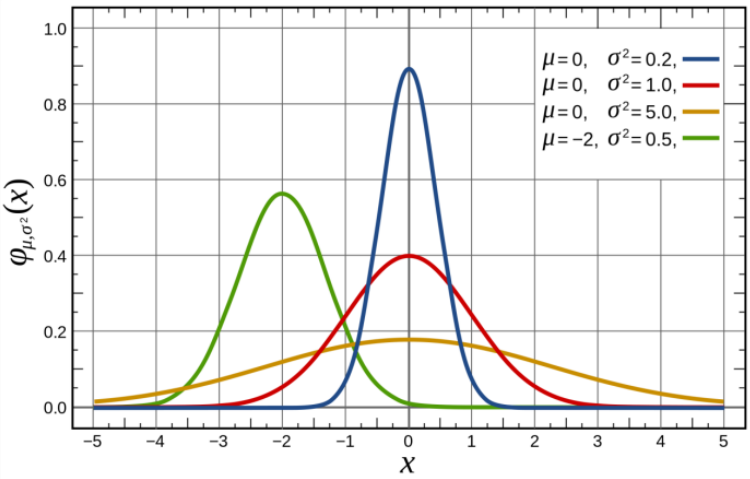


Figure 5. 6 Density function with different σ

The parameter σ in Gaussian kernel function also has an influence on the clustering result. *Yeast 384* is chosen to test the influence of σ on the clustering result. As shown in Figure 5.7, the clustering result fluctuates with σ. An optimal value of σ is essential for a successful clustering.

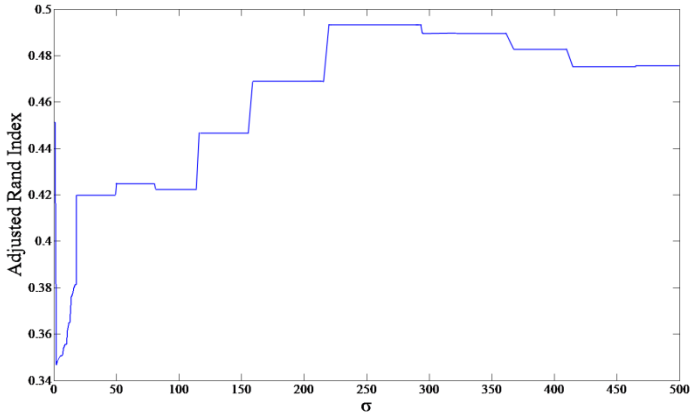


Figure 5. 7 ARI vs σ for *Yeast 384*

The parameter σ in Gaussian kernel has been given much investigation. However, most of the studies are based on SVM (John and Nello, 2004), while studies based on FCM are rarely seen. Clustering is generally recognized as an "unsupervised" learning problem. Prior to undertaking a clustering task, "global" information regarding the data set, such as the number of clusters and the complete data distribution in the object space, is usually unknown. However, some "partial" knowledge is often available regarding a gene expression dataset. Some genes are

strongly correlated, and the differences among the cluster structures under these different groups may be of particular interest. If a clustering algorithm could integrate such partial knowledge as some clustering constraints when carrying out the clustering task, the clustering result is expected to be more biologically meaningful. In this work, the "partial" knowledge is used to obtain the optimal value for σ.

The genes haves closest distances indicating that they are strongly correlated biologically. To utilize this information, n nearest genes around the cluster centres are selected to form a training dataset X', the cluster labels in X' is obviously known. As discussed in Chapter 4, kernel mapping has two properties: firstly, the objects in the same class should be mapped into the same area in the feature space; secondly, the objects in the different classes should be mapped into the different areas. The values of the Gaussian kernel function should be close to 1 if the samples are in the same class. The values of the Gaussian kernel function would be close to 0 if the samples are in the different classes.

$$\begin{cases} k(x,z,\sigma) \approx 1, x,z \in C_i \\ k(x,z,\sigma) \approx 0, x \in C_i, z \in C_j, i \neq j \end{cases} \tag{5.19}$$

where C_i denotes ith cluster.

Given Gaussian kernel function, the norm of every sample is one and positive, and the samples will be mapped onto the surface of a hyper sphere. If the parameter σ is close to 0, then the corresponding kernel function values are close to 0. This means that all samples in a feature space are all approximately mutually perpendicular. When σ increases, the values of the Gaussian kernel function with respect to the samples which are closer by applying the Euclidean distance in the original space increase fast. As σ is close to infinity, the corresponding kernel function values are all close to 1. So the samples in the feature space are close to a fixed point. Figure 5.8 shows the ideal distribution in the feature space.

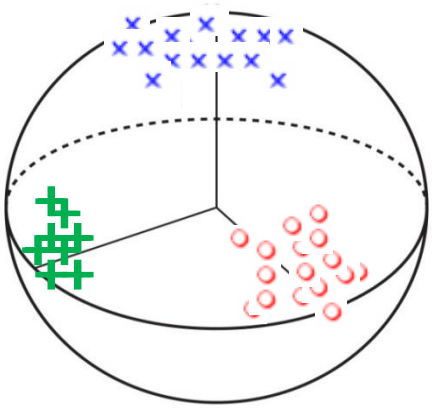


Figure 5. 8 Ideal distribution in the feature space

For example, given three vectors: $[0,1,1]^T, [0,1,0]^T, [1,0,0]^T$, where $[\ ]^T$ is the transpose operator. Table 5.1 show the vectors is the corresponding feature space with specific parameter σ. Table 5.1 shows that when σ=0.9, the three vectors is classified better than the other values, where vector *z* is close to vector *y* than the vector *x*, which is in accord to the data distribution in the original space.

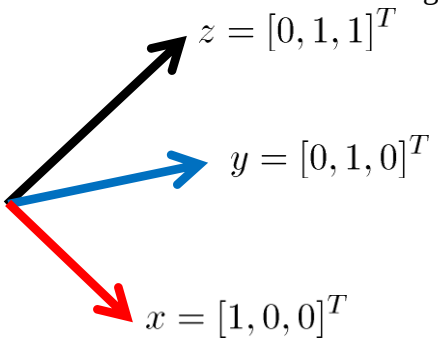
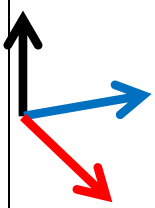
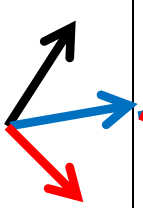
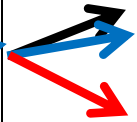


Figure 5. 9 vectors in original space

Table 5. 1 vectors vs varying parameter σ

	10^{-6}	0.6	0.9	2	100
$\|y-z\|=1$	0	0.2494	0.5394	0.8825	1.0000
$\|x-z\|=1.732$	0	0.0155	0.1569	0.6873	0.9999
Vectors in the feature space					

In this work, two criteria are proposed for measuring these properties. First one is the mean of the samples in the same class:

$$w(\sigma) = \frac{1}{\sum_{i=1}^{c} N_i^2} \sum_{i=1}^{c} \sum_{x \in \omega_i} \sum_{z \in \omega_i} k(x, z, \sigma) \tag{5.20}$$

where N_i is the number of training samples in class *i*. The parameter σ should be determined such that $w(\sigma)$ closes to 1.

Second one is the mean of samples in the different classes:

$$b(\sigma) = \frac{1}{\sum_{i=1}^{c} \sum_{\substack{j=1 \\ i \neq j}}^{c} N_i N_j} \sum_{i=1}^{c} \sum_{\substack{j=1 \\ i \neq j}}^{c} \sum_{x \in \omega_i} \sum_{z \in \omega_j} k(x, z, \sigma) \tag{5.21}$$

It is desire to find a parameter σ such that

$$\begin{cases} w(\sigma) \to 1 \\ b(\sigma) \to 0 \end{cases} \tag{5.22}$$

This means that

$$\begin{cases} 1 - w(\sigma) \to 0 \\ b(\sigma) \to 0 \end{cases} \tag{5.23}$$

Hence, the optimal σ^* can be obtained by solving the following optimization,

$$\min J(\sigma) = (1 - w(\sigma) + b(\sigma)) = (1 - w(\sigma) + b(\sigma) \tag{5.24}$$

Note that if k(x, z, σ) is differentiable, e.g., the kernel is Gaussian, with respect to σ, the gradient descent method (Chong and Zak, 2008),

$$\sigma_{n+1} = \sigma_n - \gamma_n \nabla J(\sigma_n), \quad \gamma_n > 0, n = 1, \tag{5.25}$$

is used to solve the proposed optimization problem, where

$$\nabla J(\sigma_n) = \frac{\partial}{\partial \sigma} b(\sigma_n) - \frac{\partial}{\partial \sigma} w(\sigma_n) \tag{5.26}$$

and γ_n is the step size at the nth iteration.

Otherwise, if the parameter σ is discrete, *e.g.*, the based kernel is polynomial kernel, then the best σ^* can be found that

$$\sigma^* = \arg\min\{J(\sigma)\big|\sigma = 1, 2, ..., s\} \tag{5.27}$$

where *s* is an integer and should be pre-determined.

5.3.3 Automatic parameter selection

The density weighted kernel FCM (DKFCM) is proposed and the flowchart is shown in Figure 5.10.

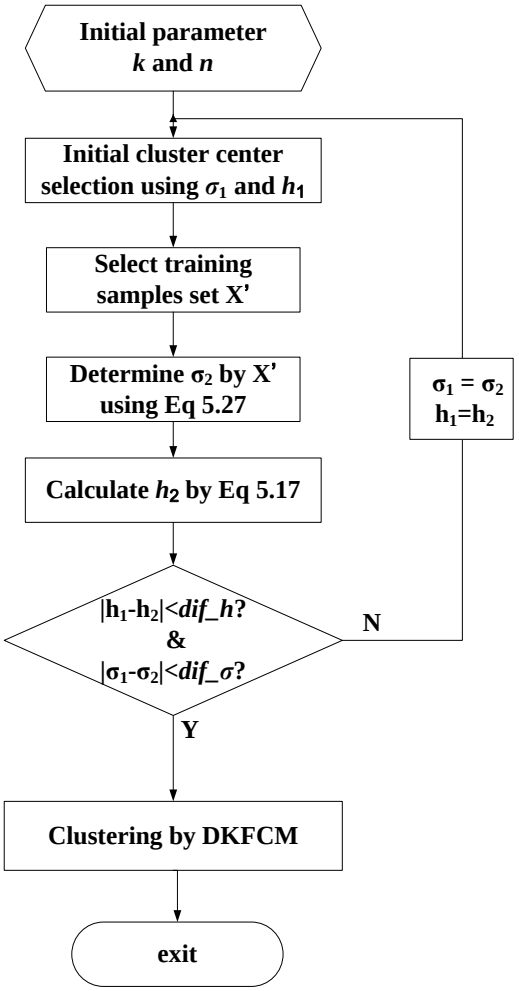


Figure 5. 10 Flowchart of DKFCM

The parameter *h* and σ are determined automatically in the iteration process. First,

k and *n* are initialized, and training set X' is established, where h_1 and σ_1 can be computed. The parameter σ_2 is determined using equation (5.27), then h_2 is computed by equation (5.17). *dif_h* and *dif_*σ are the differences between h_1 and h_2, σ_1 and σ_2 respectively. If the difference satisfies the threshold (*e.g.*, *dif_h* = *dif_*σ = 0.001), the loop ends, otherwise $h_1 = h_2, \sigma_1 = \sigma_2$ and the process continues. Finally, the algorithm obtains the optimal values of h_2 and σ_2.

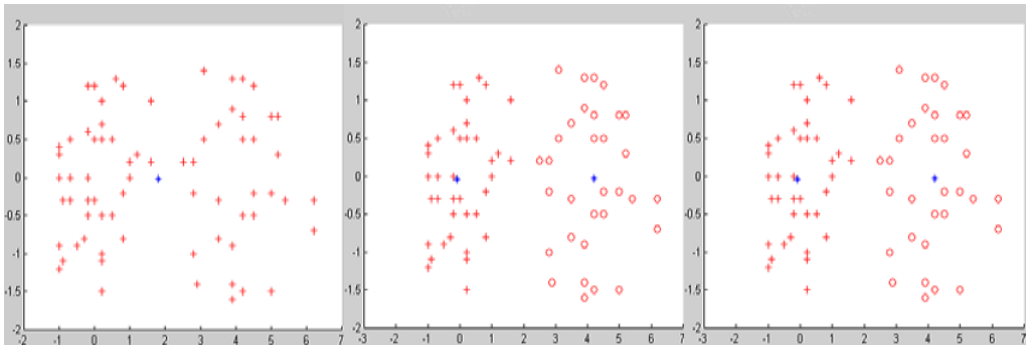
5.4 Experiments and results

5.4.1 Artificial data

In order to evaluate the performance of the proposed algorithm, KFCM and LFCM are used for comparison. Before experiments for artificial data, σ is set 150 for KFCM. For DKFCM, *k and n* are set 5 and 30 respectively.

Experiment 1: Two-cluster dataset with an outlier

In order to demonstrate the robustness, an outlier (20, 0) is added to the two-cluster dataset. Figure (a) shows the result of KFCM, the cluster is heavily affected by the outlier and become one centre alone. LFCM can identify two clusters in the dataset as shown in Figure 5.11(b). The result of DKFCM is shown in Figure 5.11(c), which also can detect the correct patterns.

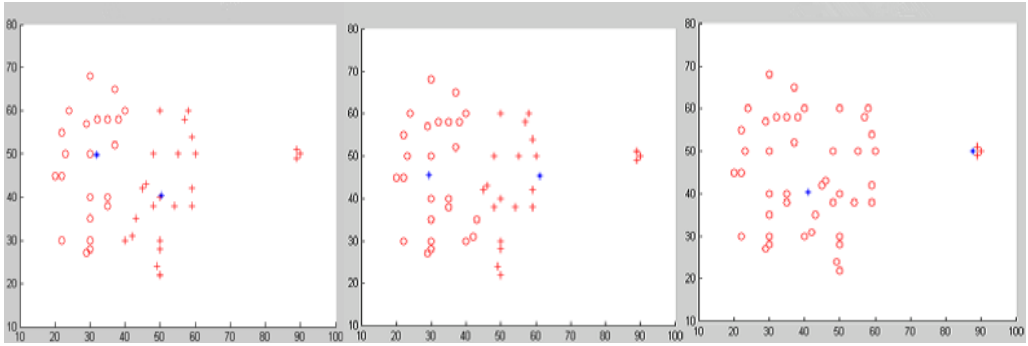


(a) KFCM (b) LFCM (c) DKFCM

Figure 6. 11 Clustering result for a two cluster data with noise

Experiment 2: Unbalance two-cluster dataset

Two clusters are produced with different size. Figure 5.12 (a)-(b) illustrate the performance of KFCM and LFCM respectively. It can be seen that both of the two algorithms fail to detect the correct patterns due to the large volume difference, where the small cluster centre shifts to the large cluster. The promising result is shown in Figure 5.12(c) by the DKFCM, where two distinct clusters have been correctly identified.

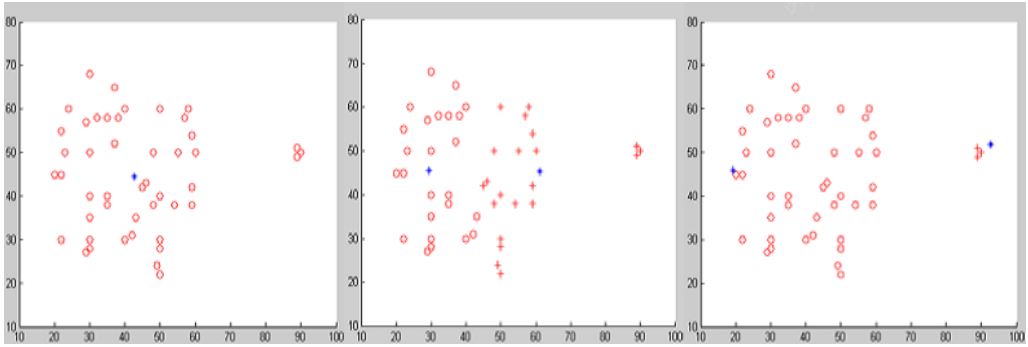


(a) KFCM (b) LFCM (c) DKFCM

Figure 6. 12 Clustering result for unbalance data

Experiment 3: Unbalance cluster with an outlier

An outlier (150, 0) is added to the unbalanced dataset. Results are shown in Figure 5.13 (a)-(c), where both LFCM and KFCM are heavily affected by the outlier and fail to find the correct clusters as shown in Figure 5.13 (a)-(b). However, the proposed algorithm can identify the correct patterns even when the resulting cluster centres shifts from the optimal position.

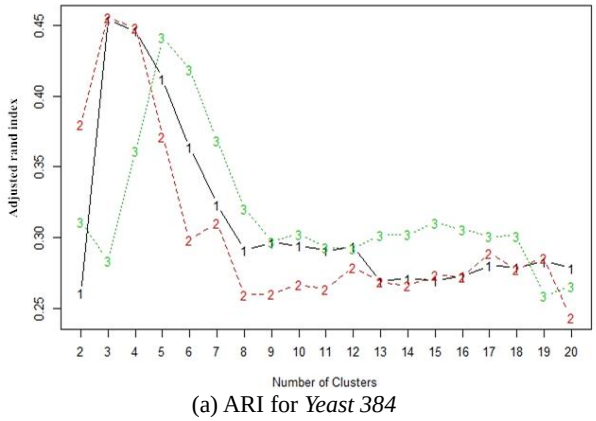


(a) KFCM (b) LFCM (c) DKFCM

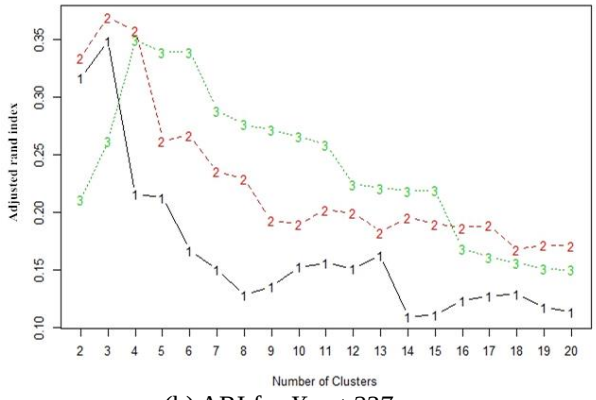
Figure 5. 13 Clustering result for unbalance data with noise

5.4.2 Gene expression data

Gene expression data *Yeast 384* and *Yeast 237* are selected to validate the proposed method. KFCM and LFCM are employed for comparison. ARI, biological relevance and FOM are used as assessment criteria for clustering performance on a range number of clusters (2-20 clusters). To remind, the parameter σ in KFCM is fixed 150; the *k* and *n* in DKFCM are chosen as 5 and 20 respectively. In term of ARI, the proposed algorithm, DKFCM, achieves the best performance on a range number of clusters (5-18) for *Yeast 384* (Figure 5.14 (a)), it peaks at the cluster number of 5 by indicating the optimal number of clusters is 5 that is agreed by Cho *et al.*(1999). However, KFCM and LFCM fail finding the optimal number of cluster. Figure 5.14 (b) shows that DKFCM attains the highest ARI value at the cluster number of 4 by indicating that the optimal clusters is four for *Yeast 237*, which is consist with the findings of Tavazoie *et al.*(1999).



(a) ARI for *Yeast 384*



(b) ARI for *Yeast 237*

Figure 5. 14 ARI for two gene expression data
(line 1,2,3 represents FCM, KFCM, and DKFCM respectively)

Figure 5.15 represents temporal behavior of *Yeast 384* identified by DKFCM. The periodic *cell cycle* pattern with different phases has been found. Genes in cluster 1 peak at the late G1 phase of the first cell cycle, and then peak again at the same phase of the second cell cycle. With time shift, genes in cluster 2 are activated in S phase, then followed by cluster 5 in early M phase. In contrast, genes in cluster 3 appear to be repressed across the whole first cell cycle period and then induced at the early G1 phase of the second cell cycle. For *Yeast 237*, Figure 5.16 shows that genes in cluster 2, 3 and 4 deviate from standard temporal pattern, no distinct patterns can be found in the three clusters.

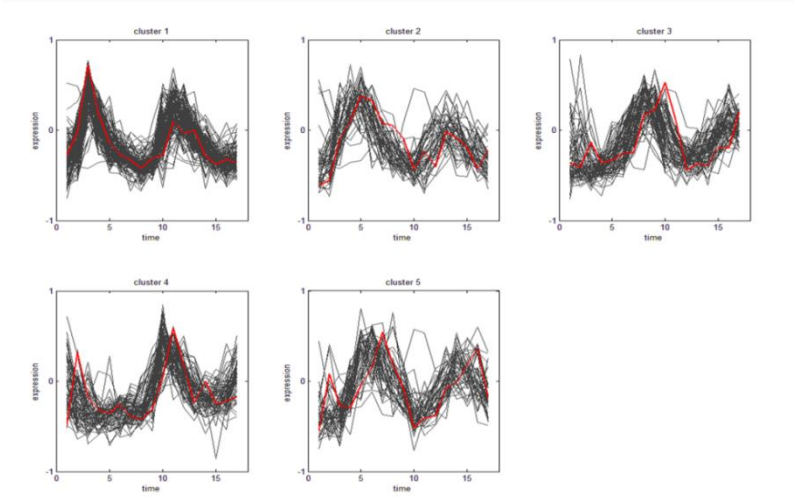


Figure 5. 15 Clustering result by DKFCM for *Yeast 384*
Red curves represent the respective clustering centre

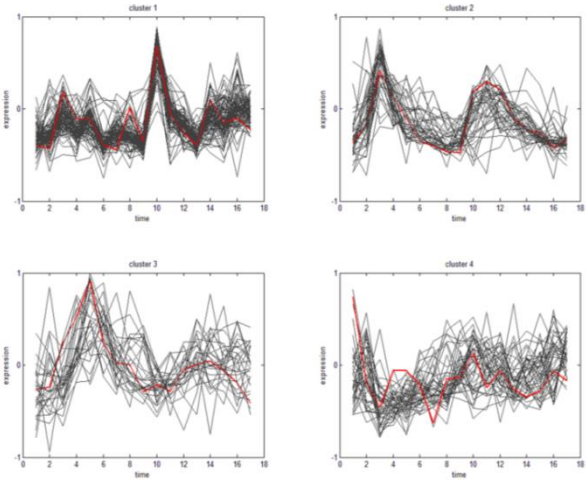
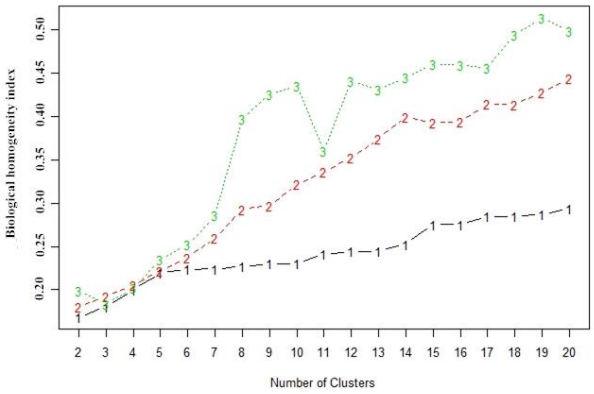
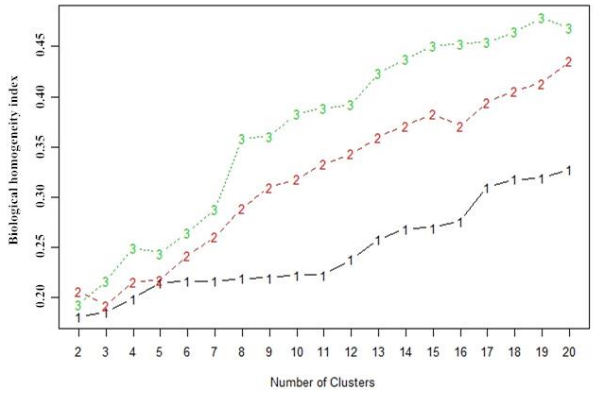


Figure 5. 16 Clustering result by DKFCM for *Yeast 237*
Red curves represent the respective clustering centre

BHI is used to assess the biological homogeneity of the produced cluster, which is to explicitly specify the functional clustering of the genes. The functional categorization of the genes in the two dataset were previously determined by Cho *et al*.(1999) and Tavazoie *et al.*(1999) respectively. Figure 5.17 (a) shows that DKFCM outperforms the other two methods for *Yeast 384* by indicating that the clusters produced by the DKFCM are of more biological significance. Similar result can be found in Figure 5.17 (b) for *Yeast 237*.



(a) BHI for *Yeast 384*



(b) BHI for *Yeast 237*

Figure 5. 17 BHI for for two gene expression data
(Line 1,2,3 represent FCM, LFCM and DKFCM respectively)

In order to test the functional enrichment of a group of genes in terms of three structured controlled ontologies, i.e., *biological processes*, *molecular functions* and *cellular components*. The functional enrichment of each GO category in each of the clusters is calculated by *p*-value (Tavazoie *et al*. 1999). The p-value is computed using a cumulative hyper geometric distribution. It measures the probability of finding the number of genes involved in a given GO term (i.e., function,

process, and component) within a cluster. From a given GO category, the probability *p* of getting *k* or more genes within a cluster of size *n*, is defined as:

$$p = 1 - \sum_{i=0}^{k-1} \frac{C_i^f C_{n-i}^{g-f}}{C_n^g} \tag{5.28}$$

where *f* and *g* denote the total number of genes within a category and within the genome respectively, and C_i^f is the binominal coefficient. The genes in a cluster are evaluated for the statistical significance by computing the p-value for each GO category. This signifies how well the genes in the cluster match with the different GO categories. *p*-value represents the probability of observing the number of genes from a specific GO functional category within each cluster. A low p-value indicates the genes belonging to the enriched functional categories are biologically significant in the corresponding clusters. To compute the p-value, FuncAssociate (Berriz *et al*., 2003) is used in this research, which is a web based tool by computing the hyper geometric functional enrichment score based on GO.

The enriched functional categories for each cluster obtained by the DKFCM for *Yeast 384* are listed in Table 5.2. The first 10 functional enrichment of GO category are extracted for each cluster. Of the 5 clusters obtained from the dataset, Cluster 1 contains genes involved in different pre-replicative processes. The highly enriched categories in Cluster 2 are cell cycle, DNA replication and DNA metabolic process. Cluster 3 contains genes involved in cytoskeleton process. The highest enriched category is microtubule cytoskeleton with p-value of 2.8e-13, which also contains an enriched category of ‘spindle’ that is related to the microtubule cytoskeleton. Cluster 4 contains the highly enriched biological process of cell cycle and cell division with a p-value of 4.2e-06 and 1.4e-05 respectively. In cluster 5, most of the functionally enriched categories are from biological process annotation with cell cycle. Similar results can be found in Table 5.3 for *Yeast 237* according to the MIPS (Mewes *et al*., 1998). From the Table 5.2 and 5.3, it can be concluded that DKFCM shows a good enrichment of functional categories and therefore project a good biological significance.

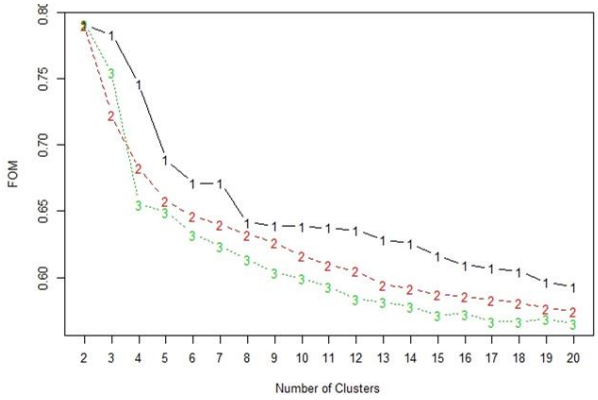
Table 5. 2 *p*-value of *Yeast 384*

	Rank	GO Attribute		P
Cluster 1	1	GO:0042555	MCM complex	5.4e-10
	2	GO:0005656	pre-replicative complex	3.9e-09
	3	GO:0006267	pre-replicative complex assembly	3.9e-09
	4	GO:0031261	DNA replication preinitiation complex	4e-08
	5	GO:0003688	DNA replication origin binding	1.7e-07
	6	GO:0005933	cellular bud	6e-07
	7	GO:0000084	S phase of mitotic cell cycle	7e-07
	8	GO:0006270	DNA replication initiation	1.3e-06
	9	GO:0043596	nuclear replication fork	1.6e-06
	10	GO:0051320	S phase	1.6e-06
Cluster 2	1	GO:0007049	cell cycle	7e-24
	2	GO:0006260	DNA replication	1.5e-22
	3	GO:0006259	DNA metabolic process	3.2e-22
	4	GO:0005694	chromosome	4.2e-21
	5	GO:0044427	chromosomal part	1.7e-20
	6	GO:0006261	DNA-dependent DNA replication	4.5e-20
	7	GO:0022402	cell cycle process	5.2e-18
	8	GO:0005657	replication fork	4.9e-17
	9	GO:0006281	DNA repair	7.1e-16
	10	GO:0022403	cell cycle phase	9.6e-16
Cluster 3	1	GO:0015630	microtubule cytoskeleton	2.8e-13
	2	GO:0005819	spindle	2.9e-12
	3	GO:0005874	microtubule	1.4e-11
	4	GO:0044430	cytoskeletal part	1.9e-11
	5	GO:0000278	mitotic cell cycle	9.4e-11
	6	GO:0007017	microtubule-based process	1.2e-10
	7	GO:0005856	cytoskeleton	1.6e-10
	8	GO:0007059	chromosome segregation	1.8e-09
	9	GO:0000226	microtubule cytoskeleton organization and biogen-esis	2.3e-09
	10	GO:0007020	microtubule nucleation	2.3e-09
Cluster 4	1	GO:0007049	cell cycle	4.2e-06
	2	GO:0051301	cell division	1.4e-05
	3	GO:0005694	chromosome	3.3e-05
	4	GO:0005935	cellular bud neck	4.9e-05
	5	GO:0000776	kinetochore	5e-05
	6	GO:0044427	chromosomal part	5.4e-05
	7	GO:0000793	condensed chromosome	5.5e-05
	8	GO:0000228	nuclear chromosome	5.6e-05
	9	GO:0007059	chromosome segregation	6.3e-05
	10	GO:0000268	nuclear chromosome	6.3e-05
Cluster 5	1	GO:0007049	cell cycle	7.6e-11
	2	GO:0022402	cell cycle process	2.2e-08
	3	GO:0022403	cell cycle phase	2.5e-08
	4	GO:0000278	mitotic cell cycle	2.8e-08
	5	GO:0005933	cellular bud	8.6e-08
	6	GO:0051301	cell division	1.4e-07
	7	GO:0030427	site of polarized growth	8.8e-07
	8	GO:0005935	cellular bud neck	1.1e-06
	9	GO:0007067	mitosis	1.9e-06
	10	GO:0000087	M phase of mitotic cell cycle	2.2e-06

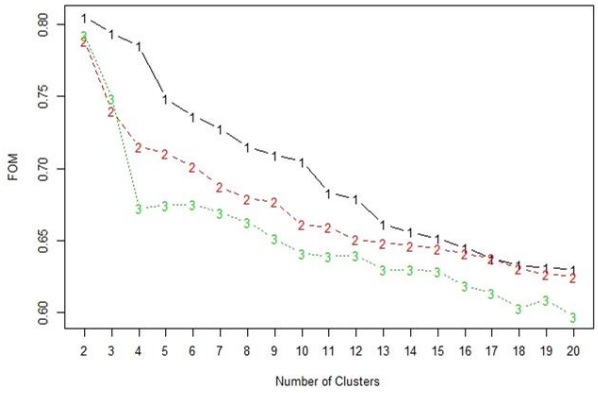
Table 5.3 *p*-value of *Yeast 237*

	Rank	GO Attribute		P
Cluster 1	1	GO:0006260	DNA replication	1.5e-40
	2	GO:0006261	DNA-dependent DNA replication	6e-37
	3	GO:0005657	replication fork	1.5e-34
	4	GO:0006271	DNA strand elongation during DNA replication	9.2e-29
	5	GO:0022616	DNA strand elongation	9.2e-29
	6	GO:0006259	DNA metabolic process	9.1e-28
	7	GO:0043596	nuclear replication fork	2.4e-27
	8	GO:0005694	chromosome	9.7e-26
	9	GO:0044427	chromosomal part	1.4e-25
	10	GO:0003677	DNA binding	5e-20
Cluster 2	1	GO:0007020	microtubule nucleation	2.5e-27
	2	GO:0005200	structural constituent of cytoskeleton	4.1e-27
	3	GO:0007017	microtubule-based process	6.3e-27
	4	GO:0000226	microtubule cytoskeleton organization and biogenesis	6.7e-26
	5	GO:0015630	microtubule cytoskeleton	1.4e-24
	6	GO:0005819	spindle	6.9e-24
	7	GO:0005815	microtubule organizing centre	1.7e-21
	8	GO:0005816	spindle pole body	1.7e-21
	9	GO:0007010	cytoskeleton organization and biogenesis	2.2e-21
	10	GO:0000922	spindle pole	5.3e-21
Cluster 3	1	GO:0006807	nitrogen compound metabolic process	1.9e-16
	2	GO:0009308	amine metabolic process	2.4e-14
	3	GO:0000103	sulfate assimilation	2.3e-12
	4	GO:0006791	sulfur utilization	2.3e-12
	5	GO:0006519	amino acid and derivative metabolic process	7.6e-12
	6	GO:0019344	cysteine biosynthetic process	1e-11
	7	GO:0006534	cysteine metabolic process	3.2e-11
	8	GO:0000096	sulfur amino acid metabolic process	5.1e-11
	9	GO:0006520	amino acid metabolic process	5.9e-11
	10	GO:0009086	methionine biosynthetic process	6.7e-11
Cluster 4	1	GO:0003735	structural constituent of ribosome	7.1e-121
	2	GO:0033279	ribosomal subunit	7.1e-116
	3	GO:0005840	ribosome	3.9e-112
	4	GO:0005198	structural molecule activity	1.5e-110
	5	GO:0022626	cytosolic ribosome	8.3e-108
	6	GO:0030529	ribonucleoprotein complex	1.2e-106
	7	GO:0044445	cytosolic part	5.1e-98
	8	GO:0006412	translation	1.7e-86
	9	GO:0009059	macromolecule biosynthetic process	7.4e-86
	10	GO:0043228	non-membrane-bounded organelle	9.5e-75

In term of stability, FOM is used to assess the prediction ability of the three algorithms. The smaller the values, the better of the algorithm is, by indicating that the probability that the clusters are not formed by chance. It can be seen from Figure 5.18 (a) and (b) that the proposed method has better performance on the three dataset.



(a) FOM for *Yeast 384*



(b) FOM for *Yeast 237*

Figure 5. 18 FOM for two gene expression data
(Line 1, 2, 3 represent FCM, LFCM and DKFCM respectively)

Although the results of two gene expression data suggested $k=5; n=30$ can achieve better performance than the other two methods, this does not indicate that this setting has best performance on all gene expression data. In order to investigate the influence of number of training sample *n* on the clustering result, ARI is used to examine the performance by fixing *k* at 5 for *Yeast 384*. It can be seen form the Figure 5.19 that with the increasing number of training samples, ARI does not have significant changes by indicating that the proposed method is not sensitive to the number of training samples as long as a few sample distribution information can be obtained, because the optimization process can obtain the 'best' value in the clustering process.

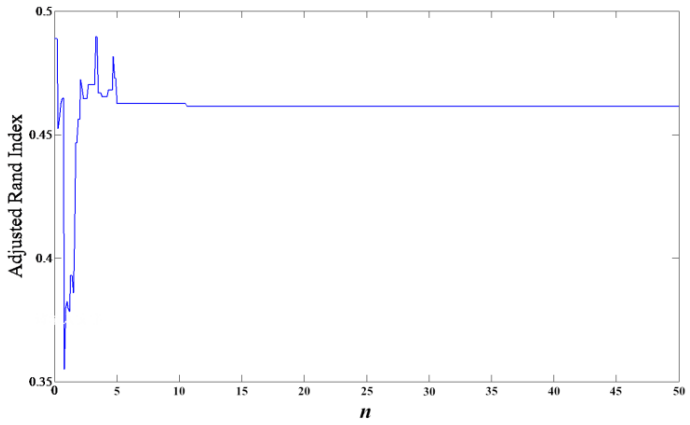


Figure 5. 19 ARI vs the number of training samples

In order to examine the influence of number of *k* minimum distances on the clustering result, ARI is used to examine the performance by fixing *n* at 30 for *Yeast 384*. It can be seen form the Figure 5.20 that the ARI does not have obvious oscillation with the increasing number of minimum distances, which indicates that the number of *k* minimum distances has not much influence on the clustering result.

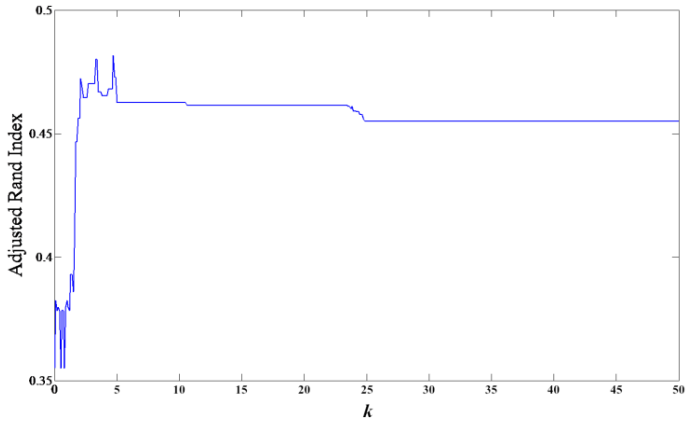


Figure 5. 20 ARI vs the number of *k* minimum distances

5.5 Conclusion

This chapter presents a new fuzzy clustering approach for gene expression data analysis, by which the initialization sensitivity is resolved for FCM. Moreover, a new weight parameter is added to the objective function by concentrating on the samples in high density area. Furthermore, this approach incorporates a parameter

optimization process, which can automatically find the optimal values for the clustering process. Compared to the conventional fuzzy clustering methods, the proposed approach can achieve better performance on artificial data and real gene expression data. More importantly, the produced clusters show significant agreement with the biological interpretation. However, the computational cost of the proposed method is higher than the other two methods due to the optimization process.

Chapter 6 Fuzzy clustering of time series gene expression data with Cubic spline

6.1 Introduction

Generally, there are two categories of gene expression data: static data and time series data. In static gene expression data, a snapshot of the expression of genes in different samples is measured (Tsai *et al.*, 2006), while in time series expression experiments, a temporal process is measured (Tang and Muller, 2009). Another important difference between the two types of data is that while static data from a sample population is assumed to be independent identically distributed, while time series data exhibit a strong autocorrelation between successive points. Most previous works analyzing time series expression used methods developed originally for static data by neglecting the time series characteristics (Belacel *et al*., 2006). Recently, several new algorithms specifically targeting time series expression data were presented. A popular procedure in time-series analysis is smoothing the data, which removes random variation and shows trends and cyclic components (Song *et al*., 2007). Bar-Joseph *et al*.(2004) used statistical spline estimation to represent time-series gene expression profiles, however, the method require data that has been sampled at a sufficiently high rate. In addition, cubic splines are used for smoothing gene expression time-series, however no appropriate similarity metric is adopted (Bar-Joseph *et al*., 2003). Later, Luan and

Li (2003) proposed a mixed-effects model using B-splines for gene expression time-series. However, the number and locations of the knots for the B-splines corresponding to the mean function and the random effects have to be specified.

6.2 Time-series gene expression data

A time-series is often defined as a series of values of variables taken in successive period of time (Tang and Muller, 2009). The length between time points can vary or be constant. The main goal in the statistical analysis of time-series is to identify the nature of the phenomenon represented by the series of observations. Selecting a suitable mathematical model is the first step in the analysis of a time-series. After choosing the model, it is possible to estimate parameters and check for the goodness of fit to the data. The fitted model can then be possibly used to understand the mechanism generating the series or to forecast. The selection of the appropriate technique will depend on the application and the user's preference (Tang and Muller, 2009).

For gene expression analysis, researchers are interested in the general trend of the gene expression. The actual gene expression values may not have been observed for two reasons. First, errors may occur in the experimental process that leads to corruption or absence of some expression measurements. Second, it is important to estimate expression values at time points different from those originally sampled. In either case, the nature of microarray data makes straightforward interpolation difficult. Data are often noisy and there are few replicates. Thus, simple techniques such as interpolation of individual gene can lead to poor estimation. If the data contain a trend, a curve can be fitted to the data and then the residuals from that fit can be modelled (Luan and Li, 2003). When the variance is non-constant, it might be stabilized by taking the square root of the series. A very popular procedure in time-series analysis is smoothing the data, which removes random variation and shows trends and cyclic components. The most common technique is the moving average smoothing which replaces each element of the

series by either the simple or weighted average of *n* surrounding elements, where *n* is the width of the smoothing window. This method will filter out the noise and convert the data into a smooth curve that is relatively unbiased by noises (Luan and Li, 2003).

6.3 Method

Time series is a special kind of microarray data. However, conventional clustering methods rarely consider the characteristics of the time series. In this work, an integrated fuzzy clustering approach (FCMS) is proposed which uses spline to smooth expression profiles. By introducing a new geometry term of radius of curvature, it can capture the general trend information between curves. Results demonstrate that the new method has substantial advantages over FCM for time-series expression data.

6.3.1 Cubic spline

A spline curve is a sequence of curve segments that are connected together to form a single continuous curve. Given *n* data points, an *n*−1 degree polynomial has exactly enough coefficients to fit the data. For example, given 5 data points, one 4*th* degree polynomial fits the data exactly. The basic idea of the cubic spline is that it represents the function by a different cubic function on each interval between data points. Specifically, a cubic spline is a piecewise third-order polynomial which is smooth in the first derivative and continuous in the second derivative. For example, given *n* data points $\{(x_i, y_i) | i = 1, 2.....n\}$, the spline *S(x)* is,

$$S(x) = \begin{cases} C_1(x) & x_0 \le x \le x_1 \\ & \vdots \\ C_i(x) & x_{i-1} \le x \le x_i \\ & \vdots \\ C_n(x) & x_{n-1} \le x \le x_n \end{cases} \qquad (6.1)$$

where each $C_i(x)$ is a cubic function on the interval,

$$C_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \tag{6.2}$$

To determine the spline, it is needed to determine the coefficients, a_i, b_i, c_i, and d_i for each interval $[x_{i-1}, x_i]$. Since there are *n* intervals, there are *4n* coefficients to determine. First the spline needs to satisfy the following equation:

$$\begin{cases} C_i(x_{i-1}) = y_{i-1} \\ C_i(x_i) = y_i \end{cases} \tag{6.3}$$

In order to make *S(x)* as smooth as possible, it is required:

$$\begin{cases} C_i'(x_i) = C_{i+1}'(x_i) \\ C_i''(x_i) = C_{i+1}''(x_i) \end{cases} \tag{6.4}$$

where C' and C'' are the first and second derivative respectively.

There are 2(*n*-1) of these conditions. Since each C_i is cubic, there are a total of *4n* coefficients in the formula for *S(x).* Using equation (6.4) and (6.5) for each interval, *4n-2* equations have been known for the spline. Two additional equations are needed to determine all the coefficients, and it requires the second derivatives at its boundaries to be zero,

$$\begin{cases} C_1''(x_0) = C_n''(x_i) = 0 \\ C_1'(x_0) = C_n'(x_i) = 0 \end{cases} \tag{6.5}$$

The coefficient can be computed by,

$$\begin{cases} a = \dfrac{x_{i+1} - x}{x_{i+1} - x_i} \\ b = \dfrac{x - x_i}{x_{i+1} - x_i} \\ c = \frac{1}{6}(a^3 - a)(x_{i+1} - x_i)^2 \\ d = \frac{1}{6}(b^3 - b)(x_{i+1} - x_i)^2 \end{cases} \tag{6.6}$$

6.3.2 Smoothing gene expression with cubic spline

There is a large component of noise in microarray data due to biological and experimental factors. The activity of genes can show large variations under minor changes of the experimental conditions. Numerous steps in the experimental procedure contribute to additional noise and bias. A usual procedure to reduce the noise in microarray data is setting a threshold for a minimum variance of the abundance of a gene. Genes below this threshold are excluded from further analysis. However, the exact value of the threshold remains arbitrary due to the lack of an established error model and the use of filtering as the preprocessing step may exclude interesting genes from further analysis (Andreas and Francis, 2005). Smoothing techniques are data transformations can decrease the impact of individual observations on the overall pattern or "shape" of the data. Smoothing can help to remove "spikes" from the data in order to focus on the signal and can be useful for comparing noisy data. Changes in gene expression levels happen gradually, and the smoothed profiles may more closely resemble what occurs in nature (Luan and Li, 2003).

Interpolating cubic splines to time course expression data (*i.e.*, splines are forced to pass through all the sampled data points) may inadvertently attribute significance to measurements dominated by noise due to over-fitting. To infer meaningful gene expression trends over time, it is expected to fit natural cubic splines to expression data in a smooth fashion. Figure 6.1 shows that a single gene expression fitted with three techniques: curve fitting with an arbitrary function, a least squares fit (the straight line), and curve fitting with a smoothing spline. It can be seen that curve fitting with a smoothing spline is the best fitting technique by identifying the general trend of gene expression.

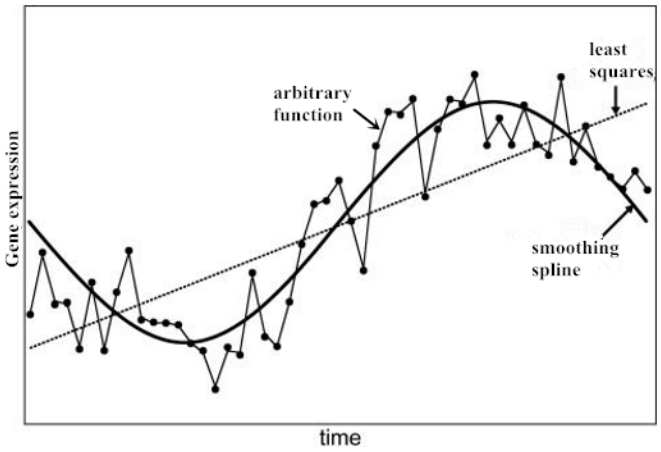


Figure 6. 1 Curve fitting

Define a single gene expression,

$$y_i^j = f(t_j) + \varepsilon_{ij} \tag{6.7}$$

where y_i^j denotes the observation for the *i*th gene at time t_j, (t_0,...,t_{k-1}) is called knot vector. f is a continuous and differentiable function, and ε_{ij} are independent and identically distributed random variables satisfying classical assumptions:

$$E(\varepsilon_{ij}) = 0, \quad Var(\varepsilon_{ij}) = \sigma^2 \tag{6.8}$$

where *E* is the expectation, *Var* is the variance.

A practice for curve fitting is to minimize the residual sum of squares (RSS):

$$\text{RSS} = \sum_{i=0}^{n} (y_i - f(t))^2 \tag{6.9}$$

In order to make the curve more flexible, a smoothness condition is imposed. Here, a standard constraint is adopted (Dejean *et al*., 2007),

$$\int |f''(t)|^2 dt < \eta \tag{6.10}$$

where η is a minimal constant.

A cubic smoothing spline $f(t_j)$ is sought for each gene, which shall be both reasonably smooth and also reasonably close to its observation value y_i^j. As a stand-

ard practice for spline smoothing, a cubic smoothing spline can be found by minimizing the following combined function (Dejean et al., 2007),

$$L=(1-\lambda)\sum_{i=0}^{n}(y_i - f(t))^2 + \lambda\int|f''(t)|^2 dt \qquad (6.11)$$

in which, the first term RSS quantifies the closeness between spline curve and gene expression profile, and the second term is the integrated squared second derivative, which quantifies the smoothness of the fitted spline. The smoothing parameter, $\lambda \in [0,1]$, is used to control the trade-off between the two criteria for closeness and smoothness. Setting λ =1 gives rise to the straight line from an ordinary linear least-squares regression. In contrary, setting λ =0 leads to a cubic interpolating spline, which passes through every data point. The influence of λ is illustrated with the gene expression data *Cyp4a10* extracted from *Yeast cell cycle* in Figure 6.2. Given different λ value, smoothed profiles exhibit various fluctuations along the time axis.

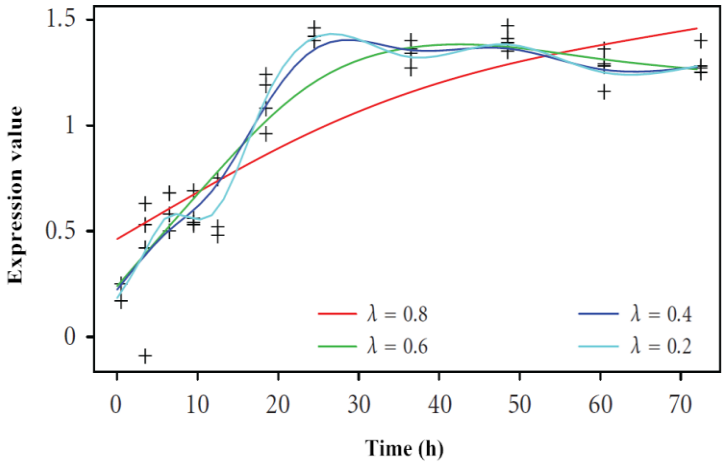


Figure 6. 2 Smoothed curves obtained for the gene *Cyp4a10* with λ= 0.2, 0.4, 0.6, and 0.8

6.3.3 Similarity

As discussed in Chapter 4, proximity measure has a key influence to the clustering results. Smoothing gene expression profiles transform the discrete expression values into continuous curves. However, the conventional choice of proximity

measures as Euclidean distance or Pearson correlation can only deal with discrete variables. In order to identify the up-and-down trend for genes with different conditions, a feature should be selected to describe the curve shape. Radius of curvature is a geometrical term describing the feature of the curve (Kuragano and Kasono, 2008), which is employed to compute the similarity between two splines.

The distance from the centre of a circle to a point on the circle is the radius. For curves, the radius of curvature at a given point is defined by the radius of a circle that mathematically best fits the curve at that point (Kuragano and Kasono, 2008). Figure7.3 shows that radius of curvature at two given points.

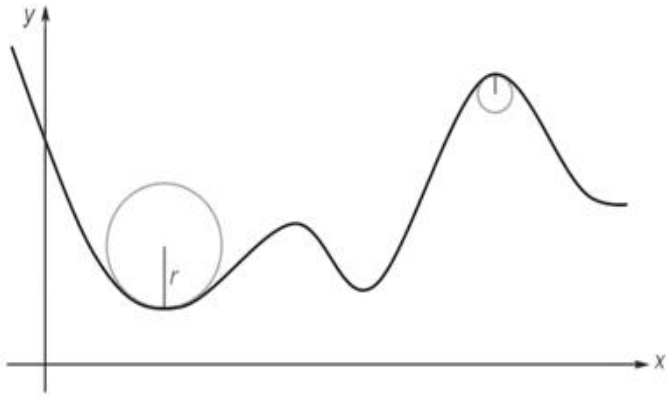


Figure 6. 3 Radius of curvature of a curve

The radius of curvature can be computed by (Kuragano and Kasono, 2008),

$$R = \left| \frac{(1+y'^2)^{3/2}}{y''} \right| \tag{6.12}$$

where y is a spline and $y' = \frac{dy}{dx}$, $y'' = \frac{d^2y}{dx^2}$

Similarity between curve A and curve B is evaluated by normalizing the dot product of the radius of curvature, which could be sampled at the knot vector.

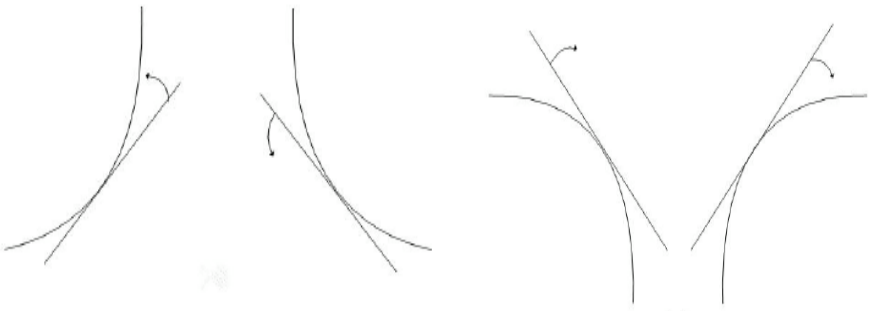
$$S = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|a||b|} \tag{6.13}$$

By definition, the curvature of a plane curve is nonnegative, which however limits its application for gene expression similarity metric. Figure 6.4(a) shows an example that two curves have the same radius of curvature. However, the two curves

have totally different expression trend. Kuragano and Kasono (2008) proposed to ascribe a sign to the curve. The choice of the sign is usually connected with the tangent rotation of the curve is positive when its tangent rotates counter-clockwise; the curvature of the curve is negative when its tangent rotates clockwise. This strategy cannot capture gene trend exactly. For example, in Figure 6.4 (a) the two curves with the same radius of curvature and same rotation (counter-clockwise), but have different trends. Figure 6.4 (b) shows two curves with the same radius of curvature and same rotation (clockwise direction), but have different trends.

In order to overcome this limitation, a modification of the radius of curvature is proposed by adding a sign function of the first derivative of the curve, which describes the trend of the curve.

$$R = \mathrm{sgn}(y') \frac{(1+y'^2)^{3/2}}{y''} \tag{6.14}$$



(a) (b)

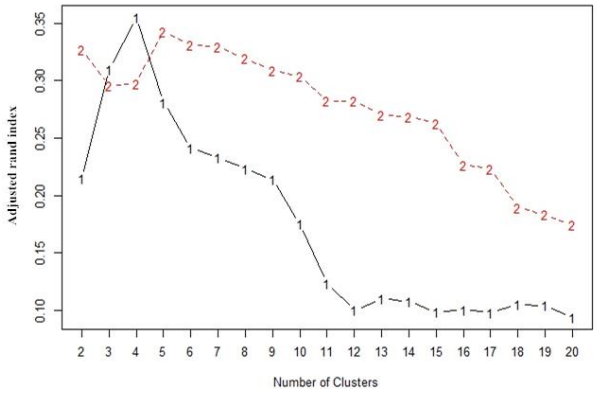
Figure 6. 4 Radius of curvature with different trend

The following algorithm is proposed,

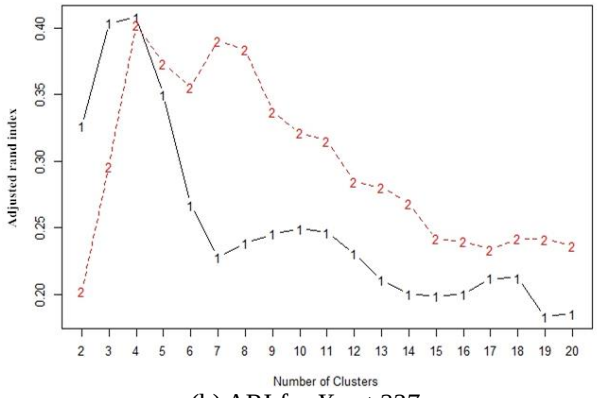
Fuzzy clustering with cubic spline (FCMS). Given *N* data $X = \{x_j\}_j^N$ and the number of cluster *C*, output a membership matrix $U = \{u_{ij}\}$
1: Using spline to model gene expression data according to **Equation 6.11**. **2**: Sampling of the spline and calculate the radius of curvature by **Equation 6.14**. **3**: Compute the spline similarity according to **Equation 6.13**. **4**: Run FCM algorithm based on the new similarity metric **Until** (prototype parameters stabilize). **5**: Evaluate the result by validity measures.

6.4 Experiments and results

Three gene expression datasets are chosen to validate the proposed algorithm: *Yeast 384*, *Yeast 237*, and *Yeast 2945*. Before experiments, the fuzziness exponent is empirically set to 1.34 1.34 and 1.68 for *Yeast 384*, *Yeast 237*, and *Yeast 2945* respectively. λ=0.8 is chosen as the smoothing parameter. In order to evaluate the performance of the proposed algorithm, ARI, BHI and FOM are used to assess the quality of the produced clusters. Figure 6.5 (a) and (b) shows the result of ARI of the two algorithms for *Yeast 384* and *Yeast 237* respectively (For *Yeast 2945*, there is no external labels for this data, ARI cannot be used). It can be seen that FCMS achieves better performance than FCM because it can capture general trend by minimizing the random variations and influence of noise, therefore, the produced clusters are more accurate than the FCM.



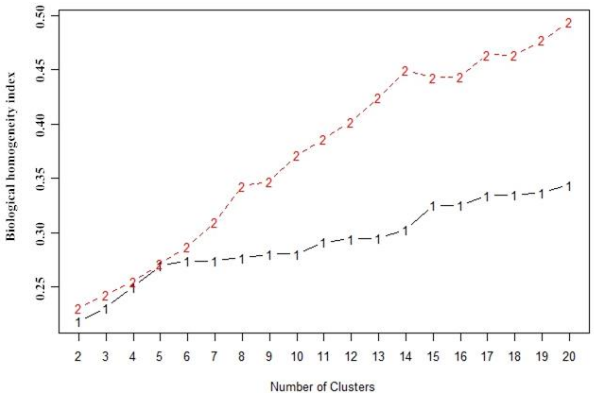
(a) ARI for *Yeast 384*



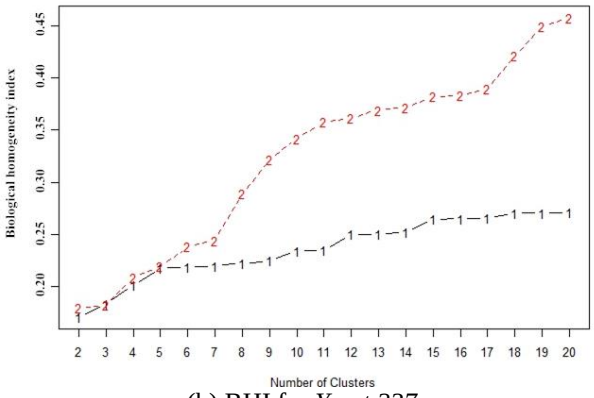
(b) ARI for *Yeast 237*

Figure 6. 5 ARI for two sets of gene expression data
(Line 1 and 2 represents the FCM and FCMS respectively)

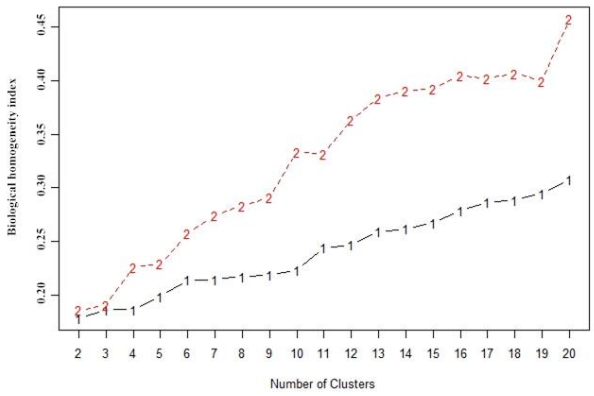
In term of BHI, Figure 6.6 shows that FCMS achieves better biological significance for the three datasets. The FCMS achieves high biological homogeneity on all number of clusters (2, 20) for *Yeast 384* (Figure 6.6 (a)). Similar results can be found in Figure 6.6 (b) and (c) for *Yeast* 237 and *Yeas*t 2945 respectively by indicating the genes placed in the same statistical cluster belong to the same functional classes.



(a) BHI for *Yeast 384*



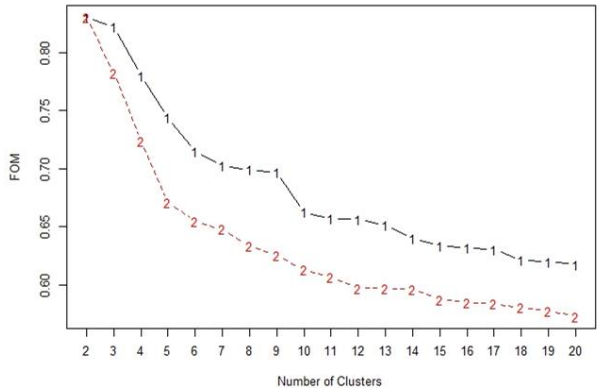
(b) BHI for *Yeast 237*



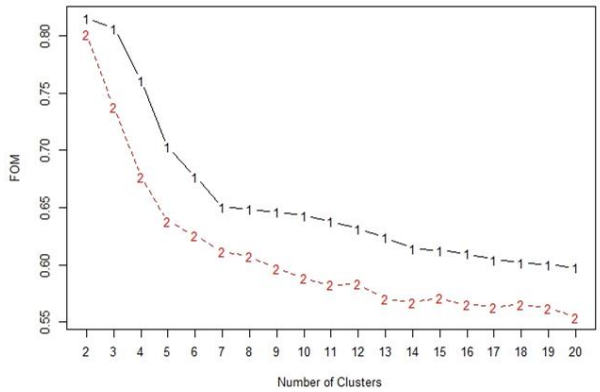
(c) BHI for *Yeast 2945*

Figure 6. 6 BHI for three sets of gene expression data
(Line 1 and 2 represents the FCM and FCMS respectively)

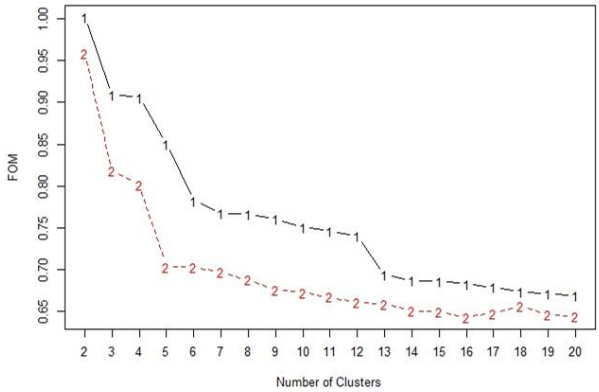
In terms of stability, Figure 6.7 shows that the FCMS has more stable performance than FCM for the three datasets, which indicates that FCMS dependence on the left-out feature is small and two cluster results reveal a similar structure.



(a) FOM for *Yeast 384*



(b) FOM for *Yeast 237*

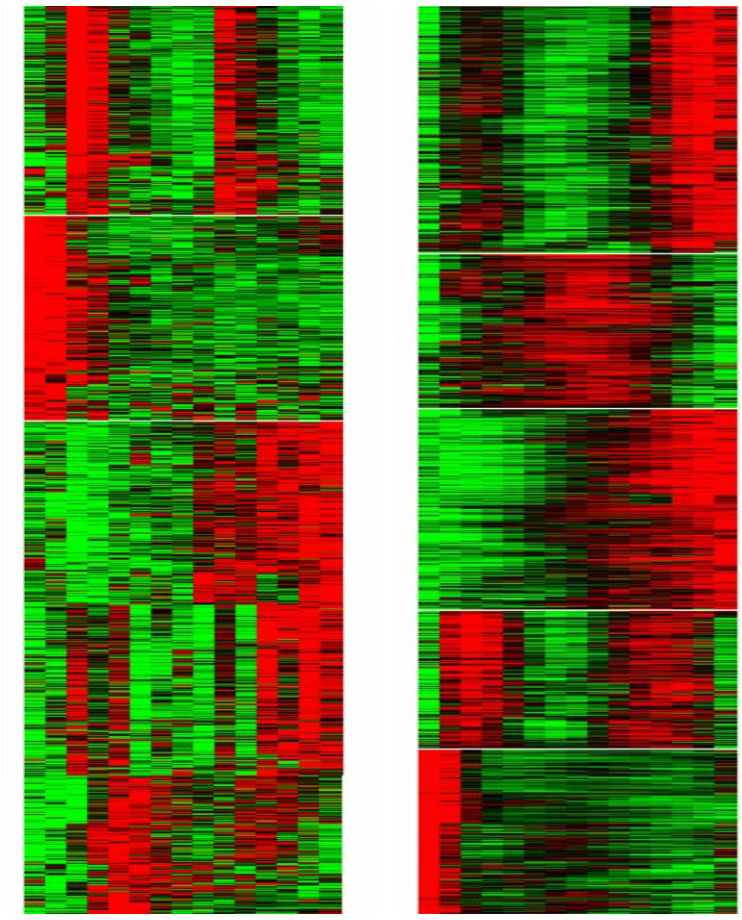


(c) FOM for *Yeast 2945*

Figure 6. 7 FOM for three sets of gene expression data
(Line 1 and 2 represents the FCM and FCMS respectively)

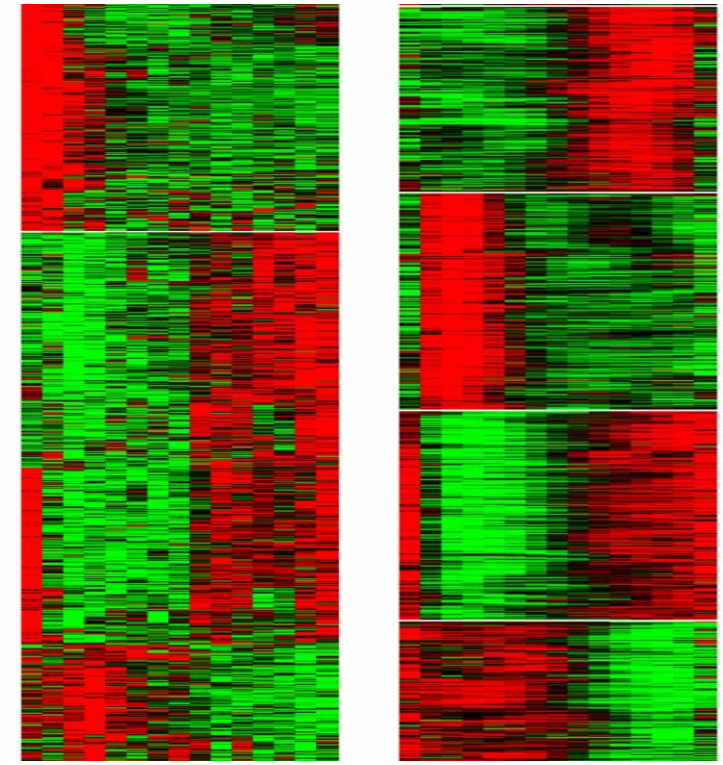
Heatmap is used to graphically represent multidimensional gene expression data subjected to clustering algorithms and gives another way to assess the quality of

clusters (Tavazoie *et al*., 1999). The colours represent the values of each gene at each time point. The lower the gene expression value is, the greener the color is. The higher the gene expression value is, the redder the colour is. In this study, Gene Expression Data Analysis Studio (GEDAS), a clustering software designed by Fu (2007), is used to produce the Heatmap. Figure 6.8(b) shows the cluster produced by FCMS has better quality (evaluated by inter separation and intra homogeneity) than that produced by FCM (Figure 6.8(a)) for *Yeast 384*. Similar results are demonstrated in Figure 6.9 and Figure 6.10 for *Yeast 237* and *Yeast 2945* respectively. The identified structure is distinct and precise by supplying the researchers and biologists an efficient way to understanding the patterns contained in the data.



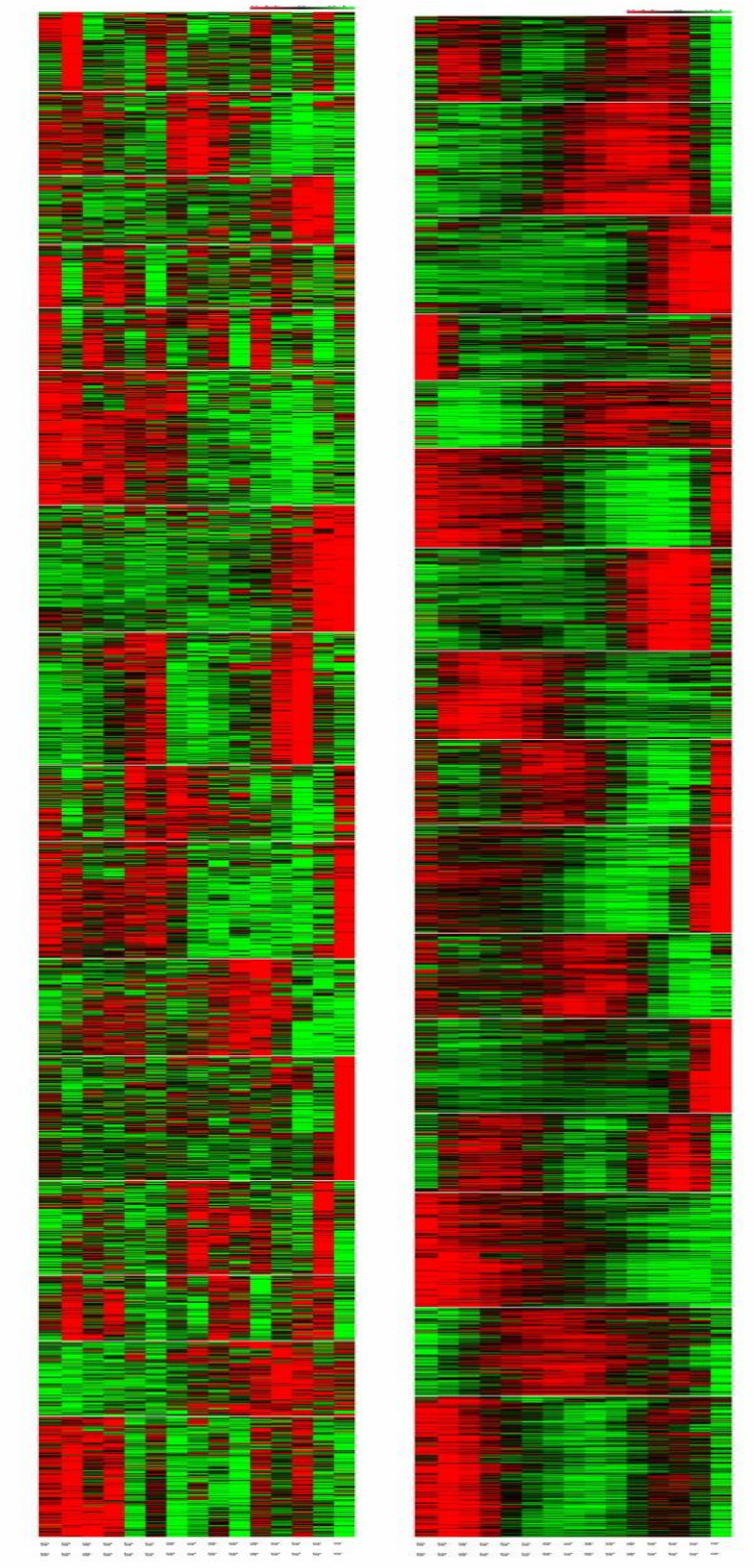
(a) Cluster structure by FCM (b) Cluster structure by FCMS

Figure 6. 8 Heatmap of cluster structure for *Yeast 384*



(a) Cluster structure by FCM (b) Cluster structure by FCMS

Figure 6. 9 Heatmap of cluster structure for *Yeast 237*



(a) Cluster structure by FCM (b) Cluster structure by FCMS

Figure 6. 10 Heatmap of cluster structure for *Yeast 2945*

In above experiments, the smoothing parameter is set as 0.8, however, this setting may not be suitable for all gene expression data. In order to examine its impact on the clustering results, ARI is used to assess the parameter influence. By setting the number of cluster as 5 and 4 for *Yeast 384* and *Yeast 237* respectively, Figure 6.11 shows that ARI fluctuates with the choice of λ for *Yeast 384*. Small smoothing value makes the spline not immune to the random variation. Increasing smoothing value improves the clustering result. For $\lambda = 0.8$, FCM achieves the best performance. However, large smoothing value makes the spline too smooth to capture the detail trend information and leads poor clustering result.

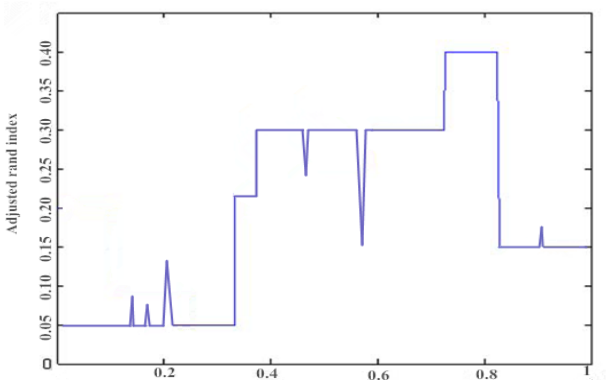


Figure 6. 11 ARI vs Smoothing parameter for *Yeast 384*

Figure 6.12 shows that ARI has an uptrend with the increase of λ for *Yeast 237*. if $\lambda \in [0.83, 0.94]$, FCMS achieves the best performance. Therefore, smoothing parameter varies according to the attributes of expression data (such as gene expression variance, noise volume. etc). The adaptive selection of smoothing parameter need to be further investigated.

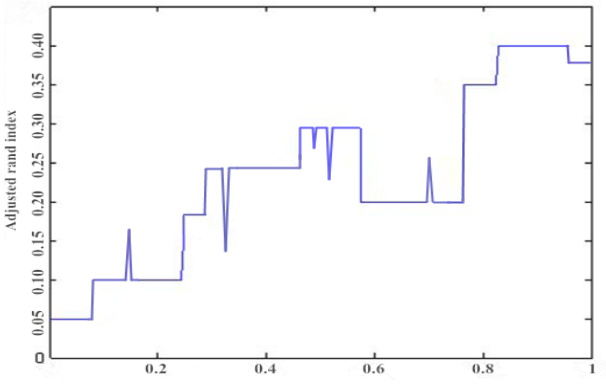


Figure 6. 12 ARI vs Smoothing parameter for *Yeast 237*

6.5 Conclusion

Conventional partition clustering methods are frequently used for gene expression analysis without consideration of the noise and random variations in expression that do not fit into any global pattern. In this chapter, an integrated fuzzy clustering approach, FCMS, is proposed by using spline to fit time-series expression data, by which noise and random variation can be filtered. In addition, a new geometrical parameter, radius of curvature, is introduced to capture the trend information between splines. Results demonstrate that the proposed method has substantial advantages over FCM for time-series gene expression data.

Chapter 7 Conclusion and future research

7.1 Conclusion

With the development of DNA sequencing techniques and microarray technology, genomic research has achieved a great success. A wealth of biological data has been extracted from microarrays. Analysis of these data on the molecular level is revolutionary in medicine because they are highly informative (Eisen et al., 1998). Innovative models are needed instead of straightforward adaptations of existing methodologies. Clustering techniques have proven to be helpful to understand gene function, gene regulation, and cellular processes (Yeung *et al*., 2001a; Covell *et al*., 2003; Belacel *et al.*, 2006; Page and Coulibaly, 2008; Iam-On and Boongoen, 2012). Genes with similar expression patterns (co-expressed genes) can be clustered together with similar cellular functions (Gasch and Eisen, 2002). This approach may help researchers to understand of many genes for which information has not been previously available. Furthermore, co-expressed genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates co-regulation (Spellman *et al.*, 1998; Tsai *et al*., 2006). In this research, clustering for gene expression data analysis is investigated from theory to applications. Several novel clustering techniques have been proposed for enhancing the cluster quality and biological significance.

Microarray data contains plenty of uncertain and imprecise information. FCM is an efficient model to deal with this type of data (Audrey and Michael, 2002). FCM unravels complex regulation mechanism of gene with consideration that one gene can be assigned to more than one cluster. FCM captures genes involved in multiple transcriptional programs and biological processes. Even though FCM had been proposed previously for gene expression data analysis, it has been hindered by several limitations. First, FCM is sensitive to initialization, and different initializations will result in different partitions (Graves and Pedrycz, 2010). Moreover, there is a tendency to equally partition the dataset (Graves and Pedrycz, 2010), which leads the clusters lack of biological interpretations. Furthermore, FCM is based on the Euclidean distance in the observation space that is only effective in finding spherical clusters (Huang *et al.*, 2012). In the last years, kernel methods are successfully applied in machine learning and SVM (Genton *et al*., 2002). Kernel FCM (KFCM) has been proposed to perform clustering in a typically higher-dimensional feature space spanned by embedding maps and corresponding kernel functions (Girolami, 2002). By KFCM, the produced clusters avoid the limitations of FCM, such as equally partitions and spherical clusters. In this thesis, KFCM is used for clustering gene expression data. In order to evaluate the cluster results, three validation methods are introduced to assess the quality of the produced clusters. For internal validation, Silhouette index is used to assess the tightness and separation of clusters using intrinsic information of the dataset. For external validation, adjusted rand index is employed to evaluate the agreement between clustering result and external labels. For biological validation, biological homogeneity index is used to examine the homogeneity of the clusters biologically. Results on artificial data and real gene expression data show that FCM and KFCM yield better performance than the other methods by producing quality clusters.

FCM is a useful mathematical model to identify the underlying patterns in gene expression data. However, FCM treats samples equally that cannot differentiate noise and meaningful data. In this thesis, motivated by the preservation of local

structure, a local weighted FCM (LFCM) is proposed which assigns weights to the samples in the neighborhood of the cluster centre. LFCM gains a proper weight by describing the neighborhood structure. Experiments show that the proposed method is good at finding quality clusters and robust to noise.

In order to make the clustering more reliable, a new fuzzy clustering approach is proposed based on FCM by utilizing kernel distance to measure the genes similarity. It not only finds the nonlinear relationship between genes, but also identifies arbitrary shape of clusters. FCM is sensitive to the initialization, in order to avoid the algorithm trapping into local minimum, an initialization method is proposed based on Parzen density estimation. In addition, the objective function is modified by adding a new weighted parameter, which accentuates the samples in high density area and reduce the influence of noise in periphery. Furthermore, an optimization method is presented which can automatically find the optimal values for the parameters in the clustering process. Experiments on synthetic data and real gene expression data show that the proposed method substantially outperforms conventional models in term of stability and cluster quality. Moreover, the produced clusters show significant agreement with the biological interpretation.

Time series microarray is a special category of gene expression data, which is characterized by time dependency. Most previous works analyzing this type of data are developed originally for static data by neglecting the time series characteristics (Belacel *et al*., 2006). In this research, the time series characteristic is investigated. An integrate FCMS approach is proposed which is consisted of two integrated steps. Firstly, gene expression data is modeled by cubic spline. By tuning the smoothing parameter, it can be smoothed with statistical consideration. Secondly, FCM is conducted based on the radius of curvature of the smooth spline. Results demonstrate that the proposed method has substantial advantages over FCM for time-series gene expression data.

7.2 Limitation and Future research

In this research, the problems defined in Chapter one are partly addressed. Given various available clustering algorithms, the selection of the most appropriate algorithm to a given gene expression dataset becomes a major problem faced by biologists. According to Yeung *et al.* (2001a), there are no omnipotent algorithms for every aspect. Researchers typically select a few candidate algorithms and compare the clustering results. Moreover, although various approaches have been developed to assess the quality or reliability of the clustering results, there is no existing standard validity metrics. Data distribution and application requirements play key roles in the performance of clustering algorithms as well as validation approaches (Xiao *et al.,* 2008). Therefore, the choice of the clustering algorithm and validity metric highly depend on the evaluation criteria. Gene expression data typically contains thousands of genes, biologists however often have interests on specific proportion of them or typically cluster for different subsets (David, 2001). For example, biologists sometimes may be particularly focus on certain small and tight clusters by neglecting other fuzzy clusters. If biologists may be interested in the gene's multiple biological function, FCM will become the favorable one. Although several novel clustering approaches have been proposed in this research, some limitations still exist, which can be improved in many different ways in future research.

Firstly, clustering gene expression has two different applications: gene based clustering and sample based clustering. This dissertation only addresses gene based clustering approach. However, clustering samples via genes as features is also significant, such as class discovery, normal and tumor tissue classification and drug treatment evaluation. Therefore, it is desirable to investigate the sample based clustering in future. In addition, clustering is the initial step for gene expression analysis, which is an unsupervised learning process. It should depend as little as possible on prior knowledge. For example, a clustering algorithm with parameter selection manually is expected to be replaced by the automatic one.

Specifically, if an algorithm estimates the "true" number of clusters in a dataset, it would be more favored than one requiring the pre-determined number of clusters.

Secondly, the purpose of clustering gene expression data is to reveal the underlying patterns and gain some biological insights of the data. Although global information regarding the dataset is usually unknown, (*e.g*. the number of clusters, the fuzzy exponent *etc*), some partial knowledge is often available. For example, some genes are known to be strongly correlated, and some genes participate in the same biological process. If a clustering algorithm could utilize this valuable information in clustering process, the clustering results would be more biologically meaningful. In this way, clustering would become a semi-supervised learning process by interactive exploration of the dataset.

Thirdly, it is should be noted that simultaneously expressed genes may not always share the same function or regulatory mechanism. Even when similar expression patterns are related to similar biological roles, discovering these biological connections among co-expressed genes is not a trivial task and requires substantial additional work. Conventional algorithms just take into account experimental measurements by ignoring available biological information about genes. Marie *et al*. (2013) propose a new unsupervised gene clustering algorithm achieving better biological significance. It relies on a new distance between genes by integrating biological knowledge into expression data. The concept of co-expressed biological function is proposed which can be assimilated to a set of genes that are involved in the function. Therefore, biological validation measures are further employed to examine the biological connections in the clusters. In future, biological information from sources will be investigated and integrated to identify the co-regulated genes, by which the clustering results will be more reliable and biological significance.

Fourthly, current approaches assign all samples into several clusters. However, it is possible that some sample, such as noises and noises, do not belong to any

clusters. In the future, these samples can be assigned into a ‘noise cluster’, which not only minimizes the noise influence, but also increase the cluster biological interpretation. (Yeung *et al*., 2001a) A good clustering algorithm can not only partition the dataset but also provide some biological representation of the cluster structure. In addition, biologists sometimes need a coarse overview of the data structure without consideration of the detailed information in the clusters. In fact, most of the existing clustering algorithms may not be flexible to different requirements for cluster scales on a dataset. For gene expression data, it is desirable to provide a scalable representation of the data structure, such as hierarchical clustering and SOM, can graphically represent the cluster structure.

Bibliography

Alter O., Brown P.O. and Bostein D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci*. USA, Vol. 97(18):10101–10106.

Andreas, D. and Francis B.F. (2005) *Bioinformatics: a practical guide to the analysis of genes and proteins* 3rd edition, wiley

Asyali, M.H. and Alci, M. (2005). Reliabiltiy analysis of microarray data using fuzzy c-means and normal mixture modeling based classification methods, *Bioinformatics*, Vol. 21, Issue. 5, pp. 644-649.

Audrey P. G. and Michael B. E., (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering, *Genome Biology*, 3(11): research0059.1–0059.22

Augenlicht, L.H., Taylor, J., Anderson, L. and Lipkin, M. (1991). Patterns of gene expression that characterize the colonic mucosa in patients at genetic risk for colonic cancer, *Proc Nati Acad Sci* USA, Vol. 88, Issue. 8, pp. 3286-3289.

Ayano S., Ken-ichi M., Takahisa N., Vo Thi., Takanori H., Akira A., Yoshihide F. and Hiroyuki S. (2013) Genetic lineages of undifferentiated-type gastric

carcinomas analysed by unsupervised clustering of genomic DNA microarray data. *BMC Medical Genomics*. 6:25

Badsha MB, Mollah MN, Jahan N, Kurata H. (2013) Robust complementary hierarchical clustering for gene expression data analysis by β-divergence. *J Biosci Bioeng*. 116(3): 397-407

Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics, 20*(16), 2493-2503.

Bar-Joseph, Z., Gerber, G. K., Gifford, D. K., Jaakkola, T. S., and Simon, I. (2003). Continuous representations of time-series gene expression data. *Journal of Computational Biology, 10*(3-4), 341-356.

Belacel, N., Cuperlovic-Culf, M., and Ouellette, R. (2004). Fuzzy J-Means and VNS methods for clustering genes from microarray data. *Bioinformatics,* 20(11), 1690-1701.

Belacel, N., Wang, Q., and Cuperlovic-Culf, M. (2006). Clustering methods for microarray gene expression data. *OMICS, 10*(4), 507-531.

Belkin, M., and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14, Vols 1 and 2, 14*, 585-591.

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of Computational Biology, 6*(3-4), 281-297.

Berendzen, K. W., Harter, K., and Wanke, D. (2009). Analysis of plant regulatory DNA sequences by transient protoplast assays and computer aided sequence evaluation. *Methods Mol Biol, 479*, 311-335.

Berriz, G. F., King, O. D., Bryant, B., Sander, C., and Roth, F. P. (2003). Characterizing gene sets with FuncAssociate. *Bioinformatics, 19*(18), 2502-2504.

Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Fucntion Algorithms, New York: Plenum Press.

Boratyn, G. M., Datta, S., and Datta, S. (2006) Biologically supervised hierarchical clustering algorithms for gene expression data. Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference 1, 5515-5518

Camastra, F., and Verri, A. (2005). A novel Kernel Method for clustering. *Biological and Artificial Intelligence Environments*, 245-250. doi: Doi 10.1007/1-4020-3432-6_29

Chen, X. (2009). Curve-based clustering of time course gene expression data using self-organizing maps. *J Bioinform Comput Biol, 7*(4), 645-661.

Chen Y, Bittner M, Dougherty E (1999) Issues associated with microarray data analysis and integration. Nature Genetics:213-215.

Chiang, J. H., and Hao, P. Y. (2003). A new kernel-based fuzzy clustering approach: Support vector clustering with cell growing. *Ieee Transactions on Fuzzy Systems, 11*(4), 518-527.

Chong E. K. P. and Zak S. H., *An Introduction to Optimization*, 3rd ed., John Wiley and Sons, Inc., New York, 2008.

Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J. (2006). Fuzzy c-means clustering with spatial information for image segmentation, Computerized Medical Imaging and Graphics, Vol. 30, Issue. 1, pp. 9-15.

Covell, D. G., Wallqvist, A., Rabow, A. A., and Thanki, N. (2003). Molecular classification of cancer: unsupervised self-organizing map analysis of gene expression microarray data. *Mol Cancer Ther, 2*(3), 317-332.

David R. Bickel. (2001) Robust Cluster Analysis of DNA Microarray Data: An Application of Nonparametric Correlation Dissimilarity. Proceedings of the Joint Statistical Meetings of the American Statistical Association (Biometrics Section).

Dejean, S., Martin, P. G., Baccini, A., and Besse, P. (2007) Clustering time-series gene expression data using smoothing spline derivatives. *EURASIP journal on bioinformatics and systems biology*, 70561

Dembele, D., and Kastner, P. (2003). Fuzzy C-means method for clustering microarray data. *Bioinformatics, 19*(8), 973-980.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via Em Algorithm. *Journal of the Royal Statistical Society Series B-Methodological, 39*(1), 1-38.

DeRisi, J., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P. S., Ray, M., . . . Trent, J. M. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genetics, 14*(4), 457-460.

Ding, Chris. Analysis of gene expression profiles: class discovery and leaf ordering. (2002) In Proc. of International Conference on Computational Molecular Biology (RECOMB), pages 127–136,Washington, DC

Dragomir, A., Mavroudi, S., and Bezerianos, A. (2004). Som-based class discovery exploring the ICA-reduced features of microarray expression profiles. *Comp Funct Genomics*, 5(8), 596-616. doi: 10.1002/cfg.444

Du, P., Gong, J., Syrkin Wurtele, E., and Dickerson, J. A. (2005) Modeling gene expression networks using fuzzy logic. *IEEE transactions on systems, man, and cybernetics*. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society 35, 1351-1359

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A, 95*(25), 14863-14868.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition, 41*(1), 176-190. doi: DOI 10.1016/j.patcog.2007.05.018

Foss, A. and Za ̈ane, A. (2002) A Parameterless Method for Efficiently Discovering Clusters of Arbitrary Shape in Large Datasets. In IEEE Intl. Conf. on Data Mining

Fraley, C. and Raftery, A. E. (2002). Clustering method for gene expression data. *Bioinformatics, 17*(10), 977-987.

Fu, L., and Medico, E. (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC bioinformatics* 8, 320-330

Futschik, M. E., and Carlisle, B. (2005) Noise-robust soft clustering of gene expression time-course data. *Journal of bioinformatics and computational* biology 3, 965-988

Gail, M., Krickeberg, K., Samet, J., Tsiatis, A., and Wong, W. (2005) *Statisical Method in Bioinformatics: An Introduction* , 2nd edition, Springer

Gasch, A. P., and Eisen, M. B. (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome biology* 3, RESEARCH0059

Genton, M. G. (2002). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research, 2*(2), 299-312.

Ghosh A., Mishra N. S., and Ghosh S. (2011) Fuzzy clustering algorithms for unsupervised change detection in remote sensing images, *Information Sciences*, Vol. 181, pp. 699-715.

Ghosh, D. and Chinnaiyan, A.M. (2002) Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18:275–286

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *Ieee Transactions on Neural Networks, 13*(3), 780-784.

Golub T.R., Slonim D.K., Tamayo P., Huard C., Gassenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield D.D., and Lander E.S. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science,* Vol. 286(15):531–537

Graves, D., and Pedrycz, W. (2010). Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy Sets and Systems, 161*(4), 522-543.

Guess, M. J., and Wilson, S. B. (2002). Introduction to hierarchical clustering. *J Clin Neurophysiol*, 19(2), 144-151.

Halder A., Pramanik S., and Kar A. (2011) Dynamic Image Segmentation using Fuzzy C-Means based Genetic Algorithm, *International Journal of Computer Applications*, Vol. 28, pp. 15-20.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001) On Clustering Validation Techniques. *Intelligent Information Systems Journal*, Vol. 6.pp. 452-458

Hartuv, Erez and Shamir, Ron. (2000) A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181

Hathaway, R. J., Huband, J. M., and Bezdek, J. C. (2005). A kernelized non-euclidean relational fuzzy c-means algorithm. *Fuzz-Ieee 2005: Proceedings of the Ieee International Conference on Fuzzy Systems*, 414-419.

Hautaniemi, S., Yli-Harja, O., Astola, J., Kauraniemi, P., Kallioniemi, A., Wolf, M., Ruiz, J., Mousses, S., Kallioniemi, O.P. (2003). Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps, *Machine learning*, Vol. 52, Issue. 1-2, pp. 45-66.

Herrero, J., Valencia, A., and Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17(2), 126-136.

Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: identification and analysis of coexpressed genes. *Genome Res, 9*(11), 1106-1115.

Hoffmann, R., Seidl, T., and Dugas, M. (2002). Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biology,* 3(7).

Hu, X., Yoo, I., Zhang, X., Nanavati, P., and Debjit, D. (2005) Wavelet transformation and cluster ensemble for gene expression analysis. *International journal of bioinformatics research and applications* 1, 447-460

Huang D and Pan W (2006): Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics* 22(10):1259-1268.

Huang, J. Z., Ng, M. K., Rong, H., and Li, Z. (2005). Automated variable weighting in k-means type clustering. IEEE Trans Pattern Anal Mach Intell, 27(5), 657-668.

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., . . . Friend, S. H. (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102(1), 109-126.

Iam-On, N., and Boongoen, T. (2012). A new locally weighted k-means for cancer-aided microarray data analysis. *J Med Syst*, 36 Suppl 1, S43-49.

Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C. F., . . . Brown, P. O. (1999). The transcriptional program in the response of human fibroblasts to *Serum*. *Science, 283*(5398), 83-87.

Jiang, D., Pei, J. and Zhang, A. (2003) DHC: A Density-based Hierarchical Clustering Method for Timeseries Gene Expression Data. In Proceeding of BIBE2003: 3rd IEEE International Symposium on Bioinformatics and

Bioengineering, Bethesda, Maryland, March 10-12

John, S. T., and Nello, C. (2004) *Kernel Methods for Pattern Analysis*", Cambridge University Press

Kim, S. (2006). A method of designing nonlinear channel equalizer using conditional fuzzy c-means clustering. *2006 6th International Conference on Signal Processing Proceedings, Vols I and Ii*, 1355-1358.

Krinidis, S., and Chatzis, V. (2010). A Robust Fuzzy Local Information C-Means Clustering Algorithm. *Ieee Transactions on Image Processing, 19*(5), 1328-1337.

Kohonen T. (1984) Self- Organization and Assiciative Memory. Spring- Verlag, Berlin

Kuragano, T., and Kasono, K. (2008) Curve Generation and Modification based on Radius of Curvature Smoothing. Ma Comput Sci Eng, 80-87

Lianjiang Z., Shouning Q., and Tao D. (2010) Adaptive fuzzy clustering based on genetic algorithm, In Proc. of 2nd conference on advanced computer control, Shenyang China, pp. 79-82.

Liang, L.R., Lu, S., Wang, X., Lu, Y., Mandal, V., Patacsil, D. and Kumar, D. (2006). FM-test: a fuzzy-set-theory-based approach to differential gene expression data analysis, *BMC Bioinformatics*, Vol. 7, Suppl No. 4, S7

Loquin K. and Strauss O. (2008) Histogram density estimators based upon a fuzzy partition", *Statistics and Probability Letters*, Vol. 78, pp. 1863–1868.

Luan, Y. H., and Li, H. Z. (2003). Clustering of time-course gene expression data

using a mixed-effects model with B-splines. *Bioinformatics, 19*(4), 474-482.

Maji P, Paul S.(2013) Rough-fuzzy clustering for grouping functionally similar genes from microarray data. *IEEE/ACM Trans Comput Biol Bioinform*, 10(2):286-99

Makretsov, N. A., Huntsman, D. G., Nielsen, T. O., Yorida, E., Peacock, M., Cheang, M. C., . . . Gilks, C. B. (2004). Hierarchical clustering analysis of tissue microarray immunostaining data identifies prognostically significant groups of breast carcinoma. *Clin Cancer Re*s, 10(18 Pt 1), 6143-6151.

Marie V., Sébastien L. and Jérôme P. (2013) A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data. *BMC Bioinformatics*, 14:42

Mehdizadeh E, Sadi-Nezhad S And Tavakkoli-Moghaddam R (2008) Optimization of Fuzzy Clustering Criteria by a Hybrid Pso and Fuzzy C-Means Clustering Algorithm, *Iranian Journal of Fuzzy Systems*, Vol. 5, pp. 1-14.

McLachlan, G.J., Bean R.W. and Peel D. (2002) A mixture model-based approach to the clustering of microarray expression data. Bioinformatics, 18:413–422

Memisevic, R., and Hinton, G. (2005). Improving dimensionality reduction with spectral gradient descent. *Neural Netw, 18*(5-6), 702-710.

Mewes, H. W., Hani, J., Pfeiffer, F., and Frishman, D. (1998). MIPS: a database for protein sequences and complete genomes. *Nucleic Acids Res, 26*(1), 33-37.

Mohammadi, A., Saraee, M.H, and Salehi, M. (2011). Identification of diseasecausing genes using microarray data mining and gene ontology, BMC Med Genomics, Vol. 4, pp. 12-23.

Nock, R., and Nielsen, F. (2004). An abstract weighting framework for clustering algorithms. *Proceedings of the Fourth SIAM International Conference on Data Mining*, 200-209.

Noordam, J. C., van den Broek, W. H. A. M., and Buydens, L. M. C. (2000). Geometrically guided fuzzy C-means clustering for multivariate image segmentation. *15th International Conference on Pattern Recognition, Vol 1, Proceedings*, 462-465.

Page, G. P., and Coulibaly, I. (2008). Bioinformatic tools for inferring functional information from plant microarray data: tools for the first steps. *Int J Plant Genomics, 2008*, 147563.

Pal, N. R., Aguan, K., Sharma, A., and Amari, S. (2007). Discovering biomarkers from gene expression data for predicting cancer subgroups using neural networks and relational fuzzy clustering. *BMC Bioinformatics, 8*, 5.

Pascual-Marqui, R. D., Pascual-Montano, A. D., Kochi, K., and Carazo, J. M. (2001). Smoothly distributed fuzzy c-means: a new self-organizing map. *Pattern Recognition*, *34*(12), 2395-2402.

Qin, J., Lewis, D. P., and Noble, W. S. (2003). Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19(16), 2097-2104.

Qu, Y. and Xu, S. (2004). Supervised cluster analysis for micorarray data based on multivariate Gaussian mixture, *Bioinformatics*, Vol. 20, No. 12, pp,

1905-1913

Richard O. D, Peter E. H, David G. S, *Pattern Classification*, Second Edition. (2001) Wiley Interscience.

Roded S.,, Adi M. and Ron S., (2003) CLICK and EXPANDER: a system for clustering and visualizing gene expression data, *Bioinformatics*, Vol. 19, No. 14, pp, 1787-1799

Rousseeuw, J.P. (1987). Silhouettes: a graphical aid to the interpration and validation of cluster analysis, *J. Comp. Appl. Math.*, Vol. 20, pp. 53-65.

Rui F, Asoke K. N. and Li G (2012) Clustering analysis for gene expression data: a methodology review. Proceedings of the 5th International Symposium on Communications, Control and Signal Processing, Rome, Italy

Shamir R. and Sharan R. Click: A clustering algorithm for gene expression analysis. In In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00). AAAI Press., 2000.

Shen, H. B., Yang, J., Wang, S. T., and Liu, X. J. (2006). Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets. *Soft Computing, 10*(11), 1061-1073.

Song, J. J., Lee, H. J., Morris, J. S., and Kang, S. (2007). Clustering of time-course gene expression data using functional data analysis. *Comput Biol Chem, 31*(4), 265-274.

Spellman PT, Sherlock, G., Zhang, M.Q., Iyer, V.R., Kirk Anders, K., Eisen, M.B., Brown, B.O., Botstein, B., Futcher, B.(1998) Comprehensive

Identification of Cell Cycle--regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridizationh. Molecular Biology of the Cell 9:3273-3297.

Steinley, D. (2003). Local optima in k-means clustering: what you don't know may hurt you. *Psychol Methods*, 8(3), 294-304.

Steinley, D. (2006). k-means clustering: a half-century synthesis. Br J Math Stat Psychol, 59(Pt 1), 1-34.

Sun, P.G., Gao, L. and Han, S. (2011). Prediction of human disease-related gene clusters by clustering analysis, *Int J Biol Sci*, Vol. 7, pp.61-73.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., . . . Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A, 96*(6), 2907-2912.

Tang, C., Zhang, A., and Pei, J. (2003) Mining phenotypes and informative genes from gene expression data. In In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'03), Washington, DC, USA

Tang C., Zhang L., Zhang A. and Ramanathan M. (2001) Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In Proceeding of BIBE2001: 2nd IEEE International Symposium on Bioinformatics and Bioengineering, pages 41–48, Bethesda, Maryland

Tang, R., and Muller, H. G. (2009) Time-synchronized clustering of gene expression trajectories. *Biostatistics* 10, 32-45

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., and Church, G. M. (1999). Systematic determination of genetic network architecture. *Nat Genet, 22*(3), 281-285.

Thomas J.G., Olson J.M., Tapscott S.J. and Zhao L.P. (2001) An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles. *Genome Research*, 11(7):1227–1236

Torkkola, K., Gardner, R.M., Kaysser-Kranich, T., Ma, C. (2001) Self-organizing maps in mining gene expression data, *Information Science*s, Vol. 139, Issue, 1-2, pp. 79-96.

Trevino, S., Sun, Y., Cooper, T. F., and Bassler, K. E. (2012). Robust detection of hierarchical communities from Escherichia coli gene expression data. *PLoS Comput Biol*, 8(2), e1002391.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., . . . Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics, 17*(6), 520-525.

Tsai, T. H., Milhorn, D. M., and Huang, S. K. (2006) Microarray and gene-clustering analysis. *Methods in molecular biology* 315, 165-174

Tseng, G. C. (2007). Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, 23(17), 2247-2255.

Tzortzis, G. F., and Likas, A. C. (2009). The Global Kernel k-Means Algorithm for Clustering in Feature Space. *Ieee Transactions on Neural Networks, 20*(7), 1181-1194.

Van Lung, H., and Kim, J. M. (2009). A Generalized Spatial Fuzzy C-Means Algorithm for Medical Image Segmentation. *2009 Ieee International Conference on Fuzzy Systems, Vols 1-3*, 409-414.

Veit S. and Ole N. J. (2010) A simple and fast method to determine the parameters for fuzzy c-means cluster analysis *Bioinformatics* Vol. 26 no. 22, pages 2841 – 2848

Wang, F. A. Z., Zhang, B. X., Wang, S. Y., Qi, M. A., and Kong, J. (2010). An adaptively weighted sub-pattern locality preserving projection for face recognition. *Journal of Network and Computer Applications, 33*(3), 323-332.

Wang, H, Wang, W, Yang, J and Yu, P. (2002) Clustering by Pattern Similarity in Large Data Sets. In SIGMOD 2002, Proceedings ACM SIGMOD International Conference on Management of Data, pages 394–405

Wang, J., Bo, T. H., Jonassen, I., Myklebost, O., and Hovig, E. (2003). Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics, 4*, 60.

Wang, J., Delabie, J., Aasheim, H., Smeland, E., and Myklebost, O. (2002) Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. *BMC bioinformatics* 3, 36

Wang, Y. , Angelova, M. and Ali, A. (2013) Fuzzy clustering of time series gene expression data with cubic-spline. *Journal of Biosciences and Medicines*, 1, 16-21.

Wu, F. X. (2008). Genetic weighted k-means algorithm for clustering large-scale

gene expression data. *BMC Bioinformatics*, 9 Suppl 6, S12.

Wu, K.L., Yang, M.S., 2002. Alternative c-means clustering algorithms. *Pattern Recognition* 35, 2267-2278.

Wu, W., Liu, X., Xu, M., Peng, J. R., and Setiono, R. (2005). A hybrid SOM-SVM approach for the zebrafish gene expression analysis. *Genomics Proteomics Bioinformatics, 3*(2), 84-93.

Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., and Somogyi, R. (1998). Large-scale temporal gene expression mapping of central nervous system development. *Proc Natl Acad Sci U S A, 95*(1), 334-339.

Xiao, J., Wang, X., and Xu, C. (2008). Comparision of supervised clustering methods for the analysis of DNA microarray expression data, *Agricultural Sciences in China*, Vol.7, Iss. 2, pp. 129-139

Xing, E.P. and Karp, R.M. Cliff: (2001) Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics,* Vol. 17(1):306–315

Yager, R. R., and Filev, D. P. (1994). Approximate Clustering Via the Mountain Method. *Ieee Transactions on Systems Man and Cybernetics, 24*(8), 1279-1284.

Yang, C. M., Wan, B. K., and Gao, X. F. (2003) Data preprocessing in cluster analysis of gene expression. *Chinese Phys Lett* 20, 774-777

Yang Q., Zhang D. and Tian F. (2010) An initialization method for Fuzzy C-means algorithm using Subtractive Clustering, Third International Conference on

Intelligent Networks and Intelligent Systems, Vol. 10, pp. 393–396.

Yang M.S. and Wu K.L. (2005) A modified mountain clustering algorithm, *Pattern Anal Applic*, Vol. 8, pp. 125–138.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001a). Model-based clustering and data transformations for gene expression data. *Bioinformatics, 17*(10), 977-987.

Yeung, K.Y., Haynor, D.R. and Ruzzo, W.L. (2001b) Validating Clustering for Gene Expression Data. *Bioinformatics*, Vol.17(4):309–318.

Yin, L., Huang, C. H., and Ni, J. (2006) Clustering of gene expression data: performance and similarity analysis. *BMC bioinformatics* 7 Suppl 4, S19

Yu W., Maia A., and Yang, Z. (2013a) A Framework for Density Weighted Kernel Fuzzy c-Means on Gene Expression Data. *Advances in Intelligent Systems and Computing* Volume 212, pp 453-461

Yu W., Maia A., and Akhtar A. (2013b) Fuzzy clustering of time series gene expression data with cubic-spline. *Journal of Biosciences and Medicines*, Volume 1, pp16-21.

Yu W., and Maia A.,(2012) Weighted kernel fuzzy c-means method for gene expression analysis, 2012 Spring Congress on Computational Biology and Bioinformatics (CBB-S), Xi'an China

Zhang, M., Adamu, B., Lin, C. C., and Yang, P. (2011) Gene expression analysis with integrated fuzzy C-means and pathway analysis. Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Conference 2011, 936-939

Zhang, D. Q., and Chen, S. C. (2003). Clustering incomplete data using kernel-based fuzzy C-means algorithm. *Neural Processing Letters, 18*(3), 155-162.

Zhang, D.Q. and Chen, S.C. (2004). A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, *Artificial Intelligence in Medicine*, Vol. 32, Issue. 1, pp. 37-50.

Zhang, S.H., Wang, R.S., Zhang, X.S. (2007). Identification of overlapping community structure in complex networks using fuzzy c-means clustering, *Physica A*, Vol. 374, pp. 483-490.

Zhang, Z. G., Cao, H., Liu, G., Fan, H. M., and Liu, Z. M. (2013). Bioinformatic analysis of microarray data reveals several key genes related to heart failure. *Eur Rev Med Pharmacol Sci, 17*(18), 2441-2448.