

CLUSTERING ALGORITHMS FOR TIME SERIES
GENE EXPRESSION IN MICROARRAY DATA

Guilin Zhang

Thesis Prepared for the Degree of
MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2012

APPROVED:

Qunfeng Dong, Major Professor
Yan Wan, Major Professor
Xiang Gao, Committee Member
Sam Atkinson, Chair of Department of
Biological Sciences
Mark Wardell, Dean of the Toulouse
Graduate School

Zhang, Guilin. Developing Efficient Clustering Algorithms for Time-Series Gene Expression Data. Master of Science (Biology), August 2012, 53 pp., 24 tables, 3 illustrations, 75 numbered references.

Clustering techniques are important for gene expression data analysis. However, efficient computational algorithms for clustering time-series data are still lacking. This work documents two improvements on an existing profile-based greedy algorithm for short time-series data; the first one is implementation of a scaling method on the pre-processing of the raw data to handle some extreme cases; the second improvement is modifying the strategy to generate better clusters. Simulation data and real microarray data were used to evaluate these improvements; this approach could efficiently generate more accurate clusters. A new feature-based algorithm was also developed in which steady state value; overshoot, rise time, settling time and peak time are generated by the 2nd order control system for the clustering purpose. This feature-based approach is much faster and more accurate than the existing profile-based algorithm for long time-series data.

Copyright 2012

by

Guilin Zhang

ACKNOWLEDGMENTS

I would like to greatly express my gratitude to my advisor, Dr. Qunfeng Dong and Dr. Yan Wan for their guidance and support. I would like to thank Department of Biological Science, Toulouse Graduate School in University of North Texas for funding my work. I would also thank Dr. Xiang Gao for her support as committee member of my thesis. I am so grateful for the support of my families, my fiancée Yixuan Liu, my friends and colleagues Guangchun Cheng, Claudia Vilo, Ruichen Rong, Michael Plunkett, Kashi Vishwanath. Generally, I appreciate all the people I know for helping me finish the thesis.

CONTENTS

ACKNOWLEDGMENTS	iii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. BACKGROUND STUDY AND RELATED WORK	4
2.1. Microarray Data	5
2.2. Clustering on Static Data	7
2.3. Clustering on Time Series Data	12
2.3.1. Raw-Data-Based Approaches	13
2.3.2. Feature-Based Approaches	17
2.3.3. Model-Based Approaches	21
CHAPTER 3. OVERVIEW ON ONE CLUSTERING ALGORITHM	26
3.1. Preprocessing Raw Data	26
3.2. Selecting Model Profiles	26
3.3. Grouping Significant Profiles	29
3.4. Problems on This Algorithm	29
CHAPTER 4. MODIFICATION ON THE GREEDY CLUSTERING ALGORITHM	31
CHAPTER 5. CONTROL SYSTEM-BASED CLUSTERING ANALYSIS	40
CHAPTER 6. DISCUSSION AND FUTURE RESEARCH DIRECTION	44
BIBLIOGRAPHY	47

CHAPTER 1

INTRODUCTION

Microarray has been established over decades as an efficient technique to simultaneously produce large-scale gene expression data, and has found its value in diverse biological applications, such as genome sequencing, phylogeny analysis, pathway construction, and bio-marker detection, among many other applications [8][75][65]. To overcome analytical difficulties caused by the high-dimensional data, clustering analysis, which groups genes with similar expression patterns, has been practically proven to be a very useful pre-processing technique. For instance, in the application of pathway construction, the advantages of clustering analysis are primarily three-fold. First, clustering analysis significantly reduces the dimension of parameter identification, by transiting computational efforts from individual genes to gene groups/clusters. Second, clustering analysis helps with identifying genes with common transcription factors, as genes with similar expression patterns are highly possible to be correlated. Third, clustering also facilitates the inference of unknown genes' biological functions, if these unknown genes are grouped with other genes with known functions.

In recent years, bolstered by the advances in microarray techniques, that collects microarray data at multiple sampling time points becomes possible. Compared to traditional static microarray expression data, time-series microarray data can capture the dynamics of gene expression variation and thus provide rich insight into underlying biological systems [4][30][48]. For instance, time-series data can trace how the impact dynamics is propagated through the pathways when the subject are exposed to some disturbance. Such causality information cannot be reached otherwise.

The emergence of time-series microarray data necessitates clustering algorithms that can fully exploit the rich information provided by time-series data. Many clustering algorithms (such as k means and fuzzy c means) have long been established to process static data

[67][11][12]. One intuitive way to adapt these algorithms to the time-series case is to view the time-series data as a series of static data, and use static data clustering algorithms at each of these time points. However, this approach cannot be effective because it wipes out the critical correlation information along the time scale. As such, it is necessary to develop clustering algorithms that can capture such correlation.

There exist few clustering algorithms in the literature to process time-series microarray data. Roughly speaking, these clustering algorithms can be classified into three categories: model-based, profile-based, and feature-based (please see a thorough review in Chapter 2). In this thesis, I thoroughly study a profile-based clustering algorithm designed for short time-series data. In particular, this algorithm involves the following steps: preprocessing, representative profile selection, and grouping. I identify two fundamental problems of this algorithm: 1) the processing step amplifies noise and hence makes the clustering extremely sensitive to noise, and 2) the performance of the grouping is highly dependent upon the selection of a threshold. As such, I modify the algorithm to account for the above two deficiencies, and compare the performance of the two algorithms using real microarray expression data of bacteria *Helicobacter pylori* wild type. I find that our modified algorithm is more efficient and has better performance compared to the original algorithm.

In addition, as the profile-based method is not effective for long-time-series data, I introduce a new feature-based clustering algorithm, from dynamical systems point of view. In particular, I take advantage of the fact that many time-series microarray data is designed to track gene expression changes in response to disturbances. For instance, to discover PTSD pathways, battle-field-like stress conditions are applied to target mice, and mouse brain samples are collected as the disease develops over time so as to understand gene expression changes in response to the stress condition. In the new algorithm, I use control-theoretic features (that are widely used in the field of control to capture system responses, such as rise time, overshoot, and steady-state value), to group time-series microarray response data.

The approach has the following advantages: 1) the control-theoretic features represent the most important features that reveal the underlying physics of the dynamical systems, 2) the algorithm is fast as it only uses the key temporal features for clustering, and 3) the algorithm is flexible and can be used for both short- and long-time series data. In this thesis, we present our preliminary result on using this algorithm to cluster time-series response data.

This thesis is organized as follows. In Chapter 2, the existing algorithms to cluster static and time series data are thoroughly reviewed. In Chapter 3, we describe in detail the procedure and the implementation of a clustering algorithm designed for short time series data. Chapter 4 includes two improvements that we have developed to enhance the performance of the short time series clustering algorithm. We describe the implementation of the modified algorithm and then compare our modified algorithm with the original algorithm. In Chapter 5, we illustrate our idea about a novel control-theoretic clustering algorithm suitable for long time-series microarray data, and illustrate the algorithm through a simple example. Finally, a brief conclusion and future work is described in Chapter 6.

CHAPTER 2

BACKGROUND STUDY AND RELATED WORK

Microarray has been established as an efficient technique to exam large scale information for genetic investigation simultaneously, and has been used in diverse biological applications [8][75][65]. As for its applications on different aspects, especially different biological parts, there are many questions that have been explained at depth. Meanwhile, the size of a microarray has increased rapidly with the advances of genome projects because of the developments in experimental designs, image processing, statistical analysis and data warehousing, e.g, a whole genome array easily contains several thousand genes/genetic sequences, and can have as many as a hundred thousand sequences [2]. The data that microarray technique generate is high-dimensional which has several to thousands of dimensions, and each dimension can be thought of as one characteristics from one measurement. High-dimensional data are very difficult to visualize, think about, analyse and impossible to enumerate. In addition, it is very hard to design tools and methods to deal with the high dimensions of this kind of data. Thus, it is necessary to decrease the dimensions of microarray data in order to effectively mine the information [35].

Generally, there are three major advantages to clustering analysis on microarray data, which are one kind of typical time series data.

First, clustering analysis significantly reduces the dimension of an identification problem by transiting computational efforts on individual genes to gene groups/clusters; this facilitates the construction of pathways at a coarse level. Such reduction is crucial because microarray experiments often suffer from excessive freedom in model inference, i.e. they cannot provide sufficient data information for the inference of high-dimensional models, due to

the high cost and difficulties in performing biological experiments. In this thesis, many algorithms and procedures which can deal with high dimensional data are reviewed and evaluated so that they may be implemented for microarray data.

Second, genes with similar expression patterns are highly possible to be co-regulated and hence they are potential resources for identifying common transcription factors; this allows constructing gene expression pathways at a more detailed level. In addition, they are constructed by using common features of gene clusters rather than individual genes in the following analysis like pathway construction. The analysing information can be used in following analysis like pathway construction.

Third, clustering analysis also aids the inference of unknown genes' biological functions, which also helps the understanding of the mechanisms of involved pathways. Some unknown genes can be grouped into the same clusters since they have similar biological functions or expression patterns, and then they will probably have much smaller distances to some genes but not to some other genes.

Clustering analysis is treated as the first step on microarray data for analysis such as data mining, feature selection and machine learning [73]. From this point, different clustering algorithms and procedures need to be surveyed on their efficiencies and accuracies, and then should be implemented with real data which can be obtained free from multiple sources.

2.1. Microarray Data

Microarray is an efficient technology that collects microscopic spots attached to solid surface. It can generate information for hundreds or thousands of genes. It is used in many biological disciplines including gene sequencing, molecular evolution, pathology, and so on. The precursor of microarray is southern blotting; it is to attach fragmented DNA to solid substrate and probe them with known DNA or fragments [49]. An microarray can contain tens of thousands of plots, and each plot contains a tiny specific DNA fragment, known as probe; and each probe can be short sections of genes or fragments of introns which can

hybridize objective DNA under particular conditions; and finally signals like fluorescence are released to check levels of gene expressions. The size of microarray has been increased dramatically with more and more genetic information can be read simultaneously [63]. Due to the high-throughput information microarray can yield, it is widely used in many aspects; for example, thousands of gene expressions are monitored simultaneously to study pathogens by comparing which gene expression is changed between infected and uninfected cells and tissues [1]; it can help people to assess gene content in cells and organisms [52]; identifying single nucleotide polymorphism in populations to detect candidate drugs, predict disease, evaluate germline gene mutations and so on [25].

There are some disadvantages on microarray technique. For instance, diverse platforms have different standards so that it is hard to replicate in different platforms even using the same samples [5]. It is necessary for scientists to build a robust and reliable data analysis platform and this is currently a major limiting factor for microarray technique. Recently, scientists are incorporating microarrays as a comprehensive tool to monitor global gene expression patterns [64].

Through analysis of microarray, people study different topics by investigating hidden information by elucidating patterns of gene expression levels. The information collected from microarray platforms can be further used in some other research like genome sequencing, phylogeny studies, pathway construction etc. [8][28][40]. Generally, after obtaining microarray data, the work include preprocessing raw data to make data discrete; differential expression and validation to integrate information to particular format; clustering to decrease dimensions; and doing gene and genome annotation as well as pathway construction[64]. Generally, the goal of most microarray experiments is to collect gene expression levels of tens of thousands of genes [31]. To survey questions that we are interested in, it is important and necessary to transform microarray data using different methods.

DNA microarray is a powerful tool used in many aspects especially on time series data [58]. Currently, the challenging step is to interpret mass of data which are commonly composed of millions of measurements, to useful information. Clustering analysis is an efficient way to address this challenge and essential to data mining [29], including genome annotation and pathway construction. There are a bunch of different methods to execute cluster analysis in microarray data. In the following sections, we will specially concentrate on some classic cluster analysis and introduce our particular methods.

2.2. Clustering on Static Data

Clustering analysis is trying to partition gene expression data into groups for further analysis such as data mining and biological studies. For the terms *cluster* and *clustering*, we would better to clarify the differences between them. In this thesis, the definition of *cluster* is simple: a group of data. The definition of clustering has two aspects: it could be the process of grouping analysis or a set of such grouping data and generally it contains all the objects. In other words, *clustering* is generally the process that groups data objects into different classes, and those classes are called a *cluster*, in which the grouped objects in one certain cluster should be more similar to each other and more dissimilar to separate clusters. In addition, it is necessary for us to distinguish clustering analysis to some other similar methods such as pattern recognition, decision analysis and discriminant analysis which are commonly used in statistical areas. Among them, pattern recognition, decision analysis and discriminant analysis belong to supervised classification which means they rely on predefined classes and training examples in classifying steps; on the other hand, clustering analysis is unsupervised classification which does not need rules for grouping data from a given set of pre-defined objects.

A typical microarray experiments contain a lot of genes (up to 100), but contains fewer samples (less than 100). Clustering analysis can be classified into two kinds of methods [29]. The first clustering analysis is to treat genes as objects and samples as features. In this

case, people believe that co-expressed genes with similar expression patterns can be grouped into same clusters [7] [13]. Genes in same groups in which the expression patterns are similar may well have common functions in different biological samples, such as co-regulation and expression pathways. The second clustering analysis is based on samples and to consider genes as features; objects in each group may have similar phenotypes, especially in biomedical research, such as cancer samples and other difficult and complicated clinical syndromes [20]. The major difference between the two kinds of clustering analysis is that they have different aims to classify gene expression data. In this thesis, we will pay our attention on first clustering analysis which is based on genes. So far, most algorithms, including hierarchical and K-means approaches, can be used in both kinds of analysis.

A large of number of previous clustering analyses was based on static data. As for static data, it refers to those data which do not change with time. Clustering approaches handling static data can be classified into five categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods [27].

The first method on static data is partitioning methods. Assume that we have n unlabeled data tuples, then developed a parameter k which $k \leq n$, that stands for k partitions and there is at least one object in them. If each object belongs to exactly one cluster, the partition is crisp; otherwise, the partition is marked as fuzzy if one object can be classified into more than clusters. To handle the crisp partitions there are two famous methods: k -means [36][50] and k -medoids [16]. The k -means method takes the mean value of all objects as the value for each cluster while the k -medoids method grab most centrally located objects in the cluster as the value of cluster. On the other hand, to handle the fuzzy partitions, there are two methods like above. The first one is fuzzy c -means [26] algorithm and the other one is fuzzy c -medoids [38] algorithm. It is believed that the above heuristic algorithms are good at solving spherical-shaped clusters and suitable for small and medium data sets, but they do not work well for non-spherical clusters and large data sets. Some other algorithms including

density based methods, Gustafson-Kessel and adaptive fuzzy clustering algorithms [37] were designed for non-spherical-shaped clusters and large data sets. Generally, many clustering algorithms and procedures for static data are implemented with the spirit of partitioning methods, as we introduced above, and the partitioning methods are the most famous for handling static data sets, especially the k-means algorithm, the k-medoids algorithms and the fuzzy c-means algorithm.

A second method is called density-based method. Here the definition of density is number of objects or data points. One algorithm in this category is OPTICS [3] which computes a set of increased clustering ordering in those both automatic and interactive clustering. A wide range of parameter settings are preprocessed and then density-based clustering are converted by obtaining the information of the clustering ordering, and then it is much easier to select suitable parameter values. One another algorithm in density-based methods is called DBSCAN [15]. It produces a clustering explicitly compared to OPTICS by computing the density in the neighborhood which exceeds the predefined thresholds and growing the cluster.

The third method for clustering static data is called hierarchical method, which classifies objects as results of tree clusters. Generally, people claim two types of methods which implement the spirit of hierarchical methods. The first one is called agglomerative method. Like the natural agglomerative procedures, this kind of methods start with grouping data objects into small clusters, and then merging those different small clusters into relatively larger ones, back and forth. Finally, all the data objects are grouped into one huge cluster or the grouping process are terminated when some condition thresholds are satisfied, such as predefined number of clusters. The second method is so called divisive method, since the procedures of this method is just opposite to the agglomerative method. It starts with all objects as a single cluster and then all the data objects are partitioned into smaller clusters. Divisive method is more complicated than agglomerative methods since the former one needs

flat clustering method which is different from hierarchical method to partition data objects. Above all, agglomerative method is also called bottom up method and starts with single data object and aggregate them into clusters; divisive method is also called top down method and starts with complete data set and divide the data set into partitions. Hierarchical clustering methods are widely used but do not perform well in many cases when merging or splitting data objects in different situations. Thus, the hierarchical methods are implemented with some other algorithms to improve its clustering quality. For example, BIRCH [74] method make agglomerative method much better by introducing iterative relocation to refine results; and Chameleon [32] and CURE [23] methods improve divisive method by using analysis of object linkages for each hierarchical partitioning step.

Measurement	Formula
Euclidean distance	$d_{ab} = \sum_c e_{ac} - e_{bc} $
Manhattan distance	$d_{ab} = \sqrt{\sum_c (e_{ac} - e_{bc})^2}$
Mahalanobis distance	$d_{ab} = (e_a - e_b)' \Sigma^{-1} e_a - e_b$; here is covariance matrix of data
Pearson correlation	$d_{ab} = 1 - \frac{\sum_c (e_{ac} - e_{ac}) (e_{bc} - e_{bc})}{\sqrt{\sum_c (e_{ac} - e_{ac})^2} \sqrt{\sum_c (e_{bc} - e_{bc})^2}}$
Uncentered correlation	$d_{ab} = 1 - \frac{\sum_c e_{ac} e_{bc}}{\sqrt{\sum_c e_{ac}^2} \sqrt{\sum_c e_{bc}^2}}$
Spearman rank correlation	Similar to Pearson correlation, but using rank of e_{ac} and e_{bc} instead of direct expression values, the ranks are selected expression value among conditions $c = 1 \dots c$
Absolute or squared correlation	$d_{ab} = 1 - \left(\frac{\sum_c (e_{ac} - e_{ac}) (e_{bc} - e_{bc})}{\sqrt{\sum_c (e_{ac} - e_{ac})^2} \sqrt{\sum_c (e_{bc} - e_{bc})^2}} \right)^2$
Correlation coefficient	$d_{ab} = \frac{dCov(a,b)}{\sqrt{dVar(a)dVar(b)}}$

Figure 2.1. Table some similarity measures for static data in clustering analysis. d_{ab} : distance between expression patterns for gene a and b ; e_{ac}/e_{bc} : expression levels of gene a/b under the condition c [32].

One another method different from the methods is grid-based method. Just as the name implies, grid-based methods generate a grid structure by quantizing the object space into finite number of cells, and then do all the operations on the grid structure. Let's set an example to illustrate grid-based methods, Sting [69] creates grid structure by using different

sizes of rectangular cells and each kind of cells represents several levels of resolution for each data object. The attributes for each cell are calculated as the form of statistical information and then that information would be computed and stored. Each cell should be in different layers in grid structure. Generally the start point of the query is from a relatively high level of the structure. When the query is on different levels, the confidence interval which reflects the cells relevance for the given query is calculated and then the relevance of data objects before quantizing are assessed. If the cells are irrelevant to the query, they would be removed temporarily and be considered latter. When the high level of the structure is traversed by the query, the query will move to those lower levels of relevant cells recursively. The whole process will be terminated when all the layers of the structure are done.

The last clustering analysis for static data we introduced here is called model-based methods. This kind of methods builds a model for each cluster and tries to find a best matched data to the model. There are different approaches to develop model-based approaches. One approach is called neural network approach which is about competitive learning. For example, ART and self-organizing feature maps [9] [33] are two artificial neural network methods. They generate low-dimensional, discreted input space for samples. On the other hand, AutoClass [10] is the representation of another method called statistical. It uses Bayesian statistical analysis to gather some criteria to estimate clusters and the related numbers of clusters.

Static data are much easier to handle because they have few dimensions. Thus, most algorithms and procedures are developed to mine information on this kind of data. However, trend for data analysis and data mining are based on time series data since they contain more information and are more meaningful than static data. Respectively, the relative algorithms and procedures are more complicated than those introduced algorithms and procedures above. Fortunately, many approaches for time series data are derivative from or common to the approaches especially for static data. In the following, I would like to review some algorithms and procedures for time series data.

2.3. Clustering on Time Series Data

The above five algorithms/procedures are for static data but not time series data. Time series of data are unlike static data because they change with time. These kinds of data are more common than static data in different field like science, engineering, healthcare, finance, and so on [43]. Grouping time series data is challenging but acquisition is desirable. As to the data which are time series, they can be from multiple sources through natural process, biological process and engineered process in different periods. Compared to clustering analysis for static data, the approaches for time series data are deficient. However, developing new approaches on clustering analysis for time series data will be the trend in future. For unlabelled time series data, they also need algorithms or procedures to classify objects into certain groups. When time series data are studied, several distinctions should be identified: equal or unequal length, discrete-valued or continuous valued, uniformly or non-uniformly sampled, and univariate or multivariate [43]. We need to make data equal length, discrete-valued, and uniformly. There is a wide range of methods of make data suitable to the requirement including log ratio and scaling data we used in this thesis.

Time series clustering has been proven an effective way to be part of data mining research on gene expression data. There is a rapid increasing interest to develop new algorithms and procedures for clustering analysis on time series data. By reviewing most previous clustering analyses on time series data, T. Warren Liao [43] had divided past approaches into three categories: raw-data-based approach, feature-based approach and model-based approach. It is claimed that those approaches were mainly based on some common points that they modified previous algorithms and procedures which are particularly for static data clustering analysis so that they can handle time series data, or they try to find some methods to transform time series data to forms of static data such that the existing algorithms and procedures can be used. The so called raw-data-based approach is mainly based raw time series data and its major modification on existing algorithms and procedures for static data is

that it changes the measurements of similarity distance and then different kinds of distance matrices are generated. However, they do not process raw time series data anyway. On the other hand, the feature-based and model-based approaches will pre-process raw time series data more or less. The feature-based approach is to take feature extraction firstly. As we talked above, feature extraction is a supervised classification which needs pre-defined class in advance. The model-based approach is firstly to extract model parameters for raw time series data and then build suitable models. Both feature-based and model-based approaches will process raw time series data in advance and then take advantage of existing algorithms to analyze data. In other words, the latter two methods do not need new algorithms or procedures. In this thesis, we will take a new method which belongs to one of the above approaches we apply control system theory to time series data as we take time series gene expression as input and consider experiments (heat shock, hormone treatments, etc.) as disturbance for the control system. Our method should be classified into feature-based approach since we pre-process raw data and grab some features based on those data, and then we do analysis on those features. To build the control system, we capture several transient performance as the features for each data, and then we consider the similarities among different control systems based on the transient performance parameters including rise time, settling time, overshoot and peak time.

2.3.1. *Raw-Data-Based Approaches*

Raw-data-based approaches handle raw data which collected from observations. Analysis are taken directly on the original values of the data or some converted forms based on the raw data, like logarithm ratio. One advantage of this method is to keep enough information from data and the disadvantage is that the analytical results can be influenced by some information which do not contribute to the results we are interested in.

Gath and Geva [18] developed a new set of unsupervised fuzzy clustering algorithms. They firstly applied to a series of data points (P) as set of unordered observations and

calculate the membership matrix for certain number of clusters, and then divide the series into P segments that in each segment there are K data points. The membership value for each data point in each segment is computed from temporal value in each segment. One parameter, symmetric Kullback-Leibler distance is calculated for each cluster to make the optimal number of clusters. Given a small threshold value, optimal value will be retained if it is bigger than the threshold value otherwise it will be discarded. The final membership matrix and number of clusters shows the weights in time varying, mixture probability distribution function. Policker and Geva [55] took advantage of the algorithm on non-stationary time series data and estimate temporal drift of series probability distribution function (PDF) which is often related to hidden Markov model.

Shumway [61] used local stationarity Kullback-Leibler discrimination measures of distance to generate the optimal time-frequency statistics time series and then obtain final clusters combined to agglomerative hierarchical cluster analysis. Since the goal of their work was to distinguish signals between regular earthquakes and mining explosion, the number of clusters in their resultant set was two. Golay et.al [19] studied magnetic resonance imaging (MRI) data which are time series about human brain activity under stimulus. They handled

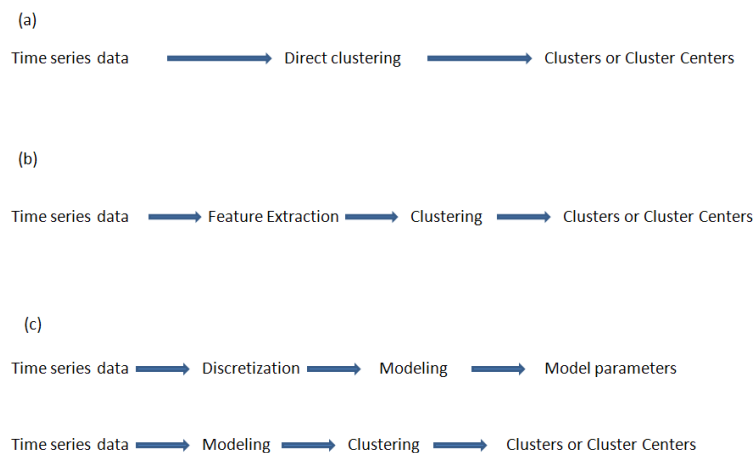


Figure 2.2. Three time series clustering approaches: (a) raw-data-based, (b) feature-based, (c) model-based [43].

raw time series data directly by measuring Euclidean distance and two cross-correlation-based distances. They defined nine different pre-processing procedures on data set in the fuzzy clustering algorithms. The nine preprocessing procedures and three distances are executed to generate clusters which are based on pixel time according to similarity but not their proximity. In addition, they also discussed the optimal number of clusters for time series data by claiming that obtaining as more as possible clusters and then cut down those clusters in order to reduce redundancy or acquisition. They talked about the different influences of the number of optimal clusters when handling time series data, however, they did not provide a procedure how to determine the exact number of optimal clusters.

Authors	Distance measure	Algorithm	Criterion
Policker and Geva	Euclidean distance	Fuzzy clustering	Symmetric Kullback-Leibler distance
Shumway	Kullback-Leibler discrimination measure	Agglomerative hierarchical	N/A
Golay et al.	Euclidean distance	Fuzzy <i>c</i> -means	Cluster variance
Möller-Levet et al.	STS distance	Fuzzy <i>c</i> -means	Cluster variance
Wismüller et al.	N/A	Neural network clustering	Cluster variance
Van wijk and van selow	Root mean square	Agglomerative hierarchical	N/A
Kumar et al.	Gaussian models of data errors	Agglomerative hierarchical	N/A
Košmelj and Batagelj	Euclidean distance	Relocation clustering	Ward
Liao	Euclidean distance & Kullback-Leibler distance	<i>k</i> -means & Fuzzy <i>c</i> -means	Cluster variance

Figure 2.3. Summary of algorithms of raw-data based approaches [61][18][71][51][39][44][19][55][34].

For the dynamic biomedical image study including MRI studies, Wismüller et al. [71] showed that a hierarchical unsupervised learning procedure can demonstrate the structures of data sets by gradually increasing resolution during clustering analysis when the minimal free energy vector quantization (VQ) is used. In their studies, activated regions, regional

abnormalities and suspicious lesion in the MRI images which are time series data can be identified.

Miller-Levet et al. [51] used Wismler's algorithm which belongs to fuzzy-c-means family on dynamic DNA microarray data. They took advantage of short time series distance (STS distance), instead of Euclidean distance and Pearson correlation coefficient, to measure the similarities of shape which are made up of amplitude and temporal information of time series data, since the latter two measurements cannot be used in dynamic analysis. STS distances are incorporated into the standard fuzzy c-means algorithm and based on this they generated a fuzzy time series clustering algorithm. This algorithm is for short time series data and no samples for long time series had been examined. Komelj and Batagelj [34] revised relocation clustering procedure which is for static data originally. They created a cross sectional approach to incorporate time dimension of the data but the procedure does not care about the correlation among variables and works only with equal length of time series. They developed a model to evaluate the time-dependent linear weights. Among all the possible clusterings, the best one will have the minimum generalized Ward criterion function. Liao et al. [45] developed some new algorithms based on this cross-sectional approach including k-means and fuzzy c-means.

Kumar et al. [39] developed a new algorithm to analyze dynamic retail data in business. They proposed a distance function based on the assumed independent Gaussian models of data errors as well as hierarchical clustering to group raw data into clusters. They assumed that the preprocessing step can eliminate non-dynamic effects and different points shall be on the same scale. Liao [44] developed a procedure for multivariate time series data. Firstly, the procedure applied the k-means or fuzzy c-means method to convert data from continuous forms to discrete forms. The traditional parameter Euclidean distance is used. After the data are converted, they used k-means or FCM algorithms to group data into clusters, and during

this step, Kullback-Liebler distance are taken to measure the distances between different time series.

2.3.2. *Feature-Based Approaches*

Clustering analysis directly on raw data will often occupy high dimensional space and may well contain strong noises [72]. This is because of many factors including background noise, fast sampling rate, system errors, etc. Thus, doing analysis directly on raw data may well not be desirable. To overcome the shortcomings on raw-data based approaches, some alternative methods are developed. Here we will introduce featured-based approaches which extract different sets of features from raw data. There are many advantages of featured-based approaches compared to raw data-based approaches, like decreasing dimensions by selecting suitable features, avoid the influence of noise dramatically, etc. The extraction features are dependent on different applications and one set of features perform well on one application but may well not work well for another application. Thus, different feature-based approaches do feature selection commonly, but their methods are pretty different. Jiang et.al [29] introduced a new algorithm which is especially designed for short time series data. They predefined some profiles and then assigned all expressed genes into the profiles. The objectives of the algorithm are the profiles, but not direct genes. They captured the potential distinct patterns from different profiles representing different genes. The preprocessing step in the algorithm is log ratio method, which can reduce the dissimilarities between data in the profiles; in other words, the preprocessing process can make data obscure so that we can focus the major differences among profiles. This algorithm took correlation coefficient as the distances among different profiles and it grouped different profiles into clusters. As for the algorithm, it firstly built model profiles which could generally be treated as models; in the following, other profiles could be grouped into the model based on the correlation coefficient scores. The authors implemented microarray gene expression data obtained from Stanford Microarray Database (SMD) to simulate the efficiency of the algorithm. In this

thesis, we made two improvements for this algorithm: one is for the preprocessing step and we considered the microarray time series data which do not change the expression levels at different time points; the second improvement we made was the grouping step. The algorithm selected the largest clusters among all the model profiles and released all the other profiles, and then repeated the grouping step again and again until all the model profiles had been added general profiles; we modified the algorithm in the aspect of grouping the general profiles directly to model profiles based on their correlation coefficient scores that means each general profile would be grouped into the cluster which contains the nearest model profile. We will further discuss the algorithm and the changes we made in the following part.

Wilpon and Rabiner [70] modified standard partitioning methods k-means method to adjust to time series data for use in isolated word recognition. They modified the original algorithm in many aspects including that how to find the center of clusters among all time series data objects; how to change the value of k in order to increase the number of clusters; how to obtain the final clusters, etc. In their research, there are two dimensions to catch the state of word recognition: each frame represents a vector of linear predictive coding coefficients and each pattern is the representation of one specific spoken work which has replication and inherent duration. The distance between two frames is measured to check the similarity by using a symmetric distance measure which is defined based on Itakura distance. Since this modified algorithm was developed in 1980s, it did not meet some requirements to some new time series data, however, this modified k-means clustering algorithms shown its proficiency and advantages compared to the original algorithm and those well-established unsupervised without averaging clustering algorithms. In addition, the algorithm developed by Wilpon and Rabiner is nearly the earliest effort on modification to traditional methods for use in time series data. One another effort of modification on traditional clustering methods and based on features extraction was developed by Shaw and King [60]. They did analysis on spectra which was normalized by the amplitude of the largest peak as well as those spectra

were obtained from original time series which means were adjusted to zeros. The scientists used two indirect hierarchical algorithms to cluster the time series data, which were called Wards minimum variance algorithm and the single linkage algorithm. At the same time, they used principal component analysis to filter and clusters data objects, and Euclidean distance was measured to check the similarity among different objects. Vlachos et al. [66] used one mathematical transform to develop a new modified algorithm on traditional k-means method. It was Haar wavelet transform at various resolutions to generate incremental clustering on time series data. Among all the data objects which were time series, Haar wavelet decomposition is used to compute as the characters for each data object and then using traditional k-means method to do cluster starting from those objects which are in coarse level and gradually considering those objects which are in finer levels. After one resolution is finished, it will generate a final center for all time series data and this center will be considered as the initial center for the next resolution. When this step is executed, the length of data structure reconstructed from Haar decomposition doubles as the algorithm moves from one resolution level to next level. At the same time, the coordinates of centers at different resolution levels change somehow. For example, the coordinate of one center at the end of one level i is doubled so that it can match the dimensionality of the value at the resolution level $i + 1$. Each cluster is divided by the cardinality of the dataset and the sum of number of incorrectly clustered objects is calculated at the end of each resolution level as the clustering error.

Owsley et al. [46] developed a new modified algorithm which group machine tool monitoring time series data into clusters which are related to discrete hidden Markov models (HMM). This algorithm is called sequence cluster refinement algorithm (SCRA) and it represented clusters as HMM instead of template vectors which is something related to vector quantization algorithm. This algorithm extracted transient events in the data signals by template matching and then forms a high resolution time-frequency representation for transient

events. In their research, self-organization map algorithm was used to decrease the data dimensions.

Authors	Features	Distance measure	Algorithm	Criterion
Wilpon and Rabiner	LPC coefficients	Itakura distance	k -means	Cluster variance
Ernst et.al	Model profiles	Correlation coefficient	Model-based	Cluster variance
Shaw and King	Normalized spectra	Euclidean distance	Agglomerative, PCA	N/A
Vlachos et al.	Haar wavelet transform	Euclidean distance	k -means	Cluster variance
Goutte et al.	Cross-correlation	Euclidean distance	k -means	Cluster variance
Fu et al.	PIP	Squared distance	SOM	Squared error

Figure 2.4. Summary of feature-based algorithms [70][46][66][21][17].

Goutte et al. [22][21] paid their attention on analysis of special magnetic resonance images (MRI) which is called fMRI. They clustered the time series images based on their similarities of voxels by using two modified algorithm: k -means and Wards hierarchical clustering. They did not use raw time series images directly and instead they took cross-correlation function of fMRI activation and stimulus as the feature of data objects. Through the cross-correlation function, the value of the function is calculated in terms of measured fMRI time series and activation stimulus at different time, and the activation stimulus could be many factors such as square wave. In the following, they modified the same algorithm which is also feature-based clustering method. In the new methods, they decreased data dimensions by only extracted two features: significantly delays and strength of activation. They measured these two features voxel-by-voxel and believed that the different regions in the images could be identified through these two features. After the feature extraction, k -means algorithm was used to evaluate the performance of some information criteria which can be used to determine the clusters. In addition, this algorithm could be also used in voxel analyses by assessing their similarities and dissimilarities. Their algorithm is a standard feature-based algorithm since it extracted particular features instead of analyzing raw data directly. Although this algorithm

was developed specially for analyzing MRI time series data, it is still useful in developing new algorithms for microarray time series data.

Fu et al. [17] modified self-organizing maps by introducing one new term perceptually important point (PIP) to analyze data sequences and try to find their similar temporal patterns among data objects. Perceptually important point can reduce the dimension of input data sequence D accordance to query sequence Q . Both in D and Q , the distance is computed as sum of mean squared distance along the vertical scale and horizontal scale. The vertical scale and horizontal scales separately stand for magnitude and time dimension. By the way, training samples are used to figure out different resolution patterns in which SOM process was used once only. Generally, there are two main modifications of the new method on SOM method: firstly, filtering out nodes or patterns in output layer did not considered in recall process; secondly, taking a redundancy removal step when adding a general pattern for consolidating discovered patterns during clustering.

2.3.3. *Model-Based Approaches*

The third category of clustering algorithms and procedures on time series data is called model-based approaches since some kind of model or underlying probability distributions are come out to generate time series. Time series are considered equally in which time series have been characterized and time series are still unmarked. Model-based approaches can handle time series data with both equal length and unequal length data series and with those model-based approaches using log-likelihood distance measure, the highest likelihood is concluded for the clustering analysis. Ramoni et al. [57][56] developed an algorithm called BCD based on Bayesian method for clustering dynamic data. This algorithm is suitable for discrete-valued, not continuous-valued time series data. Assume we have a data set S with n number of data objects in it, the model BCD will transfer each time series into Markov chain (MC) and group MCs into same clusters based on their similarities. The model BCD belongs to unsupervised agglomerative clustering approach. The process of clustering is considered to be Bayesian model selection problem and when partitions generated in the

process, they treated each partition as a hidden discrete variable C , each variable C_k for each state means one cluster of time series and then generate transition matrix. The maximum posterior probability is computed for each data to make sure each data object will be compared to all models and all models are treated equally. The comparison between data and models can base on marginal likelihood. It measures the likelihood of data objects under the situation that if model MC is true. The parameter symmetrized Kullback-Liebler distance is calculated as the similarity between two transition matrices, and the distance is according to corresponding rows in the matrices. When clustering, loss of data information based on different clustering methods is the evaluation of final clusters. They also developed the algorithms for multivariate time series, which measure the similarity of Kullback-Liebler distance so it is heuristic search method based on similarity to look for most probable set of clusters. The distance is between two comparable transition matrices. In this modified algorithm, it used similarity measure instead of grouping criteria as a heuristic guide. The posterior probability of each obtained clustering is the basis of grouping and stopping criteria. For this model-based method, it tries to find a maximum posterior probability for each in according to each set of Markov chain. Joseph et.al [43] presented an algorithm for time series gene expression data. There are three functions of this model-based approach: alignment of gene expression datasets; estimation of missing experiments and clustering. They treated each gene expression as one profile and built a cubic spline as the relative model. The cubic spline is a piecewise polynomial that came from the observed data and number of time points and it influences the shape of gene expression shapes. Their algorithm can also estimate the unobserved time points in gene expression data so that it did not depend on the number of time points by measuring sampling rates. This method used a continuous warping representation to avoid temporal discontinuities and can handle low resolution datasets on gene expression. This algorithm can also work on the gene expression data without giving the class representation. In the clustering, the model would generate expression synthetic curves for each gene and estimate the expression values for each gene and then group genes into clusters. The authors compared their algorithm to k-means algorithm to show their algorithm can identify the detailed expressions much better.

Piccolo [54] concerned Euclidean distance as the metric related to autoregressive expansions. The metric can be considered to be distance since it has the basic properties of distance including non-negativity, symmetry, triangularity, etc. For each pair of time series models, the distance matrix was

used to build dendrogram by applying complete linkage clustering method. Some people also developed model-based algorithm by modifying the traditional agglomerative hierarchical clustering algorithms. Maharaj [47] came up with analysing p-value in the hypothesis test for each pair of stationary time series data. This method introduced a linear AR(k) model to evaluate each stationary time series data. The model can be expressed as the vector of parameters $\pi = [\pi_1, \pi_2 \dots \pi_k]$. Before the hypothesis test, the null hypothesis is that there is no difference between two stationary time series data ($H_0: \pi_x = \pi_y$) and then chi-square test is executed. There is a pre-specified significance level on p-value, and two time series data will be grouped into one cluster if their p-value is both greater than the pre-specified p-value. Finally, the clustering results are double checked by difference between actual number of final clusters and the number of generated clusters which are exactly correct, and this kind of difference is called discrepancy. Li and Biswas [41] use the spirit of hidden Markov model representation to analyse time series data. There is one assumption on the time series data that the data should have Markov property or the data are resulting from indirectly observable states. Their HMM clustering method took advantage of nested searches at four levels from outer most to inner most: 1) partition is the subset of all data objects and there are many clusters in one partition; the first search level is the number of clusters in each partition according to partition mutual information measure; 2) the second level is the structure for a given partition size in respect to depth-first binary divisive clustering and k-means. At this level, the object-to-model likelihood distance measure was with object-to-cluster assignments; 3) the third level is HMM structure for each cluster with highest marginal likelihood. For this level, the HMM structure starts with initial model configuration and then increases or decreases the model based on HMM state splitting and merging operations; 4) considering the segmental k-means method, the parameters of each HMM structure. In this model-based algorithm, three random generative models in which there are three, four and five states are taken to create man-made data set, and by using correct model size and suitable model parameter values this algorithm can reconstruct HMM. In the following, Li et al. [42] developed a modified model-based algorithm called Bayesian HMM which used Bayesian information criterion (BIC) as the selection criterion in level 1 and 3, at the same time, a sequential search was built. The algorithm starts with simplest model and then gradually increases the size of model and stop when one BIC score is less than the score of previous

model. The authors apply artificial generalized data and ecology data to certificate the effectiveness of the algorithm.

Authors	Model used	Distance measure	Algorithm	Criterion
Ramoni et al.	Markov chain	Symmetrized Kullback-Liebler search and Posterior probability	Agglomerative clustering	Marginal likelihood
Joseph et al.	Cubic spline	N/A	Partitioning	Coverage for random center
Li and Biswas	HMM	Log-likelihood	Nested search at four levels	Partition mutual information
Maharaj	ARcoefficients	N/A	Agglomerative clustering	Number of dwelling units
Piccolo	AR	Euclidean distance	Agglomerative clustering	N/A
Wang et al.	Discrete HMM	HMM parameters	Log-likelihood	Log-likelihood
Tran and Wagner	Gaussian mixture	Log-likelihood	Fuzzy c-means	Variances in clusters
Baragona	ARMA	Cross-relation	Tabu search	N/A
Oates et al.	Discrete HMM	HMM parameters	Log-likelihood	Log-likelihood

Figure 2.5. Summary of model-based algorithms [57][41][68][62][53][47].

Wang et al. [68] proposed discrete hidden Markov models in a machining process. This method was similar to feature-based approaches but it is in model-based category. They extracted features in the data obtained from wavelet analysis which were vibration signals. The exacted features are stored in vectors and then were converted into symbol sequences, and then the symbol sequences are taken as input as training hidden Markov model to look for clusters.

Among all the model-based clustering approaches, hidden Markov model is most commonly considered like said above. Oates et al. [53] came up with a hybrid clustering method which can determined the k number of HMMs and generated related parameters automatically. They took advantage of agglomerative clustering algorithm to estimate k and obtain the original clusters for dynamic data. The generated original clusters was the input for training the model HMM on each cluster and then put the clusters together based on their likelihoods presented by various HMMs iteratively. Tran and Wagner [62] modified a fuzzy c-means clustering-based method by applying

Gaussian mixture model to find good scores for speaker verification. This algorithm avoided equal weights of all likelihood values for speakers in background among normalization methods.

Among those model-based algorithms, modelling part is various dependent on different prerequisites, parameters and final goals. Markov chain and hidden Markov models are used in many model-based methods since they can handle time dimension. As the rapid growth of time series data, the model-based approaches will be developed to adjust to figure out more complicated questions on clustering process.

CHAPTER 3

OVERVIEW ON ONE CLUSTERING ALGORITHM

In chapter 2, we have reviewed some clustering algorithms; most of them are implemented without microarray data, thus we can try to implement them in this aspect. Comparing to other time series data, microarray data are more complicated since they are generated from organisms which are most complicated systems. In order to handle microarray data, clustering analysis is taken for solving comprehensive biological problems, like pathway construction, discovering new genes, new drug design [6], etc.

In this chapter, we will introduce one greedy algorithm designed for short time series gene expression data in which time points is less than 8 [14]. Here profiles are a set of gene expression data correlated in different time points for each gene. There are two major steps for the algorithm: the first step is predefining a set of model profiles which can represent the potential distinct patterns among all expression data; and the second step is to group other general profiles into the predefined model profiles. Generally, the algorithm starts with picking some potential profiles as model profiles which can represent the trends of other gene expressions; and then computes the distances of each pair of profiles, assign genes into profiles and group the profiles into clusters.

3.1. Preprocessing Raw Data

The raw gene expression data are preprocessed by log ratio discrepancy. Log ratio discrepancy is a processing method commonly used in microarray data. Every time point will be transformed with respect to the first time point value so that each time point values are divided by the first time point value, and then they are taken to logarithm transformation. Consequently, every time point value are processed as log-ratio and the first time point value will be always 0.

3.2. Selecting Model Profiles

The first step is selecting several model expression profiles. People need to define a parameter c in advance which can control the amount of change of gene expression between two successive time points. This parameter assumes that one gene expression value will not change beyond c units compared to its previous time point. For example, if $c = 1$ then the successive time point value can

go up or go down one unit or stay at the same value; and if $c = 2$ then the successive time point value can go up one or two units, go down one or two units, or stay at the same value. Thus, for n time points, there will be $(2c + 1)^{n-1}$ distinct profiles. The value of c greatly control the amount of model profiles. For example, if $c = 1$ and the number of time points is 5, there will be $3^4 = 81$ model profiles which will be selected for the whole gene expression time series data. As the value of c increases, the number of model profiles increase dramatically.

After defining the parameter c , this algorithm will create two sets for model profiles and all profiles. It assumes that m representative profiles are of interest by users. Let set P contains all the $(2c + 1)^{n-1}$ profiles and let a set $R \subset P$, and the number of profiles in R which are selected as model profiles meets ($|R| = m$). Among the R , the minimum distance of each pair of model profiles has been maximized. The distance can be expressed as the following formula with respect to correlation coefficient (d is distance metric):

$$\max_{R \subset P, |R|=m} \min_{p_1, p_2 \in R} d(p_1, p_2)$$

Distances are calculated between profiles in the set R . The $b(R)$ is considered to be the minimum distance between model profiles which are in R , and the ideal state of the minimum distances between two profiles in R are maximized. Here it is necessary to explain a little bit on the two definitions of distances here. The minimum distance between model profiles in R means the “real” distance between two profiles separately in set P and set R , and the maximum distance means the algorithm would select one distance which has largest value among all distance to make sure all model profiles in R would be most distinct from each other.

The original algorithms assumes that R' is the a set of profiles that is the optimal object solution for the above equation and $b(R')$ are the optimal distances which are maximized for all the minimum distances. However, the original algorithm claims that maximizing the above equation and finding such a set R' or even $b(R')/2$ is a NP-Hard problem. In this situation, it is assumed that the set of R can achieve the $b(R')/2$, and a greedy algorithm to find such a set which is equal or worse than $R/2$ was implemented. The result set can be marked as R which $b(R) \gtrsim b(R')/2$. This greedy algorithm starts with some extreme profiles which are often one or two types and always going down


```

procedure SelectVectorsMaxMinDist( $d, P, m$ )
  let  $p_1$  be the profile that always goes down one unit between
    time points
   $R = \{p_1\}$ ; ( $\star$  The set of selected vectors  $\star$ )
   $L = P \setminus \{p_1\}$ ;
  for  $i = 2$  to  $m$  do
    let  $p \in L$  be the profile that maximizes:
       $\min_{p_1 \in R} d(p, p_1)$ ;
     $R = R \cup \{p\}$ ;  $L = L \setminus \{p\}$ ;
  end for;
  return  $R$ ;

```

Figure 3.1. Greedy algorithm to choose the model profiles.

for each time points based on the value of c . When the first model profile has been selected, the set R is built and other profiles will be added accordingly that the model profiles in the set will be most distinct from each other. The following profiles added into set R should meet the following equation (d is distance metric):

$$\max_{p \subseteq (P \setminus R)} \min_{p_1 \in R}$$

When picking model profiles, each profile in will be selected, and the distances between this profile and all profiles already in R will be calculated, among all the distances, the smallest distance will be picked as the real distance between this profile and set R . Same as above, all the smallest distances for each profile in set P will be calculated and among all the smallest distances, the profile with largest distance will be added into R . At the same time, the profile with largest smallest distance between itself and set R will be deleted from set P . The stop point is defaulted by users by setting a parameter m ; for example, when the number of model profiles in R reaches to $m = 50$, the iteration will end and there will be 50 distinct model profiles representing the whole gene expression profiles. The pseudocode of this algorithm is like the following:

This algorithm has m iteration and each of the iteration take $m(2c + 1)^{n-1}$ time and for the time complexity, it will be $m^2(2c + 1)^{n-1}$, thus, this algorithm needs a small m for model profiles and small n for time points.

3.3. Grouping Significant Profiles

The second step is to add profiles to model profiles in R and then generate clusters. One threshold value δ is predefined to judge whether two profiles are similar and will group two similar profiles into the same cluster. A graph is created as (V, E) ; V is the model profile and E is the edges. $v_1, v_2 \subset V$ if the edge $d(v_1, v_2) \leq \delta$. Then this graph is partitioned into small cliques; each clique stand for a set of significant profiles. The algorithm will check the distance between model profiles and general profiles to generate cluster C_i . It assumes that the p_j is the closest profile to p_i which is not already included in the cluster. If $d(p_j, p_k) \leq \delta$ for all profiles in C_i , then this profile p_j will be added into cluster C_i , otherwise, the profile will be ignored. The whole process will be repeated until obtaining clusters for all significant model profiles. Next, comparing all the clusters to each other and then pick out the one profile which has largest number of profiles as the first cluster. As for other clusters which do not have largest number of profiles, in which the general profiles will be released back to the set P . This procedure will be repeated to select other profiles and will stop when all model profiles have been assigned to clusters.

3.4. Problems on This Algorithm

This greedy algorithm are based on raw data by treating each gene expression level one profile; it can efficiently separate objectives efficiently with short time points. However, there are some shortcomings for it and we listed two problems on its preprocessing step and grouping step.

The first problem is the preprocessing step, that the log ratio discrepancy was used for preprocessing gene expression raw data. Log ratio is often used in preprocessing raw data including microarray data since it does not change the relative gene expression values[24]. The original preprocessing method in the algorithm is taking log ratio discrepancy with respect to the first value of raw data; it means every raw value will be divided by the first value and then be taking log, thus that the first value for each gene will be always 0. However, let's assume two cases with three time points: the first gene expression level is $[10, 10, 10]$ and the second gene expression $[3000, 3000, 3000]$, and when we execute log ratio on these two gene expressions, we will get the first set as $[0, 0, 0]$ and the second set as $[0, 0, 0]$. The two gene expressions are the same after log ratio discrepancy preprocessing and it cannot identify which gene is fully expressed or non-expressed at all. For the second case we assume

that the first gene expression is [2, 20, 200] and the second gene expression is [100, 1000, 10000], the normalized sets are separately [0, 1, 2] and [0, 1, 2]. The two expression levels will be considered the same pattern and then will be grouped into the same clusters. Even the changing patterns of these two gene expression levels are the same, the high or low levels of one gene expression cannot be identified.

The second problem of the algorithm is at the grouping step. The algorithm generates several clusters for each iteration, and then keep the cluster with largest number of profiles and release the profiles in other clusters. This process will be repeated several times until each one of model profile has been added general profiles. This process would take time and cannot make sure every profile be grouped into one model profile. Furthermore, the algorithm will let the users select one threshold value δ for similarity judging purpose. When the value of distance between two profiles is smaller than value of δ , the corresponding general profile would be grouped into the cluster with the model profile; otherwise it will be discarded. Thus, to select appropriate δ every time is decisive in order to avoiding problems that all profiles are grouped into one cluster if δ is too small or all profiles cannot be grouped into any cluster if δ is too large. However, to select such a suitable δ is time-consuming since the value of δ depends on different time series data.

CHAPTER 4

MODIFICATION ON THE GREEDY CLUSTERING ALGORITHM

In this chapter, we introduced two improvements on the algorithm to avoid the above two problems we mentioned in chapter 3. We carefully checked this greedy algorithm and implemented it by Matlab 2010b by using simulation data and real microarray data downloading from Stanford Microarray Data (SMD:<http://smd.stanford.edu/>). The species we selected for evaluating the algorithm is *Helicobacter pylori* wild type G27 under trial 4. And we selected 24192 genes expression levels as the objectives in the clustering analysis. We also compare the results from the original algorithm and the modified algorithm.

The first improvement is introducing scaling preprocessing method on gene expression microarray data. Assume the above two cases: when the first gene expression level is [10, 10, 10] and the second gene expression level is [3000, 3000, 3000], and if we scale the raw data values down by unit 10 and get [1, 1, 1] and [300, 300, 300]; we still know these two gene expression levels are different. For example, we will see the differences between these two gene expression levels when plotting on the figures; as for the second case that the first gene expression level is [2, 20, 200] and the second one is [100, 1000, 10000]. When we scale the raw data down by 10, we will get [0.2, 20, 200] and [10, 100, 1000] and then we still know they are different levels. Above all, scaling down method can avoid the two problems we mentioned above. In addition, we take rounding values after scaling down to make the preprocessing step discrete in the later analysis.

The second modification we made is the grouping strategy. The original algorithm generates several clusters for one iteration, and then keep the cluster with largest number of profiles and release the profiles in other clusters. This process will be repeated several times until each one of model profile has been added general profiles to generate clusters. Instead, we take a grouping step to classify each general profile into exact cluster based on their scores to different model profiles. It means we directly group one general profile into its nearest model profile.

The major shortcoming of the original algorithm is its big time complexity is $O(n^4)$, and it means it should be very time consuming when time points increased. We modified the grouping step and the time complexity is only $O(n^3)$, and every general profile would be considered to assign to

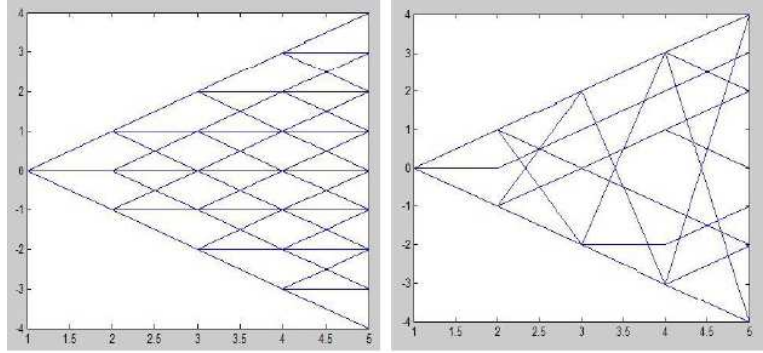


Figure 4.1. The simulated gene expression profiles we made, each value at one time point will go up or down c units. The left figure showed all profiles in the model which the number of profiles is 81 when $c = 1$ and time points is 5; the right figure showed that the model profiles we generated and here $m = 10$

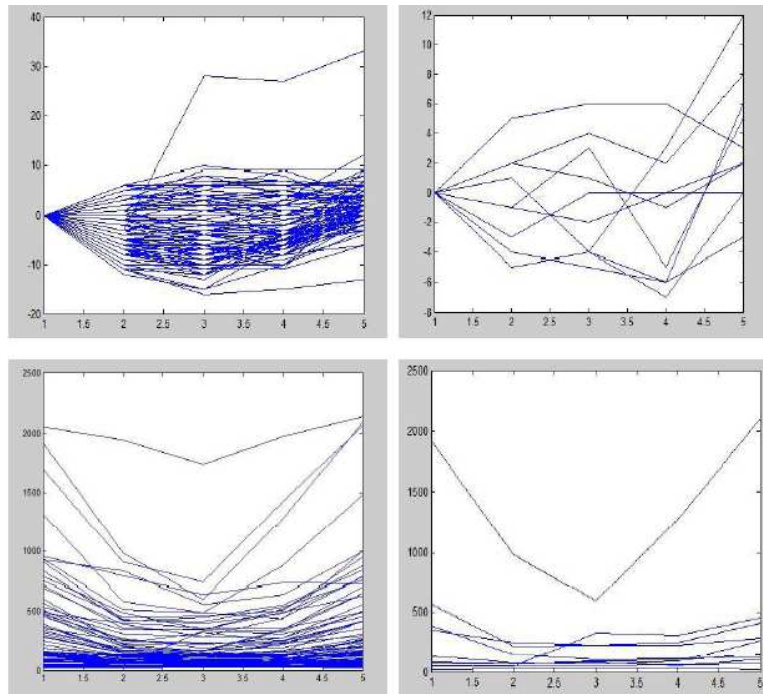


Figure 4.2. The two types of preprocessing methods on raw data which are 100 gene expression profiles here. The up left figure is the plot diagram of all the 100 profiles after log ratio preprocessing; the up right figure shows 10 model profiles selected from log ratio discrepancy; the down left figure is all the 100 profiles with scaling down preprocessing; and the down right figure is about the 10 model profiles with scaling down preprocessing. The dataset is from species *Helicobacter pylori* and so there are no above extreme situations in this experiment. And the two results to obtain the same 10 gene expression profiles as model profiles.

one cluster based on its scores to different model profiles. This is also greedy algorithm to put gene expression profiles into different clusters.

In order to compare the modified algorithm to original algorithm, we downloaded 24192 gene expression profiles on *Helicobacter pylori* from SMD. By using perl scripts, it is found that there are no

Table 4.1. The 10 model profiles by using Log ratio.

Profiles	Gene name	Time point 1	Time point 2	Time point 3	Time point 4	Time point 5
Profile 1	zinc finger protein, X-linked	10713	7487	6916	6811	8722
Profile 2	ESTs	8872	7115	7378	5826	5318
Profile 3	ATPase, Ca ⁺⁺ transporting, plasma membrane 1	2567	3775	3527	2274	3555
Profile 4	ESTs	10165	13406	12608	10870	6690
Profile 5	Homo sapiens cDNA FLJ37163 fis, clone BRACE2026971	869	1547	1084	835	2958
Profile 6	apoptotic chromatin condensation inducer in the nucleus	375	745	986	1187	580
Profile 7	v-ski sarcoma viral oncogene homolog (avian)	791	799	833	1242	907
Profile 8	Unnamed	5967	5636	6591	6520	5250
Profile 9	bone marrow stromal cell antigen 1	575	711	945	1125	832
Profile 10	ESTs, Weakly similar to hypothetical protein FLJ20489 [Homo sapiens] [H.sapiens]	936	961	1400	832	1078

extreme cases we mentioned above. We consider that there are two kinds of differences between the original algorithm and modified algorithm, but in the data analysis we only took log ratio pre-processing in this thesis since it is widely used in microarray data normalization. Thus, the only difference between the original algorithm and modified algorithm is from the grouping step.

By implementing the original grouping algorithm on the 24192 gene expression profiles, we can get two clusters when set the $\delta = 0.7$, in which the model profiles are separately 12828 and 10350 profiles. And the corresponding genes for the model profiles are *ATPase, Ca⁺⁺ transporting, plasma membrane 1* and *ESTs, weakly similar to hypothetical protein FLJ20489 [Homo sapiens] [H.sapiens]*

Table 4.2. The 10 model profiles by using Scaling.

Profiles	Gene name	Time point 1	Time point 2	Time point 3	Time point 4	Time point 5
Profile 1	zinc finger protein, X-linked	10713	7487	6916	6811	8722
Profile 2	apical iodide transporter	345	840	323	454	2555
Profile 3	ESTs	2171	1199	1184	1619	2172
Profile 4	meningioma expressed antigen 6 (coiled-coil proline-rich)	301	332	429	695	229
Profile 5	lipase, hepatic	311	452	284	395	20660
Profile 6	Homo sapiens cDNA FLJ25541 fis, clone JTH00915	14238	1217	1392	1322	1298
Profile 7	ESTs, Weakly similar to hypothetical protein FLJ20489 [Homo sapiens] [H.sapiens]	936	961	1400	832	1078
Profile 8	thioredoxin interacting protein	359	5136	4595	2770	1715
Profile 9	solute carrier family 15 (H ⁺ /peptide transporter), member 2	984	1057	1562	1258	1407
Profile 10	Sp3 transcription factor	326	711	391	801	347

There are still 1024 profiles which can not be grouped into any clusters. It is reasonable that not all profiles will be assigned into clusters since the iteration for adding general profiles is controlled by the number of model profiles. And when all model profiles have been analysed, the rest of profiles will left into the matrix. When we change the value of threshold score (δ) between the each general profile and model profile, the number of profiles left in the matrix will accordingly changes.

On the contrast, when we implemented the modified grouping algorithm, the result contains 4 clusters. the first two clusters are big with 6368 and 5812 profiles with the genes for model profiles are separately *solute carrier family 15 (H⁺/peptide transporter), member 2* and *ESTs*. The smallest

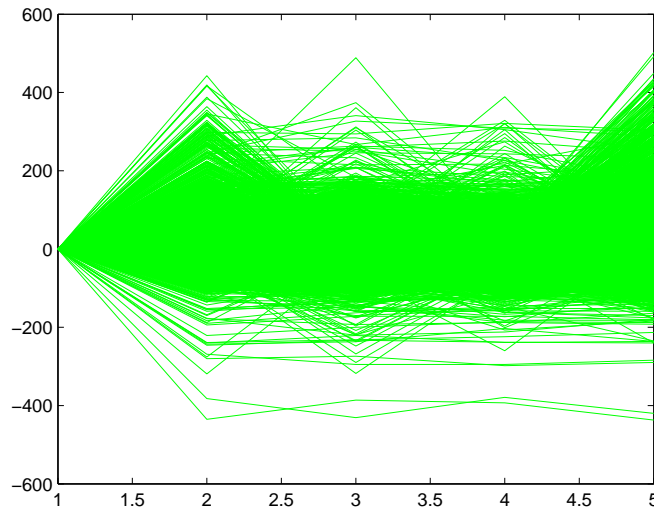


Figure 4.3. The plot diagram for 24192 gene profiles preprocessed by log ratio. The first value for each profile is always 0.

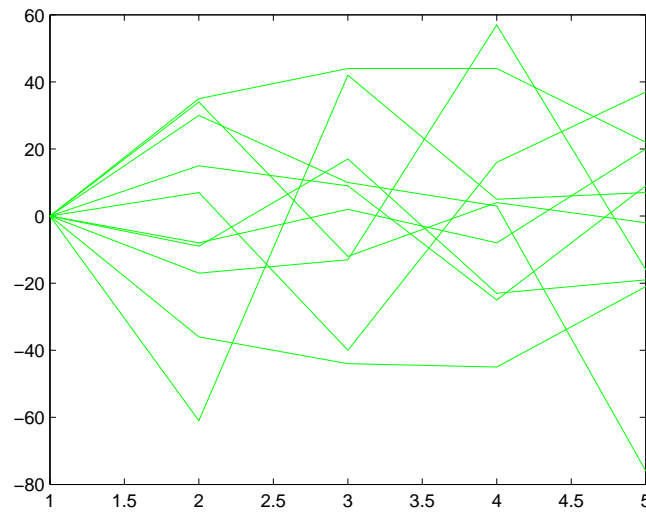


Figure 4.4. The plot diagram for 10 model profiles for 24192 gene profiles preprocessed by log ratio. The first value for each profile is always 0.

two clusters have only 1 profiles with genes for model profiles are separately *lipase*, *hepatic* and *zinc finger protein, X-linked*. For the modified algorithm, every profile will be grouped into one cluster.

The final clusters generated from two algorithms are compared. Firstly, the model profiles in each pair of two cluster are different. In addition, the number of largest and second largest clusters are different. Based on the information above, the profiles selected as model profiles are distinct to

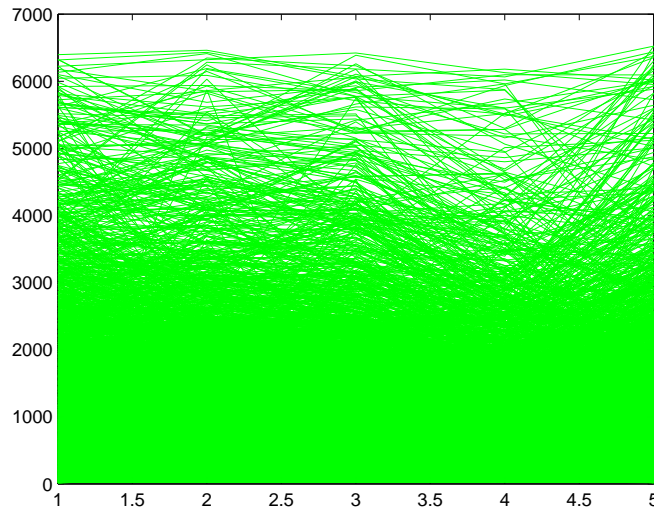


Figure 4.5. The plot diagram for 24192 gene profiles preprocessed by scaling down method. The scaling unit is 10.

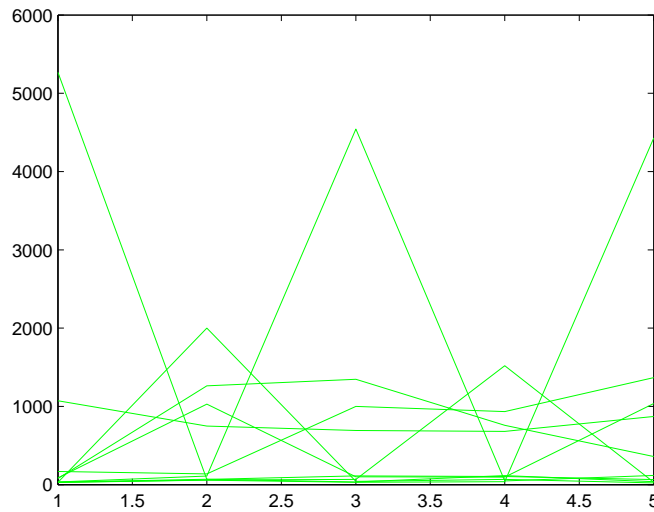


Figure 4.6. The plot diagram 10 model profiles for 24192 gene profiles preprocessed by scaling down method. The scaling unit is 10.

each other, so the cluster should be distinct. The possible reasons for this may be that the original algorithms released profiles and it only collect the largest cluster at each iteration, and consequently, some general profiles cannot assigned into the cluster which contains its nearest model profile because the nearest model profile had been taken out. Furthermore, the modified algorithm ran much faster than the original algorithm since its time complexity is lower.

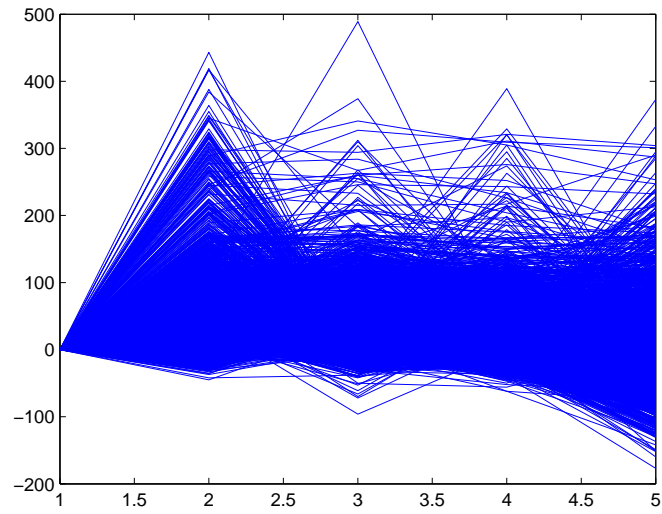


Figure 4.7. The plot diagram for largest cluster with 12828 gene expression profiles by implementing original grouping algorithm. The threshold value is 0.7. The corresponding gene for model profile is *ATPase, Ca++ transporting, plasma membrane 1*

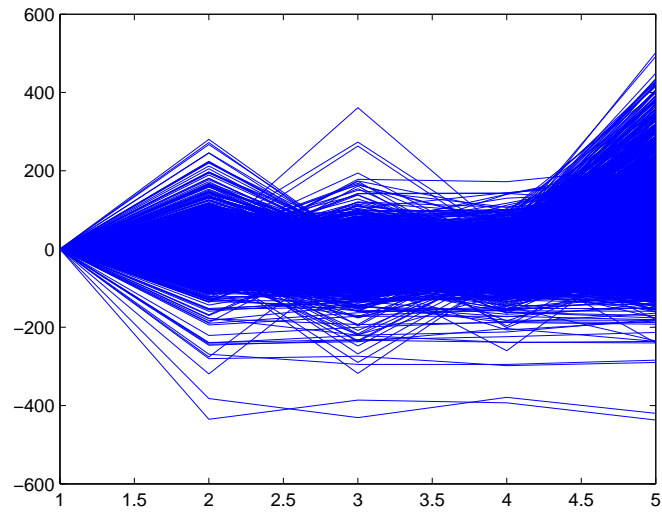


Figure 4.8. The plot diagram for second largest cluster with 11354 gene expression profiles by implementing original grouping algorithm. The threshold value is 0.7. The corresponding gene for model profile is *ESTs, Weakly similar to hypothetical protein FLJ20489 [Homo sapiens] [H.sapiens]*

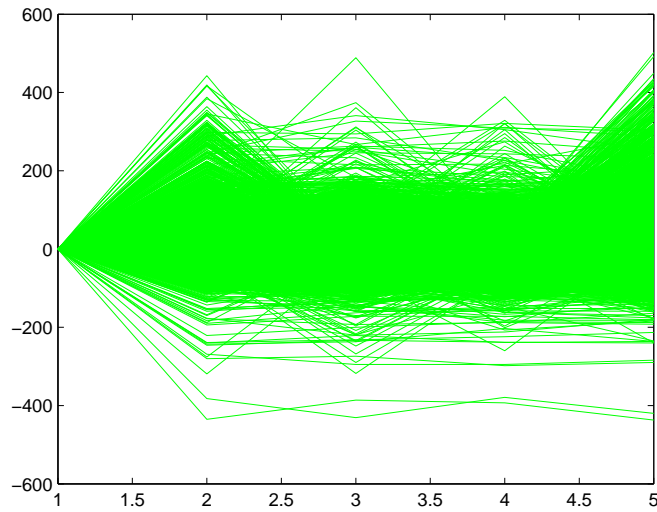


Figure 4.9. The plot diagram for largest cluster with 6368 gene expression profiles by implementing modified grouping algorithm. Every profile will be calculated to all model profiles and be assigned to the nearest one. The corresponding gene for model profile in this cluster is *solute carrier family 15 (H⁺/peptide transporter), member 2*.

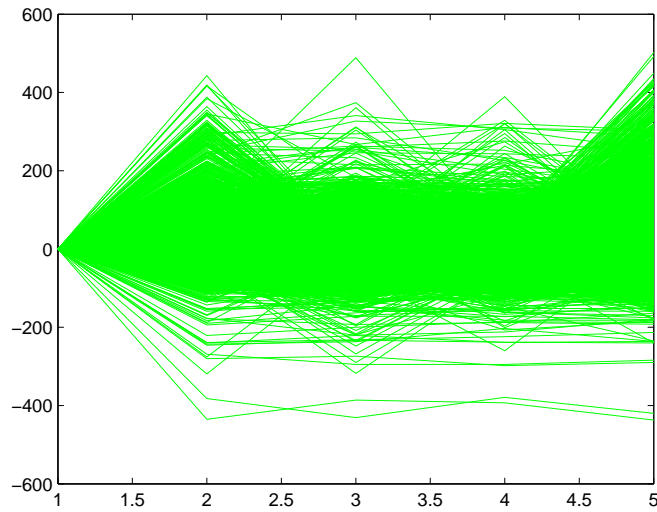


Figure 4.10. The plot diagram for the second largest cluster with 5812 gene expression profiles by implementing modified grouping algorithm. Every profile will be calculated to all model profiles and be assigned to the nearest one. The corresponding gene for model profile in this cluster is *ESTs*.

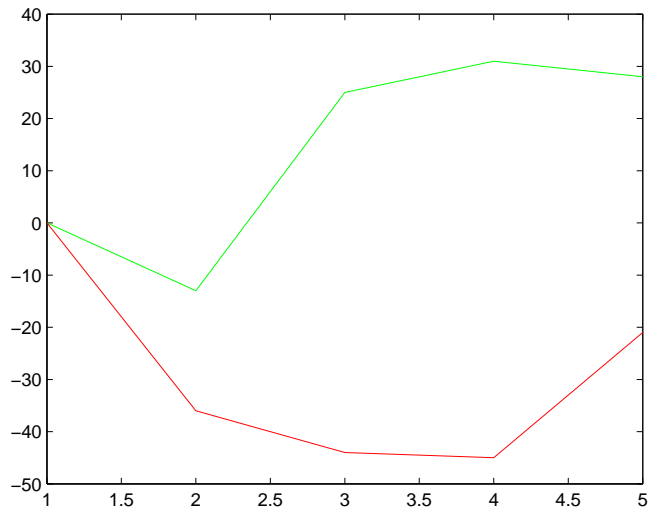


Figure 4.11. The plot diagram for the cluster with only one gene expression profile by implementing modified grouping algorithm. The red line is the model profile gene *zinc finger protein, X-linked*.

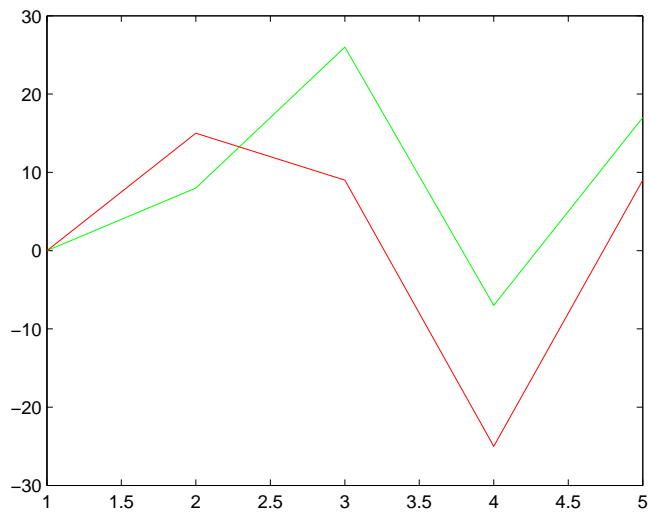


Figure 4.12. The plot diagram for the cluster with only one gene expression profile by implementing modified grouping algorithm. The red line is the model profile gene *lipase, hepatic*.

CHAPTER 5

CONTROL SYSTEM-BASED CLUSTERING ANALYSIS

In this chapter, we introduce a new clustering algorithm for time-course microarray data from a system viewpoint. Specially, we take advantage of the fact that a significant amount of time-course microarray data shows gene expression changes in response to disturbances. For example, in order to discover posttraumatic stress disorder (PTSD) pathways, battlefield-like stress conditions are applied to target mice and as the disease develops over time, mouse brain samples are collected for microarray hybridization; the microarray data obtained across time reveals gene expression changes in response to the stress condition [59]. The viewpoint of time-course microarray data as system responses is insightful, since system responses generally have characteristics that are distinguished from random point series/curves. Typically, a response subject to a stimulus goes through a transitional stage and finally reaches a stable equilibrium. As an example, one of most acquainted system responses is the response of a 2nd-order LTI system subject to a step input. The response reaches its peak value and then oscillates before settling at the steady state value.

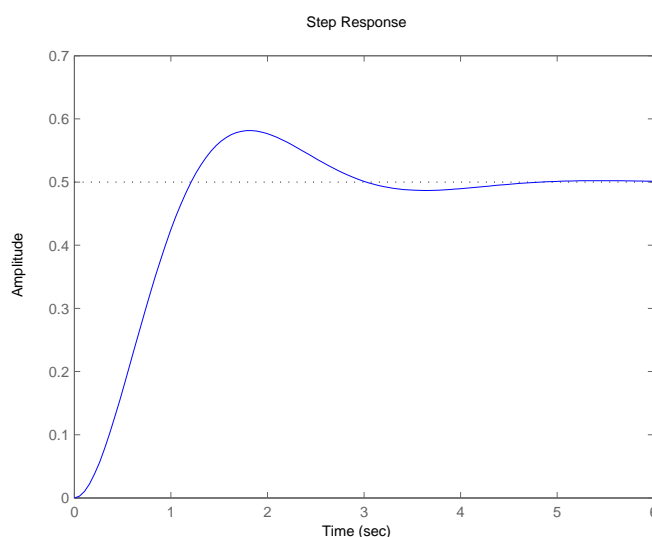


Figure 5.1. A standard 2nd-order LTI system.

The characteristics of system responses motivate us to group genes according to the most significant temporal features that can best capture a system response, such as rise time, peak value,

settling time, overshoot, among others. We recall that such measures are widely used in control-system applications to measure the performance of a control system. For a low-order linear time-invariant system, such measures could completely reveal the internal system. Even for high-order/nonlinear systems, such measures are still capable of capturing the important temporal information of a system response, and in turn suggest the underlying system because high-order/nonlinear system responses can be approximated by one or several low-order systems' responses [74]. For example, the settling time of a gene expression response is informative of the slow eigenvalues that are observed at the gene, and the rising time has to do with the local observable fast eigenvalues. By searching for these features from response data, we are able to capture the key characteristics of each response dynamics, and then perform partitioning method to group genes based on these temporal features.

For our feature-based clustering algorithm, we treat the whole microarray experiments as feedback systems and the treatment on the experiments like heat shock as disturbances. Feature selections for feedback systems were executed in order to describe their similarities. Among the step response, we select five features to represent each system: 1) steady-state error: the difference between ideal output and real output when the system tend to a steady state; 2) overshoot: the parameter when function exceeds its target and it is often considered together with damping ratio; 3) peak time: the time which measures from beginning to the first peak of the system; 4) settling time: the elapsed time for the function from instantaneous step input to an error band; 5) rise time: the time for the function which changes from 10% value to 90%value of the step height.

Firstly, we created the simulation data to implement our clustering algorithm. The coefficients of numerators and denominators of transfer functions are fixed and then a set of random 2nd order systems by generating random transfer functions which are mathematical representations for temporal frequencies. As the Figure 5.2 showed, each plot represents one system that we created for feature selection.

The feature selection have been done after the generation of linear time-variant systems. Five features, including steady-state error, overshoot, peak time, settling time and rise time, are selected to describe the characteristics for each system. Five features for one system are stored as a vector and then all vectors are stored into one matrix. Each feature is given different weights, and among the five features, we assume that steady-state error and overshoot contribute more on identifying

systems, thus we give much heavy weights on these two features to test the clustering results. For the features steady-state error, overshoot, rise time, settling time and peak time, we separately give the weights of 0.5, 0.2, 0.1, 0.1, 0.1 and 0.2, 0.5, 0.1, 0.1, 0.1.

After giving the weights to the five features, distance between pair of vector are calculated. We selected Euclidean distance which measures distance of two points in coordinates. Based on the Euclidean distances, we selected k means algorithm to generate clusters. As we discussed in chapter 2, k means algorithm is often used in clustering analysis. We also did a multidimensional scaling analysis in order to draw the systems on the two dimensional coordinates, since our data which are with five features are five dimensional.

The result shows that our feature-based algorithm working efficiently on grouping systems into different clusters. Each random system have been separated to the three clusters. As the number of systems increases, we can accordingly increase the number of clusters predefined before we execute the k means step. In addition, the cluster elements are different by giving different weights to the features; in our work, two kinds of clusters are generated, separately by giving more weights to steady-state error and overshoot.

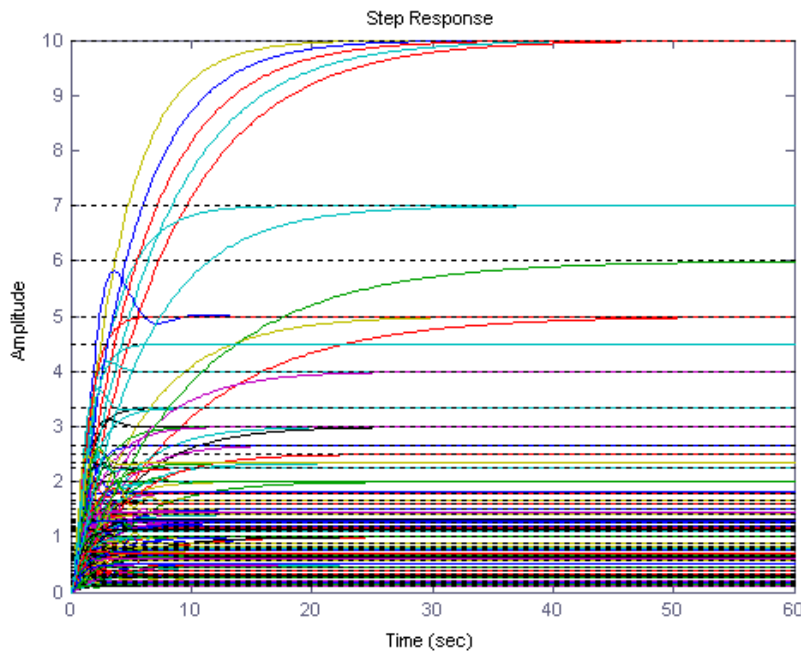


Figure 5.2. Linear time-variant systems generated in our algorithm

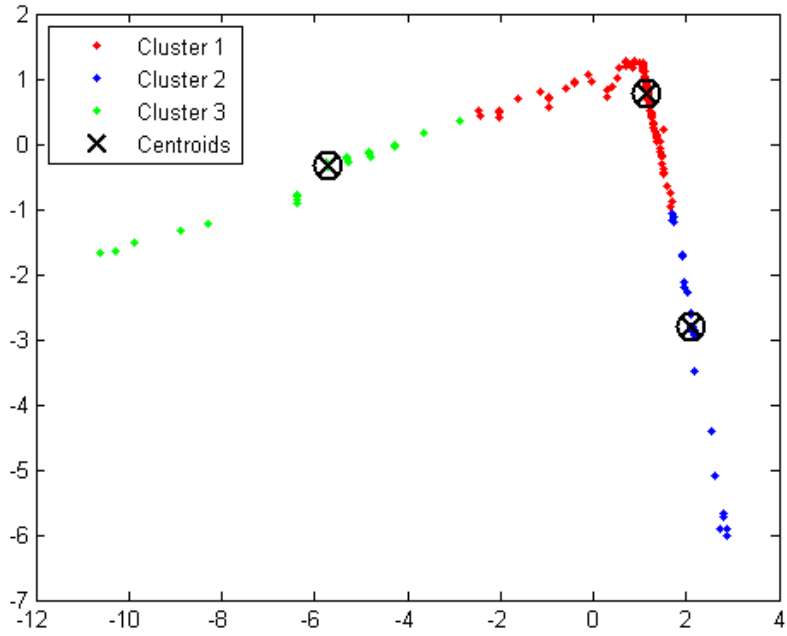


Figure 5.3. Three clusters by giving more weights on steady-state error, three clusters have similar number of systems.

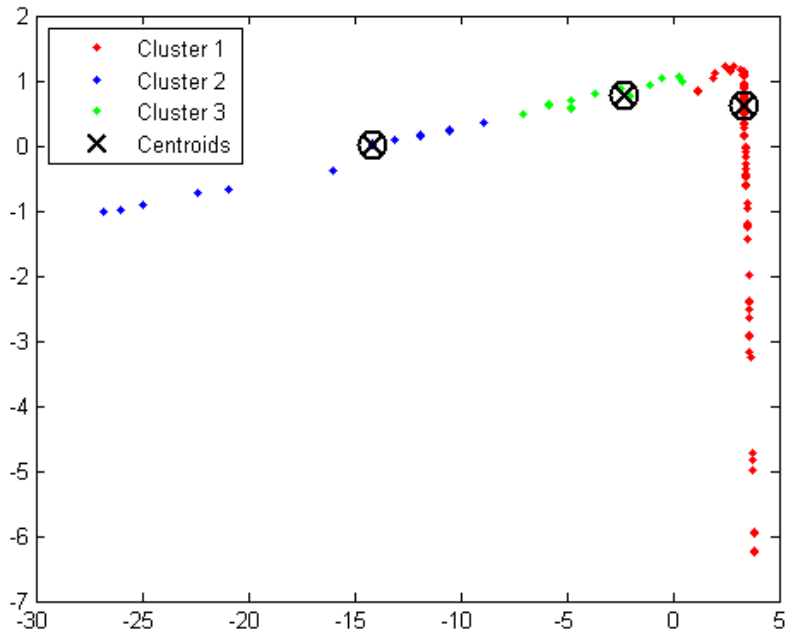


Figure 5.4. Three clusters by giving more weights on overshoot, three clusters do not have similar number of systems.

CHAPTER 6

DISCUSSION AND FUTURE RESEARCH DIRECTION

As the development of technologies in microarray and related disciplines like parallel processing, data warehousing and statistical methodologies, the depth and width of microarrays are becoming strong. Time points in each microarray experiment have been increased dramatically. Possibly the perceptive idea on designing corresponding algorithms and procedures should be aimed to handle long time series experiments and large scale information in the data. For example, more and more gene expression levels can be detected in each array and there are much longer time points in one experiment than before. Thus, it is necessary to develop new approaches or modify some existing methods which fit to short time points, small scale information or estimate hidden time points for gene expressions, to meet the requirements of microarray data development in the aspect of longer time points and large scale information.

Clustering methods can be divided into three groups: raw-data-based approaches, featured-based approaches and model-based approaches. Our interests are those algorithms which can take microarray data as the input. In the first part of this thesis, we implemented the profile-based algorithm on gene expression time series data. This original algorithm is for short time points which are less than 8 observations in microarray experiments and its efficiency and accuracy will not be satisfied when time points become high. The two improvements we made on the normalization for raw data and grouping strategy has increased its efficiency and accuracy. By comparing results from the original and modified algorithms, the modified algorithm is much optimal in the aspects of time complexity and structure than the original algorithm, thus it is much better to handle microarray data.

Several situations should be considered for our modified algorithm. The first situation is to select suitable scaling unit; in our modified algorithm, the raw data are divided by an unit and rounded to an integer and then the clustering analysis is based on the pre-processed data. The raw data would not be cut off dissimilarities if small unit numbers are selected, in other words, the microarray data cannot be scaled down by using small scaling units. On the other hand, some data values will be rounded to zeros if large scaling unit are selected. Thus, it is necessary to explore an efficient method to choose a set of scaling units such as the average of largest and smallest data records.

The control system based method in this thesis can work efficiently and accurately in simulation data we created. We extract five features which are steady-state error, overshoot, rise time, settling time and peak time and these five features can represent control systems since the clustering results shows that each system has been separated and grouped. The step input is selected as the time behavior for microarray data to systems. The five features represent the systems and each pair of systems are checked by calculating their Euclidean distance and the distance matrix are generated. Finally, the clusters are generated by implementing k means method in which the number of clusters can be defaulted before clustering analysis.

For the future work on the feature-based algorithm, one thing is that we need to calculate the weights for each feature; in our research we tried different weights on each feature and found that it results in different clusters; the weights measurement should be taken to calculate the weights for different features so that optimal clusters would be generated. Since there are not previous work to refer, the unsupervised weights measurements should be taken. One another issue we can carry on is the distance between every two systems; in our research the Euclidean distance is taken, which can distinct two vector with feature values; Euclidean distance is a basic trial distance and some other distances may be taken to represent similarities/dissimilarities between systems. Furthermore, the k means algorithm is used to generate final clusters, however, it is necessary to use different clustering methods like fuzzy c means and hierarchical methods to handle distance matrix and generate better clusters. What's more, we take step input as the time behavior for control systems, and it should be a good temptation to try some other inputs such as standard input and impulse input and the following features may represent systems much better. At last, real microarray data should be taken as input to test this algorithm and the corresponding results can be compared to other clustering analysis; for the algorithm in the thesis when taking real microarray data, linearization is necessary step for microarray data to be discrete to adjust the requirements of input for linear time-variant systems.

For both the modified algorithm and our new algorithm, it is necessary to detect the significance of generated results. For example, whether the generating clusters are significant; whether the features can well represent control systems; and as the change of shape of profiles - the change of time point values, which type of systems can be chosen to describe the data change. To solve these questions,

it is desirable to introduce some statistical approaches, e.g., hypothesis test, principal component analysis, linear discriminant analysis, to provide more evidence on these issues.

BIBLIOGRAPHY

- [1] A. Adams, G. Heller, A. Olson, J. Osborne, M. Karlsson, J. Nahalkova, L. Van Zyl, R. Sederoff, J. Stenlid, R. Finlay, and FO Asiegbu, *Comparative analysis of transcript abundance in pinus sylvestris after challenge with a saprotrophic, pathogenic or mutualistic fungus*, *Tree physiology* 28 (2008), no. 6, 885–897.
- [2] Can Alkan, Bradley P. Coe¹, and Evan E. Eichler, *Genome structural variation discovery and genotyping*, *Nature Genetics* 12 (2011), 363–376.
- [3] M. Ankerst, M. Breunig, HP. Kriegel, and J. Sander, *Optics: ordering points to identify the clustering structure*, *Proceedings of the 1999 ACM-SIGMOD International Conference on Management of Data*, Philadelphia, PA (1999), 49–60.
- [4] Aoki1990, *State space modeling of time series masanao aoki springer-verlag*, Book review- *Methods for system identification:in Econometric Theory* 6 (1990), no. 2, 263–267.
- [5] MH Asyali, MM Shoukri, O. Demirkaya, , and KS Khabar, *Assessment of reliability of microarray data and estimation of signal thresholds using mixture modeling*, *Nucleic Acids Res* 32 (2004), 2323–2335.
- [6] A. Banerjee, *Fuzzy clustering in drug design: Application to cocaine abuse*, *Fuzzy Information*, 2004. *Processing of NAFIPS '04. IEEE Annual Meeting* (2004), 27–30.
- [7] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini, *Clustering gene expression patterns*, *Journal of Computational Biology* 6 (1999), no. 3-4, 281–297.
- [8] PO Brown and D. Botstein, *Exploring the new world of the genome with dna microarrays*, *Nature Genetics Supplement* 21 (1999), 33–37.
- [9] GA. Carpenter and S. Grossberg, *A massively parallel architecture for a self-organizing neural pattern recognition machine*, *comput, Vision Graphics Image Process* 37 (1987), 54–115.
- [10] P. Cheeseman and J. Stutz, *classification (autoclass): theory and results*, in: *U.m. fayyard, g. piatetsky-shapiro, p. smyth, r. uthurusamy (eds.)*, *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge (1996).

- [11] Tung-Shou Chen, Tzu-Hsin Tsai, Yi-Tzu Chen, Chin-Chiang Lin, Rong-Chang Chen, Shuan-Yow Li, and Hsin-Yi Chen, *A combined k-means and hierarchical clustering method for improving the clustering efficiency of microarray*, Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium (2005), 405–408.
- [12] Tak chung Fu, *A review on time series data mining*, Journal of Time Series Analysis 24 (2011), no. 1, 164–181.
- [13] MB. Eisen, PT. Spellman, PO. Brown, and D. Botstein, *Cluster analysis and display of genome-wide expression patterns*, Proceedings of National Academy Sciences 95 (1998), no. 25, 14863–14868.
- [14] J. Ernst, G.J. Nau, and Z.B. Joseph, *Clustering short time series gene expression data*, Bioinformatics 21 (2005), no. 1, 159–168.
- [15] M. Ester, HP. Kriegel, J. Sander, and X. Xu, *A densitybased algorithm for discovering clusters in large spatial databases*, Proceedings of the 1996 International Conference on Knowledge Discovery and Data Mining (KDD96) (1996), 226–231.
- [16] V. Estivill-Castro and A.T. Murray, *Spatial clustering for data mining with genetic algorithms*, <http://citeseer.nj.nec.com/estivill-castro97spatial.html>.
- [17] TC. Fu, FL. Chung, V. Ng, and R. Luk, *Pattern discovery from stock time series using self-organizing maps*, KDD 2001 Workshop on Temporal Data Mining, August 2629, San Francisco (2001), 27–37.
- [18] I. Gath and A.B. Geva, *Unsupervised optimal fuzzy clustering*, IEEE Transaction on Pattern Analysis 7 (1989), 773–781.
- [19] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P.Boesiger, *A new correlation-based fuzzy logic clustering algorithm for fmri*, Magnetic Resonance in Medicine 40 (1998), 249–260.
- [20] TR. Golub, DK. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, JP. Mesirov, H. Coller, ML. Loh, JR. Downing, MA. Caligiuri, CD. Bloomfield, and ES. Lander, *Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring*, Science 286 (1999), no. 5439, 531–537.
- [21] C. Goutte, LK. Hansen, MG. Liptrot, and E. Rostrup, *Featurespace clustering for fmri meta-analysis*, Human Brain Mapping 13 (1999), 165–183.

- [22] C. Goutte, P. Toft, and E. Rostrup, *On clustering fmri time series*, Neuroimage 9 (1999), no. 3, 298–310.
- [23] S. Guha, R. Rastogi, and K. Shim, *Cure: an efficient clustering algorithm for large databases*, Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, WA (1998), 73–84.
- [24] Balazs Györfy, Bela Molnar, Hermann Lage, Zoltan Szallasi, and Aron C. Eklund, *Evaluation of microarray preprocessing algorithms based on concordance with rt-pcr in clinical samples*, PLoS ONE (2009), no. 5, 5645–5651.
- [25] JG Hacia, JB Fan, O. Ryder, L. Jin, K. Edgemon, G. Ghandour, RA Mayer, B. Sun, L. Hsie, CM Robbins, LC Brody, D. Wang, ES Lander, R. Lipshutz, SP Fodor, and FS Collins, *Determination of ancestral alleles for human single-nucleotide polymorphisms using high-density oligonucleotide arrays*, Nat Genet 22 (1999), no. 2, 164–167.
- [26] LO. Hall, B. zyurt, and JC. Bezdek, *Clustering with a genetically optimized approach*, IEEE Transaction on Evolutionary Computation 3 (1999), no. 2, 103–112.
- [27] J. Han and M. Kamber, *Data mining: Concepts and techniques*, Morgan Kaufmann, San Francisco (2001), 346–389.
- [28] D. Huson and D. Bryant, *Application of phylogenetic networks in evolutionary studies*, Molecular Biology Evolution 23 (2006), 254–267.
- [29] D. Jiang, C. Tang, and A. Zhang, *Cluster analysis for gene expression data: a survey*, IEEE Computer Society 20 (2004), no. 7, 1370–1386.
- [30] Kang-Kun Lee Jin-Yong Lee, *Use of hydrologic timeseries data for identification of recharge mechanism in a fractured bedrock aquifer system*, Journal of Hydrology 229 (2000), no. 3-4, 190–201.
- [31] J. Quackenbush, *Microarray data normalization and transformation*, Nature Genetics 32 (2002), no. 10, 496–501.
- [32] G. Karypis, EH. Han, and V. Kumar, *Chameleon: hierarchical clustering using dynamic modeling*, Computer August (1999), 68–75.
- [33] T. Kohonen, *The self organizing maps*, Proceedings on IEEE (1990).
- [34] K. Komelj and V. Batagelj, *Cross-sectional approach for clustering time varying data*, Journal of Classification 7 (1990), 99–109.

- [35] Hans-Peter Kriegel, Peer Krger, and Arthur Zimek, *Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering*, ACM Transactions on Knowledge Discovery from Data (New York, NY: ACM) 3 (2009), no. 1, 1–58.
- [36] K. Krishna and MN. Murty, *Genetic k-means algorithms*, IEEE Transactions on System Man and Cybernetics (1999).
- [37] R. Krishnapuram and J. Kim, *A note on the gustafsonkessel and adaptive fuzzy clustering algorithms*, IEEE Transaction on Fuzzy systems 7 (1999), no. 4, 453–461.
- [38] _____, *A note on the gustafson-kessel and adaptive fuzzy clustering algorithms*, IEEE Transactions on Fuzzy Systems 3 (2005), no. 9, 1281–1286.
- [39] M. Kumar, N.R. Patel, and J. Woo, *Clustering seasonality patterns in the presence of errors*, Proceedings of KDD 02, Edmonton, Alberta, Canada (2002).
- [40] SJ Kwon, MY Lee, B. Ku, DH Sherman, and JS Dordick, *High-throughput, microarray-based synthesis of natural product analogues via in vitro metabolic pathway construction*, ACS Chemistry Biology 2 (2007), no. 6, 419–425.
- [41] C. Li and G. Biswas, *Temporal pattern generation using hidden markov model based unsupervised classification*, in: D.J. Hand, J.N. Kok, M.R. Berthold (Eds.), Lecture Notes in Computer Science, vol. 164, IDA 99, Springer, Berlin (1999), 245–256.
- [42] C. Li, G. Biswas, M. Dale, and P. Dale, *Building models of ecological dynamics using hmm based temporal data clusteringa preliminary study*, in: F. Hoffmann et al. (Eds.), IDA 2001, Lecture Notes in Computer Science 2189 (2001), 53–62.
- [43] T. Warren Liao, *Clustering of time series dataa survey*, Pattern Recognition 38 (2005), no. 10, 1857–1874.
- [44] _____, *A clustering procedure for exploratory mining of vector time series*, Pattern Recognition 40 (2007), no. 9, 2550–2562.
- [45] TW. Liao, B. Bolt, J. Forester, E. Hailman, C. Hansen, R.C. Kaste, and J. OMay, *Understanding and projecting the battle state*, 23rd Army Science Conference, Orlando, FL, December (2002), 2–5.
- [46] LMD., Owsley, LE. Atlas, and GD. Bernard, *Self-organizing feature maps and hidden markov models for machinetool monitoring*, IEEE Transactions on Signal Process 45 (1997), no. 11,

2787–2798.

- [47] EA. Maharaj, *Clusters of time series*, Journal of Classification 17 (2000), 297–314.
- [48] Vilela Marco, Vinga Susana, Maia Marco, Voit Eberhard, and Almeida Jonas, *Identification of neutral biochemical network models from time series data*, BMC: System Biology 3 (2009), no. 1, 1752–1764.
- [49] U. Maskos and EM Southern, *Oligonucleotide hybridizations on glass supports: a novel linker for oligonucleotide synthesis and hybridization properties of oligonucleotides synthesised in situ*, Nucleic Acids Res. 16 (1992), no. 11, 1679–1684.
- [50] L. Meng, QH. Wu, and ZZ. Yong, *Genetic k-means algorithms*, System Series B: Application of Algorithms 9 (2002), 421–438.
- [51] C.S. Mller-Levet, F. Klawonn, KH. Cho, and O. Wolkenhauer, *Fuzzy clustering of short time series and unevenly distributed sampling points*, Proceedings of the 5th International Symposium on Intelligent Data Analysis, Berlin, Germany (2003), 28–30.
- [52] G. Moran, C. Stokes, S. Thewes, B. Hube, DC Coleman, and D. Sullivan, *Comparative genomics using candida albicans dna microarrays reveals absence and divergence of virulence-associated genes in candida dubliniensis*, Microbiology 150 (2004), no. 10, 3363–3382.
- [53] T. Oates, L. Firoiu, and PR. Cohen, *Clustering time series with hidden markov models and dynamic time warping*, Proceedings of the IJCAI-99 Workshop on Neural, Symbolic, and Reinforcement Learning Methods for Sequence Learning.
- [54] D. Piccolo, *A distance measure for classifyingarma models*, Journal of Time Series Analysis 11 (1990), no. 2, 153–163.
- [55] S. Policker and AB. Geva, *Nonstationary time series analysis by temporal clustering*, IEEE Transactions on System Man and Cybernetics 30 (2000), no. 2, 339–343.
- [56] M. Ramoni, P. Sebastiani, and P. Cohen, *Multivariate clustering by dynamics*, Proceedings of the 2000 National Conference on Artificial Intelligence (AAAI-2000), San Francisco, CA, (2000), 633–638.
- [57] ———, *Bayesian clustering by dynamics*, Maching Learning 47 (2002), no. 1, 91–121.
- [58] MO. Reese, R. Michae van Dam, A. Scherer, and SR. Quake, *Microfabricated fountain pens for high-density dna arrays*, Genome Research (2003).

- [59] E. Sega, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, *Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data*, Nature Genetics 34 (2003), no. 2, 166–176.
- [60] CT. Shaw and GP. King, *Using cluster analysis to classify time series*, Physica D 58 (1992), 288–298.
- [61] R.H. Shumway, *Timefrequency clustering and discriminant analysis*, Statistics and Probability Letters 63 (2003), 307–314.
- [62] D. Tran and M. Wagner, *Fuzzy c-means clustering-based speaker verification*, in: N.R. Pal, M. Sugeno (Eds.), AFSS 2002, Lecture Notes in Artificial Intelligence 2275 (2002), 318–324.
- [63] CA Tsai, SJ Wang, DT Chen, and JJ Chen, *Sample size for gene expression microarray experiments*, Bioinformatics 21 (2005), no. 8, 1502–1508.
- [64] JS Verducci, VF Melfi, S. Lin, Z. Wang, S. Roy, and CK Sen, *Microarray analysis of gene expression: considerations in data mining and statistical treatment*, Nucleic Acids Res 25 (2006), 355–363.
- [65] GA. Viswanathan, Jeremy Seto, Sonali Patil, German Nudelman, and Stuart C. Sealfon, *Getting started in biological pathway construction and analysis*, PLoS Computational Biology 4 (2008), no. 2, e16.
- [66] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos, *A waveletbased anytime algorithm for k-means clustering of time series*, Proceedings of the Third SIAM International Conference on Data Mining (2003).
- [67] Junbai Wang, Trond H B, Inge Jonassen, Ola Myklebost, and Eivind Hovig, *Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data*, Journal of Time Series Analysis 4 (2003), no. 60.
- [68] L. Wang, MG. Mehrabi, and E. Kannatey-Asibu, *Hidden markov model-based wear monitoring in turning*, Journal of Manufacturing Science and Engineering 124 (2002), 651–658.
- [69] W. Wang, J. Yang, and R. Muntz, *Sting: a statistical information grid approach to spatial data mining*, Proceedings of the 1997 International Conference on Very Large Data Base (1997), 186–195.

- [70] JG. Wilpon and LR. Rabiner, *Modified k-means clustering algorithm for use in isolated word recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing 33 (1985), no. 3, 587–594.
- [71] A. Wismler, O. Lange, DR. Dersch, GL. Leinsinger, K. Hahn, B. Ptz, and D. Auer, *Cluster analysis of biomedical image time series*, International Journal of Computer Vision 46 (2002), no. 2, 103–128.
- [72] KY. Yeung, DR. Haynor, and WL. Ruzzo, *Validating clustering for gene expression data*, IEEE Transaction on Fuzzy systems 17 (2001), no. 4, 309–318.
- [73] Wai-Ki Yip, Samir B Amin, Cheng Li, Henry Horng-Shing Lu, Bernhard Schlkopf, and Hongyu Zhao, *A survey of classification techniques for microarray data analysis*, Handbook of Statistical Bioinformatics in: Springer Handbooks of Computational Statistics (1990), 193–223.
- [74] T. Zhang, R. Ramakrishnan, and M. Livny, *Birch: an efficient data clustering method for very large databases*, Proceedings of the 1996 ACM-SIGMOD International Conference on Management of Data, Montreal, Canada (1996), 103–114.
- [75] J. Zhou, *Microarrays for bacterial detection and microbial community analysis*, Current Opinion in Microbiology 6 (2003), no. 3, 288–294.