2010

Microarray time-series data clustering via gene expression profile alignment

K M Numanul Hoque Subhani
University of Windsor

Microarray Time-Series Data Clustering via Gene Expression Profile Alignment

by

K M Numanul Hoque Subhani

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in partial fulfilment of the requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2010

Your file Votre référence
ISBN: 978-0-494-62711-2
Our file Notre référence
ISBN: 978-0-494-62711-2

Declaration of Co-Authorship

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

This thesis also incorporates the outcome of a joint research undertaken in collaboration with professors Dr. Alioune Ngom and Dr Conrad Burden (Australian National University) under the supervision of professor Dr Luis Rueda. The collaboration is covered in Chapter 3 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of guidance corrections and constructive criticism.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

Declaration of Previous Publications

This thesis includes two original papers that have been previously published/submitted for publication in peer reviewed journals (see next page - Table I):

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the

Table 1: Declaration of Previous Publications

Thesis Chapter	Publication title/full citation	Publication status
Chapters 3, 4	***N. Subhani, A. Ngom, L. Rueda, C. Burden**: Microarray Time-Series Data Clustering via Multiple Alignment of Gene Expression Profiles. Springer Transaction on Pattern Recognition in Bioinformatics. LNCS 5780 (2009) 377-390*	*Published*
Chapters 3, 4	***N. Subhani, L. Rueda, A. Ngom, C. Burden**: Clustering Microarray Time-Series Data using Expectation Maximization and Multiple Profile Alignment. IEEE International Conference on Bioinformatics and Biomedicine Workshop, ISBN: 978-1-4244-5121-0 (2009) 2-7*	*Published*

standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Clustering gene expression data given in terms of time-series is a challenging problem that imposes its own particular constraints, namely, exchanging two or more time points is not possible as it would deliver quite different results and would lead to erroneous biological conclusions.

In this thesis, clustering methods introducing the concept of *multiple alignment* of natural cubic spline representations of gene expression profiles are presented. The multiple alignment is achieved by minimizing the sum of integrated squared errors over a time-interval, defined on a set of profiles. The proposed approach with flat clustering algorithms like k-means and EM are shown to cluster microarray time-series profiles efficiently and reduce the computational time significantly. The effectiveness of the approaches is experimented on six data sets. Experiments have also been carried out in order to determine the number of clusters and to determine the accuracies of the proposed approaches.

Dedication

To my departed father whom I loved most, for his unconditional love, support and his endless encouragements ...

Acknowledgements

Firstly, I would like to thank my parents. Without their encouragement, support and love, it would not have been possible for me to pursue so many great achievements in my life.

I would like to express my deepest appreciation to my advisors, Dr. Alioune Ngom and Dr. Luis Rueda, for their encouragement, support and invaluable suggestions in guiding me towards the successful completion of this research work. Without their generous funding and advice, it would be hard for me to achieve many publications in this work. I would also like express to my appreciation to Dr. Conrad Burden of Australian National University for his invaluable suggestions and constructive criticisms.

I would like to express my great gratitude to Dr. Dennis Higgs, Department of Biology, and Dr. Robin Gras, School of Computer Science for giving me corrections and constructive criticism to improve the quality of this research, for their patience in arranging the time of my proposal and defense, and for being in the committee, and Dr. Jianguo Lu for serving as the chair of the committee.

I gratefully acknowledge the assistance of Mr. Yifeng Li of School of Computer Science for helping me to implement the VCD method.

Finally, I want to extend my gratitude to my friends, the faculty members and staff of the School of Computer Science for their friendly suggestions and support during my study at University of Windsor.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Microarray Technology

Micorrarrys are widely used tools in molecular biology providing a fast and cost-effective method for monitoring the expression of thousands of genes simultaneously [30]. Microarrays enable monitoring of whole-genome expression in a single experiment. The biological interpretation of large datasets is the biggest challenge for scientists confronted with gene expression data.

A cDNA microarray is an arrayed series of thousands of microscopic spots of DNAs each containing a specific DNA sequence, known as probe. A probe can be a short section of a gene or other DNA element that is used to hybridize a cDNA or cRNA sample, known as target. In oligonucleotide microarrays, the probes are short sequences designed to match parts of the sequence of known or predicted open reading frames. Oligonucleotide arrays are produced by printing short oligonucleotide sequences designed to represent a single gene or family of gene splice-variants by synthesizing this sequence directly onto the

array surface instead of depositing intact sequences. Sequences may be longer (60-mer probes) or shorter (25-mer probes) depending on the desired purpose; longer probes are more specific to individual target genes, while shorter probes can be spotted in higher density across the array and are cheaper to be produced. In spotted microarrays, the probes are oligonucleotides, cDNA or small fragments of PCR products that correspond to mRNAs. The probes are synthesized prior to deposition on the array surface and are then "spotted" onto the glass.

Microarrays are solid substrates hosting hundreds of single stranded DNAs with a specific sequence. DNA Microarrays are solid supports onto which the sequences from thousands of different genes are attached at fixed locations. The supports themselves are usually glass microscope slides, but can be silicon chips or nylon membranes on which the DNA is printed, spotted or synthesized. The whole microarray technology is based on *hybridization probing*, a technique that uses fluorescence labeled nucleic acid molecules as mobile probes to identify complementary molecules. A typical DNA microarray experiment involves the following steps:

1. Preparing the DNA chip using the chosen targets.

2. Generating hybridization mixture of fluorescence labeled cDNAs.

3. Incubating hybridization mixture with the DNA chip.

4. Detecting bound cDNA using laser technology

5. Analyzing data using advanced computational methods.

Figure 1.1* illustrates the typical process of a DNA microarray experiment.

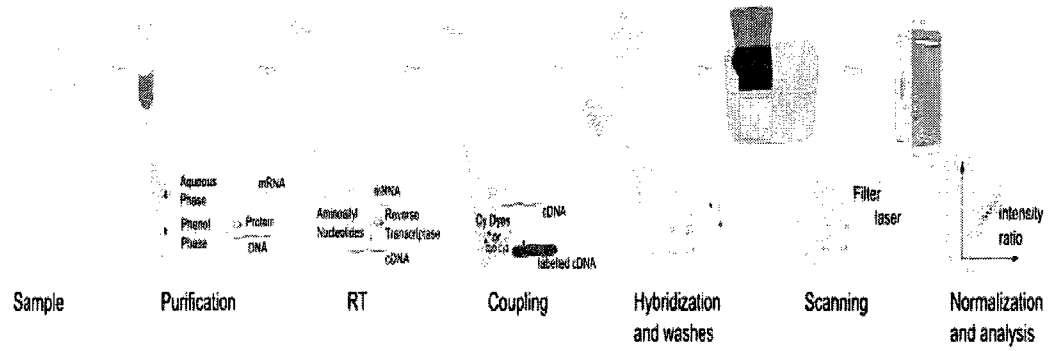

Figure 1.1: The steps required in a microarray experiment.

The amount of fluorescence emitted by each cDNA array will be proportional to the amount of mRNA produced from the gene having the corresponding DNA sequence. The above description is for DNA microarrays only but the microarray experiments vary according to the specific type of microarray. In [23], four main technology platforms of microarrays are described: 1) Nylon membrane arrays or radioactive filters; 2) cDNA arrays or red/green arrays; 3) Polynucleotide arrays; 4) Oligonucleotide arrays (also called DNA chips). cDNA technology is the most commonly used one, and allows spotting of almost any PCR product. Gene expression levels are detected over a period of time from microarrays. Then the expression ratio of genes are measured by using different logarithmic and normalization techniques. These kinds of gene expressions over a period of time are called time-series gene expression. Time-series gene expression data can be produced by any of the above microarrays and even other technologies. We are considering any form of the technology that produces *time-series gene expression data.*

*`http://en.wikipedia.org/wiki/File:Microarray_exp_horizontal.svg`, Image used under public domain license.

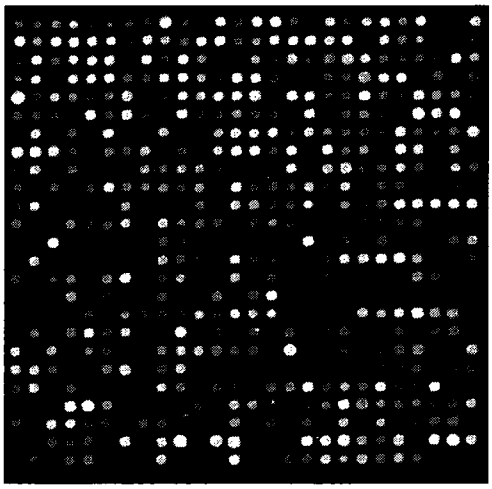

Figure 1.2: A sample DNA microarray.

1.2 Microarray Analysis

Microarray technology has the following important advantages:

1. it can measure the expression levels of thousands of genes in parallel,
2. it provides semi-quantitative data, and
3. it is sensitive enough to detect low-abundance transcripts that are represented on a given array.

DNA microarrays are used for measuring the concentration of mRNA in living cells. The concentration of a particular mRNA transcript is measured as the *expression* level of its corresponding gene. The *expression profile*, a snapshot of the total mRNA pool of living cell or tissue, can be obtained when different probes matching all mRNAs in a cell are used. It reflects the expression of every single measured gene at that particular moment. The expression can also be used to quantify the expression of a single gene over a number of conditions.

Microarrays have been successfully used in a wide range of applications including sequencing, SNP detection and cluster analysis. However, the main application still remains the investigation of the genetic mechanisms in living cells. The microarray technology has a very high throughput interrogating thousands of genes at the same time. It has been proved that microarrays can be used to generate reliable and accurate gene expression data [7, 37]. It can also be used for purely computational purposes such as in the field of DNA computing [5].

1.3 Microarray Time-Series Gene Expression

An increasingly popular method for studying a wide range of biological systems is through time-series expression experiments. In time-series expression experiments, a snapshot of the expression of genes in a temporal process is measured rather than in different samples. Another main characteristic of the time-series data is to exhibit a strong autocorrelation between successive points rather than from a sample population (which are assumed to be independent and identically distributed).

Gene expression is a measurement of expressed gene over a certain period of times under different conditions. Different proteins are required and synthesized for different functions, under different conditions and at different times. One of the most important ways of new protein generation in which the cell regulates gene expression is by using a feedback loop. In many cases, the expression program starts by activating a few transcription factors (TF), which in turn activate many other genes that act in response to the new con-

dition. It is necessary to measure a time course of expression experiments in order to determine the complete set of genes that are expressed under new conditions.

Much of the early work on analyzing time-series expression experiments used methods developed for static data [19]. Recently, several new approaches were presented specially targeting time-series expression data [3, 21, 31, 11, 36]. We are also presenting few new clustering approaches specifically for time-series gene expression profile analysis in Chapter 3 and 4.

1.4 Motivation and Objective

An important process in functional genomic studies is clustering microarray time-series data, where genes with similar expression profiles are expected to be functionally related. A common problem in biology is to partition a set of experimental data into clusters in such a way that the data points within the same cluster are highly similar while data points in different clusters are as dissimilar as possible. Profile alignment clustering is based on deciding upon the similarity often involves pairwise distance measures of co-expressions. Clustering algorithms that apply a conventional distance (e.g. the Euclidian distance, correlation coefficient) function normally do not reflect the temporal data embedded in the expression profiles.

We are proposing new *profile alignment* approaches to cluster microarray time-series gene expression profiles. Clustering time-series expression data with unequal time intervals is a very special problem, as measurements are not necessarily taken at regular time points. The area-based profile alignment

proposed in [15] takes two features vectors, and produces two new vectors in such a way that the area between the "aligned" vectors is minimized. The profile alignment method that takes the length of the intervals between the time-points into consideration was proposed in [15]. In both [15] and [25], hierarchical agglomerative clustering is used where the decision rule is based on the *furthest-neighbor* or *complete linkage* distance between two clusters. That clustering approach performs the pairwise alignment before measuring the distance between two profiles during each iteration, which slows down the computational process. Also, piecewise linear representation of gene expression profile was used which does not reflect the actual representation of the gene expression.

To reflect the actual representation of the gene expression profiles, we generalize piecewise linear profiles to natural cubic spline profiles. Taking the lengths of the time intervals into account is accomplished by means of analyzing the area between two expression profiles, joined by the corresponding measurements at subsequent time points. This is equivalent to considering the sum or average of squared errors between the infinite points in the two lines. This analysis can be easily achieved by computing the underlying integral, which is analytically resolved in advance, subsequently avoiding expensive computations during the clustering process. Our approach allows us to apply flat clustering such as k-means, which, though not optimal, provides a fast and practical solution to the problem. We also apply our approach to the expectation maximization (EM) clustering method.

1.5 Contributions

In this thesis, clustering approaches are proposed based on the concept of *Profile-Alignment*, for clustering microarray time-series gene expression profiles. Our main contributions in this thesis are:

1. Generalize the theoretical results of [15] to any *continuously integrable* representation of time-series gene expression profiles. The contributions are:

   a) Piecewise Linear (PL) function to Natural Cubic Spline (NCS) function representation.

   b) Pairwise alignment of NCS functions.

   c) Distance between two NCS functions.

   d) Analytical solutions of b) and c).

2. Multiple Alignment of NCS representations and of PL representations of gene expressions time-series profiles. The contributions are:

   a) Universal Alignment Theorem: align the profiles such that the squared error between any two *vertically shifted* profiles is minimal.

   b) Centroid of a cluster: a centroid function, which aim to find representative profile of a cluster, defined based on natural cubic spline profiles.

   c) Analytical solutions of a) and b).

3. Clustering approaches using alignment methods. The contributions are:

a) Theoretical results: clustering multiple-aligned data is equivalent to clustering original data, but is faster when using multiple-aligned data.

b) Clustering multiple-aligned data using any representation (NCS or PL):

   i. k-Means clustering via multiple alignment (k-MCMA): an algorithm that clusters multiple aligned profiles with k-means.

   ii. EM clustering via multiple alignment (EMMA): a method that combines EM and multiple alignment of gene expression profiles to cluster microarray time-series data.

   iii. Theoretical result: we can cluster with any distance-based clustering method.

c) New measure of clustering accuracy using:

   i. Hungarian matching algorithm for clustering-phase assignment.

   ii. c-Nearest Neighbor (c-NN) method: combined with cross-validation and validity indices.

Our major contribution is to use the benefit of alignment method in combination with any clustering method. Initialization is a major issue in k-means and EM methods but we are not interested in improving k-means or EM clustering methods.

1.6 Thesis Organization

The thesis is organized in five chapters. Chapter II provides a survey of clustering, microarray time-series clustering and clustering with alignment. Chapters III and IV present the proposed alignment approaches and clustering algorithms, respectively. Chapter V deals with experimental results and performance analysis, where all proposed approaches are analyzed and compared to existing methods. Finally, Chapter VI concludes the thesis and identifies open research problems arising from this work.

Chapter 2

Microarray Time-Series Data Clustering

A brief review about clustering and its uses are discussed in this chapter. Microarray time-series data clustering is also formally defined here. The literature review of previous works on clustering, and specially microarray time-series data clustering are discussed in Section 2.3.

2.1 Clustering

Clustering is a multivariate analysis technique used to discover unknown patterns or groups in data. Clustering is appropriate when there is no a *priori* knowledge about the underlying data. Clustering, the process of grouping similar entities, can be done on any data such as genes, samples, time points in a time-series, etc. The particular type of input makes no difference on the clustering algorithm. The algorithm will treat all inputs as an n-dimensional

feature vector. To group objects that are similar, we need a very precise definition of measure of similarity. There are many different ways in which such a measure of similarity can be calculated depending on the representation of gene expression profiles. We are considering clustering on microarray time-series expression profiles.

2.2 Microarray Time-Series Data Clustering

In this section, we discuss the clustering the problem of the microarray time-series gene expression profiles. Time-Series clustering problem is formally stated in order to discuss these approaches. Given a dataset $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$. $x_i = [x_{i_1}, \ldots, x_{i_n}]^t$ is an n-dimensional feature vector that represents the expression level of gene i at n different time points, $t = [t_1, \ldots, t_n]^t$. We want to partition a set of s profiles, $\mathcal{D}$, into k disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, $1 \leq k \leq s$; such that (i) $\mathcal{C}_i \neq \emptyset, i = 1, \ldots, k$; (ii) $U_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$ (iii) $\mathcal{C}_i \cap \mathcal{C}_j = \phi$; $i \neq j$; $i, j = 1, \ldots, k$. Also, each profile is assigned to the cluster whose distance is the closest. We are considering the specific case of time-series clustering, where the order of time-points cannot be permuted because of the different permutations give different results which are biologically meaningless.

2.3 Literature Review

Many clustering methods for time-series gene expression data have been developed. A partitional clustering method based on k-means applied in [29] to cluster gene expression temporal data.This approach does not require any

prior knowledge, except the value of k needs to be known *a priori*, about the structure or to make any assumptions about the dynamics of the expression profile. In [20], Tamayo *et al* applied Self-organizing maps (SOM) to visualize and interpret the patterns of gene temporal expression profiles. The SOM, a type of mathemetical cluster analysis suits well with exploratory analysis of the data and to reveal relevant patterns in a large, high-dimensional dataset. Several other methods also have proposed including a jack-knife correlation coefficient model [14], an order-restricted inference-based method [28], a statistical two-regression step approach [1], a method for assigning genes to predefined set of model profiles [12], and combined spline smoothing and first derivative computation [27]. In [6], fuzzy clustering of time-series data based on the similarity of relative change of expression level and the corresponding temporal information of the profiles.

A hidden phase model was used for clustering time-series data to define the parameters of a mixture of normal distributions in a Bayesian-like manner that are estimated by using expectation maximization (EM) [4]. A Bayesian approach in [16], partitional clustering based on k-means in [29] and an Euclidean distance approach in [20] have been proposed for clustering time-series gene expression profiles. They have applied self-organizing maps (SOMs) to visualize and interpret gene temporal expression profile patterns. Also, the methods proposed in [14, 26] are based on correlation measures. A method that uses jack-knife correlation with or without using seeded candidate profiles was proposed for clustering time-series microarray data as well [14]. Specifying expression levels for the candidate profiles in advance for these correlation-based procedures requires estimating each candidate profile, which is made

using a small sample of arbitrarily selected genes. The resulting clusters depend upon the initially chosen template genes, because there is a possibility of missing important genes. A regression-based method, which is suitable for analyzing single or multiple microarrays was proposed in [12] to address the challenges in clustering short time-series expression datasets.

A Bayesian approach for improving the clustering results of gene expression series using rough knowledge the general shapes of the classes was proposed [4]. Knowledge about the general shapes can be elementary regarding the change of the mean expression level over time. The information regarding the shape of the class are directly integrated into the model so that class with the desired profiles are favored. A Bayesian method was also applied for model-based clustering where the models are autoregressive curves of fixed order [16]. To search for the most likely set of clusters out of the given temporal expression data, an agglomerative procedure was used. The dynamic nature of gene expression time-series data explicitly takes into account during clustering. This approach also identifies the number of distinct clusters based on the well-known Akaike information criterion. This approach is a specialized version of Bayesian Clustering by Dynamics where two time-series are considered similar if they are generated by the same stochastic process.

In [28], Peddada *et al.* applied an order-restricted inference method for selecting and clustering genes expression profiles for time-series or dose-response data. The method applies the ideas of order-restricted inference and uses known inequalities among parameters. In this procedure, two profiles are placed in the same cluster only if all the inequalities between the expected expression levels at various time points are the same. This method makes

use of the ordering in a time-series study and can detect genes more sensitively using their temporal ordering and finding consistent patterns over time. A regression-based approach that identifies genes with different expression profiles across analytical groups in time-series experiments was proposed in [1].This method uses a two-step regression strategy, where the first step adjusts a global regression model with all the defined variables to identify differentially expressed genes and the next step finds statistically-significant different profiles by applying a variable selection strategy that studies the difference between the groups. This method can be used to find genes with significant temporal expression changes between experimental groups, and to analyze the magnitude of these differences.

The analysis of gene temporal expression profiles with the problem of missing values and non-uniformly sampled data was discussed in [38]. Each expression profile estimated from observed data where gene temporal expression profiles are represented as continuous curve using statistical spline estimation. The spline coefficients of the genes are constrained in such way that similar expression patterns fall into the same class. In [27], a method that focuses on the *shapes of the curves* and not on the *absolute levels of expression* was proposed to obtain relevant clustering of gene expression temporal profiles by identifying homogeneous clusters of genes. It combines first derivative computations and spline smoothing with hierarchical and partitional clustering. This approach is based on the framework of functional data analysis [24], which focuses on the first derivative of curves by means of *a priori* spline smoothing.

A similarity measure for the co-expressed genes based on the expression

level rate of change across time-points was proposed in [6]. The similarity between gene expression time-series profiles was calculated by measuring the difference of the slopes between the functions, where gene temporal profiles were represented as piece-wise linear functions. The variable time intervals are viewed as weights, where far apart expressions take smaller weights in the comparison. A clustering algorithm was proposed which is motivated by the advantages of fuzzy clustering, and incorporates the distance measure in the fuzzy-c-means clustering scheme [10].

Clustering based on profile alignment has been discussed recently [3, 21, 31, 11, 36]. In [36], the authors proposed an approach that translates gene expression into gene variation vectors and derives the proximity measure for these vectors. In [3], the authors proposed a method that finds clusters of genes such that the genes within a cluster share a common alignment, but each cluster is aligned independently of the others. The authors also present a segment-based alignment algorithm for time series. A clustering method that uses a local shape-based similarity measure based on Spearman rank correlation is proposed in [21]. In their method, similar local regions can be time-shifted to allow the detection of transcription control relationships. An alignment method that uses HMMs to align time-series gene expression to a common profile has been introduced in [31]. An Area-based profile alignment and mean-square-error profile alignment methods have been introduced in [15] and [25], respectively.

Chapter 3

Gene Expression Profile Alignment Methods

Many clustering methods have been developed, and each has its own advantages and disadvantages regarding handling noise in the measurements and the properties of the data set being clustered. In [15], hierarchical clustering was used and the decision rule was the *farthest-neighbor* distance between two clusters computed using an equivalent of Eq. (3.1) for piece-wise linear profiles. Hierarchial clustering is a greedy method that cannot be readily applied on large data sets.

An important process in functional genomic studies is clustering microarray time-series data, where genes with similar expression profiles are expected to be functionally related. A common problem in biology is to partition a set of experimental data into clusters in such a way that the data points within the same cluster are highly similar while data points in different clusters are as dissimilar as possible. Profile alignment clustering is based on

deciding upon the similarity often involves pairwise distance measures of co-expressions. Clustering algorithms that apply a conventional distance (e.g. the Euclidian distance, correlation coefficient) function normally do not reflect the temporal data embedded in the expression profiles. We are proposing new *profile alignment* approaches to cluster microarray time-series gene expression profiles.

3.1 Clustering with Alignment

There is some alignment techniques already introduced to resolve this issue before applying the distance function. Area based profile alignment proposed in [15] takes two features vectors, and produces two new vectors in such a way that the area between "aligned" vectors is minimized. The profile alignment method that takes the length of the intervals between the time-points into consideration was proposed in [15]. That approach considers the weights of the intervals equally, irrespective to the actual size of the interval of the measurement. The *Profile-Alignment* algorithm takes two feature vectors from the original space as input and outputs two feature vectors in the transformed space after aligning them in such way that the sum of squared errors is minimized. The alignment of the profiles is done using an area-based distance function rather than conventional distance functions. The area-based distance function is defined by computing the integral distance between the two aligned profiles. In both [15] and [25], hierarchical agglomerative clustering is used where the decision rule is based on the *furthest-neighbor* or *complete linkage* distance between two clusters. The *complete linkage* or *furthest-neighbors*

approach calculates the distance between the furthest pair of points for each pair of clusters and merges the two clusters that have the minimum distance among all such distances between all pair of clusters under consideration. That clustering approach does the pairwise alignment before measuring distance between two profiles during each iteration, which slows down the computational process. Also piecewise linear representation of gene expression profile does not reflect the actual representation of the gene expression.

We re-formulate the profile alignment problem of [15] in terms of integrals of arbitrary functions, allowing us to generalize from a piecewise linear interpolation to any type of interpolation one believes be more physically realistic. The expression measurements are basically snapshots taken at time-points chosen by the experimental biologist. The cells expressing genes do not know when the biologist is going to choose to measure gene expression, which one would guess is changing continuously and smoothly at all the time points. Thus, smooth spline curve through the known time-points in the cell's expression path would be a better guess. We use natural cubic spline interpolation to represent each gene expression profile, which gives a handy way to align profiles for which measurements were not taken at the same time-points. We generalize the pairwise expression profile alignment formulae of [15] from the case of piece-wise linear profiles to profiles which are any continuous integrable function on a finite interval. Next, we extend the concept of pairwise alignment to multiple expression profile alignment, where the profiles from a given set are aligned in such a way that the sum of squared errors over a time-interval defined on the set is minimized. Finally, we combine k-means clustering with our multiple alignment approach to cluster microarray time-series data. In

this thesis, we call this clustering approach as *k-Means Clustering via Multiple Alignment* (k-MCMA). Our multiple alignment approach is also combined with expectation-maximization (EM) clustering, called as *EM Clustering via Multiple Alignment* (EMMA) to cluster microarray time-series data.

3.2 Alignment Methods for Continuous and Integrable Functions

3.2.1 Pairwise Alignment

Given two profiles, $x(t)$ and $y(t)$ (either piece-wise linear or continuously integrable functions), where $y(t)$ is to be aligned to $x(t)$, the basic idea of alignment is to *vertically shift* $y(t)$ towards $x(t)$ in such a way that the *squared errors* between the two profiles is minimal. Let $\hat{y}(t)$ be the result of shifting $y(t)$. Here, the *error* is defined in terms of the areas between $x(t)$ and $\hat{y}(t)$ in interval $[0, T]$. Functions $x(t)$ and $\hat{y}(t)$ may cross each other many times, but we want that the sum of all the areas where $x(t)$ is above $\hat{y}(t)$ minus the sum of those areas where $\hat{y}(t)$ is above $x(t)$ to be minimal (see Fig. 3.1). Let a denote the amount of vertical shifting of $y(t)$. Then, we want to find the value $a_{\min}$ of a that minimizes the integrated squared error between $x(t)$ and $\hat{y}(t)$. Once we obtain $a_{\min}$, the alignment process consists of performing the shift on $y(t)$ as $\hat{y}(t) = y(t) - a_{\min}$.

The pairwise alignment results of [15] generalize from the case of piece-wise linear profiles to profiles which are *any* integrable functions on a finite interval. Suppose we have two profiles, $x(t)$ and $y(t)$, defined on the time-interval $[0, T]$.

The alignment process consists of finding the value a that minimizes

$$f_a(x(t), y(t)) = \int_0^T \Big[x(t) - \hat{y}(t)\Big]^2 dt = \int_0^T \Big[x(t) - [y(t) - a]\Big]^2 dt. \quad (3.1)$$

Differentiating yields

$$\frac{d}{da} f_a(x(t), y(t)) = 2\int_0^T \Big[x(t) + a - y(t)\Big] dt = 2\int_0^T \Big[x(t) - y(t)\Big] dt + 2aT. \quad (3.2)$$

Setting $\frac{d}{da} f_a(x(t), y(t)) = 0$ and solving for a gives

$$a_{\min} = -\frac{1}{T}\int_0^T \Big[x(t) - y(t)\Big] dt, \quad (3.3)$$

and since $\frac{d^2}{da^2} f_a(x(t), y(t)) = 2T > 0$ then $a_{\min}$ is a minimum. The integrated error between $x(t)$ and the shifted $\hat{y}(t) = y(t) - a_{\min}$ is then

$$\int_0^T \Big[x(t) - \hat{y}(t)\Big] dt = \int_0^T \Big[x(t) - y(t)\Big] dt + a_{\min} T = 0. \quad (3.4)$$

In terms of Fig. 3.1, this means that the sum of all the areas where $x(t)$ is above $y(t)$ minus the sum of those areas where $y(t)$ is above $x(t)$ is zero.

Given an original profile $x(t) = [e_1, e_2, \ldots, e_n]$ (with n expression values taken at n time-points $t_1, t_2, \ldots, t_n$), we use *natural cubic spline* interpolation, with n knots, $(t_1, e_1), \ldots, (t_n, e_n)$, to represent $x(t)$ as a continuously integrable function

$$x(t) = \begin{cases} x_1(t) & \text{if } \; t_1 \le t \le t_2 \\ \vdots & \\ x_{n-1}(t) & \text{if } \; t_{n-1} \le t \le t_n \end{cases} \quad (3.5)$$

where $x_j(t) = x_{j3}(t - t_j)^3 + x_{j2}(t - t_j)^2 + x_{j1}(t - t_j)^1 + x_{j0}(t - t_j)^0$ interpolates $x(t)$ in interval $[t_j, t_{j+1}]$, with spline coefficients $x_{jk} \in \Re$, for $1 \le j \le n - 1$ and $0 \le k \le 3$.

For practical purposes, given the coefficients, $x_{jk} \in \Re$, associated with $x(t) = [e_1, e_2, \ldots, e_n] \in \Re^n$, we need only to transform $x(t)$ into a new space as $x(t) = [x_{13}, x_{12}, x_{11}, x_{10}, \ldots, x_{j3}, x_{j2}, x_{j1}, x_{j0}, \ldots, x_{(n-1)3}, x_{(n-1)2}, x_{(n-1)1}, x_{(n-1)0}] \in \Re^{4(n-1)}$. We can add or subtract polynomials given their coefficients, and the polynomials are continuously differentiable. This yields an analytical solution for $a_{\min}$ in Eq. (3.3) as follows:

$$a_{\min} = -\frac{1}{T}\sum_{j=1}^{n-1}\int_{t_j}^{t_{j+1}}\Big[x_j(t) - y_j(t)\Big]dt = -\frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3}\frac{(x_{jk} - y_{jk})\,(t_{j+1} - t_j)^{k+1}}{k+1}. \tag{3.6}$$

Fig. 3.1(b) shows a pairwise alignment, of the two initial profiles in Fig. 3.1(a), after applying the vertical shift $y(t) \leftarrow y(t) - a_{\min}$. The two aligned profiles cross each other many times, but the integrated error, Eq. (3.4), is zero.

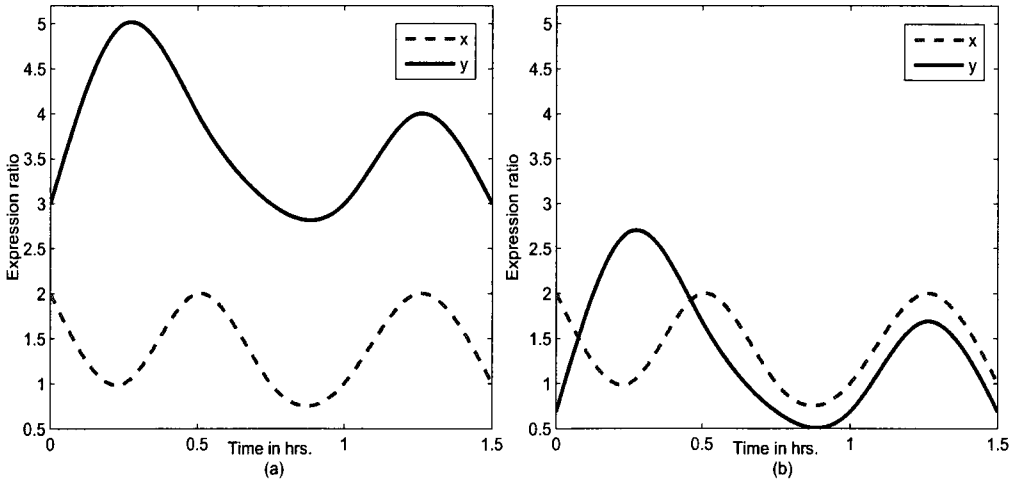

Figure 3.1: (a) Unaligned, and (b) Aligned profiles $x(t)$ and $y(t)$ after applying $y(t) \leftarrow y(t) - a_{\min}$.

In particular, from Eq. (3.4), the horizontal t-axis will bisect a profile $x(t)$ into two halves with equal areas, when $x(t)$ is aligned to the t-axis. In the next

section, we use this property of Eq. (3.4) to define the multiple alignment of a set of profiles.

3.2.2 Multiple Alignment

Given a set $D = \{x_1(t), \ldots, x_s(t)\}$, we want to align the profiles such that the integrated squared error between any two *vertically shifted* profiles is minimal. Thus, for any $x_i(t)$ and $x_j(t)$, we want to find the values of a_i and a_j that minimize

$$f_{a_i,a_j}(x_i(t), x_j(t)) = \int_0^T \left[\hat{x}_i(t) - \hat{x}_j(t)\right]^2 dt = \int_0^T \left[[x_i(t) - a_i] - [x_j(t) - a_j]\right]^2 dt, \tag{3.7}$$

where *both* $x_i(t)$ and $x_j(t)$ are shifted vertically by an amount a_i and a_j, respectively, in possibly different directions, whereas in the pairwise alignment of Eq. (3.1), profile $y(t)$ is shifted towards a *fixed* profile $x(t)$. The multiple alignment process consists then of finding the values of $a_1, \ldots, a_s$ that minimize

$$F_{a_1,\ldots,a_s}(x_1(t), \ldots, x_s(t)) = \sum_{1 \le i < j \le s} f_{a_i,a_j}(x_i(t), x_j(t)), \tag{3.8}$$

We use Lemma 3.2.1 to find the values a_i and a_j, $1 \le i < j \le s$, that minimize $F_{a_1,\ldots,a_s}$.

Lemma 3.2.1. *If $x_i(t)$ and $x_j(t)$ are pairwise aligned each to a fixed profile, $z(t)$, then the integrated error $\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\, dt = 0$.*

Proof. If $x_i(t)$ and $x_j(t)$ are pairwise aligned each to $z(t)$, then from Eq. (3.3), we have $a_{\min_i} = -\frac{1}{T}\int_0^T [z(t) - x_i(t)]\, dt$ and $a_{\min_j} = -\frac{1}{T}\int_0^T [z(t) - x_j(t)]\, dt$. Then,

$\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\, dt = \int_0^T \left[[x_i(t) - a_{\min_i}] - [x_j(t) - a_{\min_j}]\right] dt =$
$\int_0^T x_i(t)dt + \int_0^T [z(t) - x_i(t)]\, dt - \int_0^T x_j(t)dt - \int_0^T [z(t) - x_j(t)]\, dt = 0.$ □

In other words, $\hat{x}_j(t)$ is automatically aligned relative to $\hat{x}_i(t)$, given $z(t)$ is fixed.

Corollary 3.2.2. *If $x_i(t)$ and $x_j(t)$ are pairwise aligned each to a fixed profile, $z(t)$, then $f_{a_{\min_i}, a_{\min_j}}(x_i(t), x_j(t))$ is minimal.*

Proof. From Lemma 3.2.1,
$\int_0^T [\hat{x}_i(t) - \hat{x}_j(t)]\, dt = 0 \Rightarrow \int_0^T \left[[x_i(t) - a_{\min_i}] - [x_j(t) - a_{\min_j}]\right]^2 dt$ is minimal.
□

Lemma 3.2.3. *If profiles $x_1(t), \ldots, x_s(t)$ are pairwise aligned each to a fixed profile, $z(t)$, then $F_{a_{\min_1}, \ldots, a_{\min_s}}(x_1(t), \ldots, x_s(t))$ is minimal.*

Proof. From Corollary 3.2.2, $f_{a_i, a_j}(x_i(t), x_j(t)) \geq f_{a_{\min_i}, a_{\min_j}}(x_i(t), x_j(t))$, with equality holding when $a_k = a_{\min_k}$; which is attained by aligning each $x_k(t)$ independently with $z(t)$, $1 \leq k \leq s$. From the definition of Eq. (3.8), it follows that $F_{a_1, \ldots, a_s}(x_1(t), \ldots, x_s(t)) \geq \sum_{1 \leq i < j \leq s} f_{a_{\min_i}, a_{\min_j}}(x_i(t), x_j(t)) = F_{a_{\min_1}, \ldots, a_{\min_s}}(x_1(t), \ldots, x_s(t))$, with equality holding when $a_k = a_{\min_k}$, $1 \leq k \leq s$. □

Thus, given a fixed profile $z(t)$, applying Corollary 3.2.2 to all pairs of profiles minimizes $F_{a_1, \ldots, a_s}(x_1(t), \ldots, x_s(t))$ in Eq. (3.8).

Theorem 3.2.4. *Given a fixed profile, $z(t)$, and a set of profiles, $X = \{x_1(t), \ldots, x_s(t)\}$, there always exists a multiple alignment, $\hat{X} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$,*

such that

$$\hat{x}_i(t) = x_i(t) - a_{\min_i}, \text{ where, } a_{\min_i} = -\frac{1}{T}\int_0^T \Big[z(t) - x_i(t)\Big]dt, \tag{3.9}$$

and, in particular, for profile $z(t) = 0$, defined by the horizontal t-axis, we have

$$\hat{x}_i(t) = x_i(t) - a_{\min_i}, \text{ where, } a_{\min_i} = \frac{1}{T}\int_0^T x_i(t)dt. \tag{3.10}$$

We use the multiple alignment of Eq. (3.10) in all subsequent discussions. Using spline interpolations, each profile $x_i(t)$, $1 \leq i \leq s$, is a continuously integrable profile

$$x_i(t) = \begin{cases} x_{i,1}(t) & \text{if } t_1 \leq t \leq t_2 \\ \vdots & \\ x_{i,n-1}(t) & \text{if } t_{n-1} \leq t \leq t_n \end{cases} \tag{3.11}$$

where, $x_{i,j}(t) = x_{ij3}(t-t_j)^3 + x_{ij2}(t-t_j)^2 + x_{ij1}(t-t_j)^1 + x_{ij0}(t-t_j)^0$ represents $x_i(t)$ in interval $[t_j, t_{j+1}]$, with spline coefficients x_{ijk} for $1 \leq i \leq s$, $1 \leq j \leq n-1$ and $0 \leq k \leq 3$. Thus the analytical solution for $a_{\min_i}$ in Eq. (3.10) is

$$a_{\min_i} = \frac{1}{T}\sum_{j=1}^{n-1}\sum_{k=0}^{3}\frac{x_{ijk}\,(t_{j+1} - t_j)^{k+1}}{k+1} \tag{3.12}$$

3.2.3 Distance Function

The distance between any two piecewise linear profiles was defined as $f(a_{\min})$ in [15]. For convenience here, we change the definition slightly to:

$$d(x,y) = \frac{1}{T}f(a_{\min}) = \frac{1}{T}\int_0^T \Big[x(t) + a_{\min} - y(t)\Big]^2 dt. \tag{3.13}$$

For any function $\phi(t)$ defined on $[0, T]$, we also define

$$\langle \phi \rangle \triangleq \frac{1}{T} \int_0^T \phi(t) dt. \tag{3.14}$$

Then, from Eqs. (3.1) and (3.3),

$$\begin{aligned} d(x,y) &= \frac{1}{T} \int_0^T \Big[[x(t) - y(t)]^2 + 2a_{\min} [x(t) - y(t)] + a_{\min}^2 \Big] dt \\ &= \frac{1}{T} \int_0^T \Big[x(t) - y(t) \Big]^2 dt - 2a_{\min}^2 + a_{\min}^2 \\ &= \big\langle [x(t) - y(t)]^2 \big\rangle - \langle x(t) - y(t) \rangle^2. \end{aligned} \tag{3.15}$$

Apart from the factor $\frac{1}{T}$, this is precisely the distance $d_{PA}(x, y, t)$ in [15]. By performing the multiple alignment of Eq. (3.10) to obtain new profiles $\hat{x}(t)$ and $\hat{y}(t)$, we have:

$$d(x,y) = \big\langle [\hat{x}(t) - \hat{y}(t)]^2 \big\rangle = \frac{1}{T} \int_0^T \Big[\hat{x}(t) - \hat{y}(t) \Big]^2 dt. \tag{3.16}$$

Thus, $d(x,y)^{\frac{1}{2}}$ is the 2-norm, satisfying all the properties we might want for a metric. On the other hand, it is easy to show that $d(x,y)$ in Eq. (3.16) does not satisfy the triangle inequality, and hence it is not a metric. We, however, use $d(x,y)$ in Eq. (3.16) as our distance function, since it is algebraically easier to work with than the metric $d(x,y)^{\frac{1}{2}}$. Eq. (3.16) is closer to the spirit of regression analysis, and thus, we can dispense with the requirement for the triangle inequality. Also the distance as defined in Eq. (3.16) is unchanged by an additive shift, and hence, is order-preserving; that is: $d(u,v) \leq d(x,y)$ if and only if $d(\hat{u}, \hat{v}) \leq d(\hat{x}, \hat{y})$. This property has important implications for distance-based clustering methods that rely on pairwise alignments of profiles; as discussed later in the experiment chapter.

With the spline interpolations of Eq. (3.5), we derived the analytical solution for $d(x, y)$ in Eq. (3.16), using the symbolic computational package, *Maple**, as follows:

$$
\begin{aligned}
d(x,y) \;=\; & \frac{P^2(n^7-m^7)}{7} + \frac{(2PQ-6P^2m)(n^6-m^6)}{6} + \frac{(2PR-10PQm+Q^2+15P^2m^2)(n^5-m^5)}{5} + \\
& \frac{(-8PRm-4Q^2m+2PS+20PQm^2+2QR-20P^2m^3)(n^4-m^4)}{4} + \\
& \frac{(-6QRm-20Pm^3Q+R^2+6Q^2m^2+12Pm^2R-6PmS+15P^2m^4+2QS)(n^3-m^3)}{3} + \\
& \{\frac{(10Pm^4Q+6Qm^2R+2RS-8Pm^3R-2R^2m-6P^2m^5+6Pm^2S-4QmS-4Q^2m^3)}{2} * \\
& (n^2-m^2)\} - 2RmS(n-m) + S^2(n-m) + P^2m^6(n-m) + Q^2m^4(n-m) + \\
& R^2m^2(n-m) - 2Qm^3R(n-m) - 2Pm^5Q(n-m) - 2Pm^3S(n-m) + \\
& 2Pm^4R(n-m) + 2Qm^2S(n-m)
\end{aligned}
\tag{3.17}
$$

where $P = (x_{j3} - y_{j3})$, $Q = (x_{j2} - y_{j2})$, $R = (x_{j1} - y_{j1})$, $S = (x_{j0} - y_{j0} + c_y - c_x)$, $m = t_j$ and $n = t_{j+1}$.

3.2.4 Centroid of a Cluster

Given a set of profiles $D = \{x_1(t), \ldots, x_s(t)\}$, we aim to find a *centroid profile* $\mu(t)$ that well represents D. An obvious choice is the function that minimizes

$$\Delta[\mu] = \sum_{i=1}^{s} d\left(x_i, \mu\right). \tag{3.18}$$

where Δ plays the role of the *within-cluster-scatter* defined in [15]. Since $d(\cdot, \cdot)$ is unchanged by an additive shift $x(t) \to x(t) - a$ in either of its arguments, we have

$$\Delta[\mu] = \sum_{i=1}^{s} d\left(\hat{x}_i, \mu\right) = \frac{1}{T}\int_0^T \sum_{i=1}^{s} \left[\hat{x}_i(t) - \mu(t)\right]^2 dt, \tag{3.19}$$

where, $\hat{X} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$ is the multiple alignment of Eq. (3.10). This is a *functional* of μ; that is, a mapping from the set of real valued functions

*All the analytical solutions in this paper were derived by Maple.

defined on $[0, T]$ to the set of real numbers. To minimize with respect to μ we set the functional derivative to zero[†]. This functional is of the form

$$F[\phi] = \int L(\phi(t))dt, \tag{3.20}$$

for some function L, for which the functional derivative is simply $\frac{\delta F[\phi]}{\delta \phi(t)} = \frac{dL(\phi(t))}{d\phi(t)}$. In our case, we have

$$\frac{\delta \Delta[\mu]}{\delta \mu(t)} = -\frac{2}{T}\sum_{i=1}^{s}\left[\hat{x}_i(t) - \mu(t)\right] = -\frac{2}{T}\left(\sum_{i=1}^{s}\hat{x}_i(t) - s\mu(t)\right). \tag{3.21}$$

Setting $\frac{\delta \Delta[\mu]}{\delta \mu(t)} = 0$ gives

$$\mu(t) = \frac{1}{s}\sum_{i=1}^{s}\hat{x}_i(t). \tag{3.22}$$

With the spline coefficients, x_{ijk}, of each $x_i(t)$ interpolated as in Eq. (3.11), the analytical solution for $\mu(t)$ in Eq. (3.22) is

$$\mu_j(t) = \frac{1}{s}\sum_{i=1}^{s}\left[\sum_{k=0}^{3} x_{ijk}\,(t - t_j)^k\right] - a_{\min_i}, \quad \text{in each interval } [t_j, t_{j+1}]. \tag{3.23}$$

Eq. (3.22) applies to aligned profiles while Eq. (3.23) can apply to unaligned profiles.

3.3 Alignment Methods for Piecewise Linear Functions

In this chapter, we have proposed pairwise alignment, multiple alignment, distance function and centroid of a cluster for continuous integrable functions

[†]For a functional $F[\phi]$, the functional derivative is defined as $\frac{\delta F[\phi]}{\delta \phi(t)} = \lim_{\epsilon \to 0} \frac{(F[\phi + \epsilon\delta_t] - F[\phi])}{\epsilon}$, where $\delta_t(\tau) = \delta(\tau - t)$ is the Dirac delta function centered at t.

which is Natural Cubic Spline function. All the above theoretical results on Natural Cubic Spline representations including lemmas and theorems are also apply to Piecewise Linear representations of time-series profiles. Clustering algorithms that are proposed in the next chapter also apply to Piecewise Linear representations of time-series profiles.

Chapter 4

Clustering via Continuous Gene Expression Profile Alignment

In both [15] and [25], profiles were represented as piecewise linear functions. Area-based profile alignment takes two features vectors, and produces two new vectors in such a way that the area between "aligned" vectors is minimized. In [15], hierarchical-agglomerative-clustering is used where the decision rule is based on the *furthest-neighbor* or *complete linkage* distance between two clusters. The *complete linkage* or *furthest neighbors* calculates the distance between the furthest pair of points for each pair of clusters and merges the two clusters that have the minimum distance among all such distances between all pairs of clusters under consideration. The proposed clustering algorithms are discussed in this chapter. Validity indices to determine the accuracies of the proposed approaches and to determine the number of clusters are also discussed.

4.1 k-Means Clustering via Multiple Alignment

The k-means algorithm is one of the simplest and fastest clustering algorithms. It takes the number of clusters, k, as an input parameter. The program starts by randomly choosing k points as the centers of the clusters. These points may be just random points from more densely populated volumes of the input space or just randomly chosen patterns from the data itself. Once some cluster centers have been chosen, the algorithm will take each profile and calculate the distance from it to all cluster centers. Since the cluster centers were chosen randomly, it is not said that this is the correct clustering. The second steps starts by considering all profiles associated which one cluster center and calculating a new position for this cluster center. The coordinates of this new center are usually obtained by calculating the mean of the coordinates of the points belonging to that cluster. Since the centers have moved, the profile memberships need to be updated by recalculating the distance from each profile to the new cluster centers. The algorithm continues to update the cluster centers based on the new membership and update the membership of each profile until the cluster centers are such that no profile moves from one cluster to another. Since no profile has changed its membership, the centers will remain the same and the algorithm will terminate. A more formal definition of k-means clustering is stated below.

In k-means [35], we want to partition a set of s profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, into k disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, $1 \leq k \leq s$; such that (i) $\mathcal{C}_i \neq \emptyset, i = 1, \ldots, k$; (ii) $U_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$ (iii) $\mathcal{C}_i \cap \mathcal{C}_j = \phi$; $i \neq j$; $i, j = 1, \ldots, k$. Also, each profile is assigned to the cluster whose mean is the closest. It is similar to EM for

mixtures of Gaussians in the sense that they both attempt to find the centers of natural clusters in the data. It assumes that the object features form a *vector space*. Let $U = \{u_{ij}\}$ be the membership matrix defined as follows:

$$u_{ij} = \begin{cases} 1 & \text{if } d(x_i, \mu_j) = min_{l=1,\ldots,k} d(x_i, \mu_l) \; where \quad i = 1, \ldots, s \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The aim of k-means is to minimize the sum of squared distances:

$$J(\theta, U) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i, \mu_j) . \tag{4.2}$$

where $\theta = \mu_1, \mu_2, \ldots, \mu_n$.

Algorithm 1 *k-MCMA: k-Means Clustering with Multiple Alignment*

Input: Set of profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, and desired number of clusters, k

Output: Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1}, \ldots, \hat{\mathcal{C}}_{\hat{\mu}_k}$

1. Apply natural cubic spline interpolation on $x_i(t) \in \mathcal{D}$, for $1 \leq i \leq k$ (see Section 3.2.1)

2. Multiple-align transformed $\mathcal{D}$ to obtain $\hat{\mathcal{D}} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$, using Eq. (3.10)

3. Randomly initialize centroid $\hat{\mu}_i(t)$, for $1 \leq i \leq k$

repeat

4.a. Assign $\hat{x}_j(t)$ to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ with minimal $d(\hat{x}_j, \hat{\mu}_i)$, for $1 \leq j \leq s$ and $1 \leq i \leq k$

4.b. Update $\hat{\mu}_i(t)$ of $\hat{\mathcal{C}}_{\hat{\mu}_i}$, for $1 \leq i \leq k$

until Convergence: that is, no change in $\hat{\mu}_i(t)$, for $1 \leq i \leq k$

return Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1}, \ldots, \hat{\mathcal{C}}_{\hat{\mu}_k}$

In k-MCMA (see Algorithm. 1), we first multiple-align the set of profiles $\mathcal{D}$, using Eq. (3.10), and then cluster the multiple aligned $\hat{\mathcal{D}}$ with k-means. Recall that the process of Eq. (3.10) is to *pairwise align* each profile with the t-axis. The k initial centroids are found by randomly selecting k pairs of

profiles in $\hat{\mathcal{D}}$, and then take the centroid of each pair. In step (4.a), we do not use pairwise alignment to find the centroid $\hat{\mu}_i(t)$ closest to $\hat{x}_j(t)$, since, by Lemma 3.2.1, they are automatically aligned relative to each other. When profiles are multiple-aligned, any arbitrary distance function other than Eq. (3.16) can be used in step (4.a), including the Euclidean distance. Also, by Theorem 4.1.1 below, there is no need to multiple-align $\hat{\mathcal{C}}_{\hat{\mu}_i}$ in step (4.b), to update its centroid $\hat{\mu}_i(t)$.

Theorem 4.1.1. *Let $\bar{\mu}(t)$ be the centroid of a cluster of m multiple-aligned profiles. Then $\hat{\bar{\mu}}(t) = \bar{\mu}(t)$.*

Proof. We have $\hat{\bar{\mu}}(t) = \bar{\mu}(t) - a_{\min_{\bar{\mu}}}$. However, $a_{\min_{\bar{\mu}}} = \frac{1}{T}\int_0^T \bar{\mu}(t)dt$
$= \frac{1}{T}\int_0^T \frac{1}{m}\sum_{i=1}^{m} \hat{x}_i(t) = 0$, since each $\hat{x}_i(t)$ is aligned with the t-axis. □ □

Thus, Lemma 3.2.1 and Theorem 4.1.1 make k-MCMA much faster than applying k-means directly on the non-aligned dataset $\mathcal{D}$, and even more than this when the Euclidean distance is used to assign a profile to a cluster. An important implication of Eq. (3.16) is that applying k-means on the non-aligned dataset $\mathcal{D}$ (i.e., clustering on $\mathcal{D}$), without any multiple alignment, is equivalent to k-MCMA (i.e., clustering on $\hat{\mathcal{D}}$). That is, if a profile $x_i(t)$ is assigned to a cluster $\mathcal{C}_{\mu_i}$ by k-means on $\mathcal{D}$, its shifted profile $\hat{x}_i(t)$ will be assigned to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ by k-MCMA (k-means on $\hat{\mathcal{D}}$). This can be easily shown by the fact that multiple alignment is order-preserving, as pointed out in Section 3.2.3. In k-means on $\mathcal{D}$, step (4.a) would require $O(sk)$ *pairwise alignments* to assign s profiles to k clusters, whereas no pairwise alignment is needed in k-MCMA. In other words, we show that we can multiple-align *once*, and obtain the *same* k-means clustering results, provided that we initialize the

means in the same manner. This also reinforces a known fact demonstrated in [33], which is a dissimilarity function that is not metric can be made metric by using a shift operation (in our case any metric can be used in step (4.a) such as the Euclidean distance). In this case, the objective function of k-means does not change, and convergence is assured. Thus, this saves a lot of computations and opens the door for applications of multiple alignment methods to many distance-based clustering methods.

4.2 EM Clustering via Multiple Alignment

In [18], we devised a clustering approach, k-MCMA, where we combined the multiple alignment of Eq. (3.10) and the k-means clustering method with a distance function based on the pairwise alignment of Eq. (3.3). In this section, we use the EM clustering algorithm instead and combine it with the alignment methods.

EM is used for clustering in the context of mixture models [2]. The goal of EM clustering is to estimate the means and standard deviations for each cluster so as to maximize the likelihood of the observed data (distribution). In other words, the EM algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters. A mixture of Gaussians is a set of k probability distributions, where each distribution represents a cluster. With an initial approximation of the cluster parameters, it iteratively performs two steps: first, the *expectation* step computes the values expected for the cluster probabilities, and second, the *maximization* step computes the distribution and their likelihood. It iterates

until the log-likelihood reaches a (possibly local) maximum. The algorithm is similar to k-means in the sense that the centers of the natural clusters in the data are re-computed until a desired convergence is achieved.

In EM [35], we want to partition a set of s profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, into k disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, $1 \leq k \leq s$, such that; (i) $\mathcal{C}_i \neq \emptyset, i = 1, \ldots, k$; (ii) $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$; (iii) $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, i, j = 1, \ldots, k$ and $i \neq j$. Let $\mathcal{D}$ be the complete-data space drawn independently from the mixture density:

$$\textbf{E-step: } p(x|\theta) = \sum_{i=1}^{k} p(x|\mathcal{C}_i, \theta_i) P(\mathcal{C}_i) \tag{4.3}$$

where parameter $\theta = [\theta_1, \ldots, \theta_k]^t$ is fixed but unknown, and $P(\mathcal{C}_i)$ is the known posterior probability of class $\mathcal{C}_i$. The aim is to maximize the likelihood:

$$\textbf{M-step: } p(D|\theta) = \prod_{e=1}^{s} p(x_e|\theta) \tag{4.4}$$

To maximize the likelihood function, log-likelihood is used in the normal distribution of the component densities given by: $p(x_k|\mathcal{C}_i, \theta_i) \sim N(\mu_i, \Sigma_i)$ where $\theta_i = [\mu_i, \Sigma_i]^t$; μ_i and Σ_i are the means and the covariances of the classes, respectively. Both steps iterate until the log-likelihood reaches a maximum. Thus, EM assigns profiles to multiple clusters, like in *fuzzy* clustering. Also, unlike in k-means, each profile is assigned to the cluster that finds the maximum posterior probability.

In EMMA (see Algorithm 2), we first multiple-align the set of profiles $\mathcal{D}$, using Eq. (3.10), and then cluster the multiple-aligned $\hat{\mathcal{D}}$ with EM. Recall that the process of Eq. (3.10) is to *pairwise align* each profile with the t-axis. The k centroids can be initialized randomly in step (3) of EMMA, or by any initialization approach. However, to obtain better clustering results with

Algorithm 2 *EMMA: EM Clustering with Multiple Alignment*

Input: Set of profiles, $\mathcal{D} = \{x_1(t), \ldots, x_s(t)\}$, and desired number of clusters, k

Output: Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1}, \ldots, \hat{\mathcal{C}}_{\hat{\mu}_k}$

1. Apply natural cubic spline interpolation on $x_i(t) \in \mathcal{D}$, for $1 \leq i \leq k$ (see Section 3.2.1)
2. Multiple-align transformed $\mathcal{D}$ to obtain $\hat{\mathcal{D}} = \{\hat{x}_1(t), \ldots, \hat{x}_s(t)\}$, using Eq. (3.10)
3. Initialize centroid $\hat{\mu}_i(t)$, for $1 \leq i \leq k$
4. Compute the initial log-likelihood (see Eq. (4.4))

repeat

1. **E-step:** $p(x|\theta) = \sum_{i=1}^{k} p(x|\hat{\mathcal{C}}_{\hat{\mu}_i}, \theta_i) P(\hat{\mathcal{C}}_{\hat{\mu}_i})$
2. Assign $\hat{x}_j(t)$ to cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ with maximum log-likelihood, for $1 \leq j \leq s$ and $1 \leq i \leq k$
3. **M-step:** $p(D|\theta) = \prod_{e=1}^{s} p(x_e|\theta)$

until The log-likelihood reaches its maximum

return Clusters $\hat{\mathcal{C}}_{\hat{\mu}_1}, \ldots, \hat{\mathcal{C}}_{\hat{\mu}_k}$

EMMA, it is necessary to start with near-optimal centroids; thus, we applied the k-MCMA algorithm of [18] to generate the k initial centroids in step (3).

For distance-based clustering algorithms such as the k-means method, it has been shown that any arbitrary distance function (including Euclidean) can be used in the cluster assignment step of the algorithm, when the profiles are multiple-aligned first [18]. Moreover, Theorem 4.1.1 showed that, for distance-based methods, there is no need to multiple-align $\hat{\mathcal{C}}_{\hat{\mu}_i}$ to update its centroid $\hat{\mu}_i(t)$.

By Theorem 4.1.1, there is also no need to multiple-align a cluster $\hat{\mathcal{C}}_{\hat{\mu}_i}$ in step (1) of EMMA, for updating its centroid $\hat{\mu}_i(t)$. Likewise, any arbitrary distance function can be used in step (1), for computing the centroids. Thus,

Theorem 4.1.1 makes EMMA run much faster than applying EM directly on the non-aligned data set $\mathcal{D}$. EMMA is not a distance-based clustering method, nevertheless the quantities $p(x|\theta)$, $p(x|\hat{\mathcal{C}}_{\hat{\mu}_i}, \theta_i)$, $P(\hat{\mathcal{C}}_{\hat{\mu}_i})$, and $p(D|\theta)$ are also preserved when the distances are preserved. In other words, we show that we can multiple-align *once*, and obtain the *same* EM clustering results, provided that we initialize the means in the same manner.

4.3 Assessment of Clustering Quality

The two fundamental properties, number of clusters and the goodness of the clustering itself, need to be determined in any typical clustering system. To determine the appropriate number of clusters and also the goodness or validity of the resulting clusters, we have run our k-MCMA algorithm in conjunction with four cluster validity indices [17], namely Davies-Bouldin's index, Dunn's index, Calinski-Harabasz's index, and index $\mathcal{I}$. Once the appropriate number of clusters is determined, k-MCMA is used for proper partitioning of the data into the said number of clusters. Let K be the number of clusters.

Davies-Bouldin's (DB) index is a function of the ratio of the sum of *within-cluster scatter* to *between-cluster separation*. The *within-cluster scatter* for the ith cluster, S_i, and the distance between clusters is $d_{ij} = ||\mu_i - \mu_j||$. Then, the Davies-Bouldin (DB) index is defined as follows:

$$DB = \frac{1}{K} \sum_{i=1}^{K} \max_{j, j \neq i} \frac{S_{i,q} + S_{j,q}}{d_{ij,t}}. \tag{4.5}$$

The objective is to minimize the DB index for achieving the best clustering.

Dunn's index: Let S and T be two nonempty subsets of $\mathcal{R}^N$. The diameter $\triangle$ of S is defined as $\triangle(S) = \max_{x,y\epsilon S}\{d(x,y)\}$ and the distance δ between S and T is defined as $\delta(S,T) = \min_{x\epsilon S, y\epsilon T}\{d(x,y)\}$, where $d(x,y)$ denotes the distance between x and y. Then, Dunn's index defined as follows:

$$Dunn = \min_{1\leq i\leq K}\{\min_{1\leq j\leq K, j\neq i}\{\frac{\delta(C_i,C_j)}{\max_{1\leq k\leq K}\{\triangle(C_k)\}}\}\}. \tag{4.6}$$

The number of clusters that maximizes *Dunn* is taken as the optimal number of clusters.

Calinski Harabasz's (CH) index: CH index is defined as follows:

$$CH = \frac{[traceB/(K-1)]}{[traceW/(n-K)]}, \tag{4.7}$$

where B is the *between-cluster* matrix and W the *within-cluster scatter* matrix. The maximum level is used to indicate the correct number of clusters in the data.

$\mathcal{I}$ index: The $\mathcal{I}$ index is defined as follows:

$$\mathcal{I}(K) = (\frac{1}{K} \times \frac{E_1}{E_K} \times D_K)^p, \tag{4.8}$$

where $E_K = \sum_{k=1}^{K}\sum_{j=1}^{n} u_{kj}||x_j - z_k||$, $D_K = \max_{i,j=1}^{K}||z_i - z_j||$. $U(X) = [u_{kj}]_{K\times n}$ is a partition matrix for the data, and z_k is the centroid of the kth cluster. The number of clusters that maximizes $\mathcal{I}(K)$ is considered to be the correct number of clusters. We have taken $p = 2$, which is used to control the contrast between the different cluster configurations. This index is typically used in many applications.

To find the best number of clusters, we applied k-MCMA and EMMA on a data set in conjunction with the four above-discussed validity indices for

$k = 1, \ldots, \sqrt{s}$ (where s is the number of profiles) clusters. Among the four validity indices, we see which number is the most frequent and that number is chosen as the best number of clusters for that data set.

4.4 Cluster Visualization

To visualize the clusters with pre-clustered yeast phases, we face a combinatorial assignment problem. We assigned each k-MCMA and EMMA cluster to a yeast phase using the *Hungarian algorithm* [13]. The Hungarian method is a combinatorial optimization algorithm which solves the assignment problem in polynomial time. Our phase assignment problem is formulated using a complete bipartite graph $G = (C, P, E)$ with k cluster vertices (C) and k phases vertices (P), and each edge in E has a nonnegative cost $c(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$, $\hat{\mathcal{C}}_{\hat{\mu}_i} \in C$ and $\hat{\mathcal{P}}_{\hat{\nu}_j} \in P$. We want to find a perfect matching with minimum cost. The cost of an edge between a cluster vertex $\hat{\mathcal{C}}_{\hat{\mu}_i}$ and a phase vertex $\hat{\mathcal{P}}_{\hat{\nu}_j}$ is the distance between their centroids $\hat{\mu}_i, \hat{\nu}_j$; that is $c(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j}) = d(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$, and the distances are computed using Eq. (3.16). In terms of such a bipartite graph, the Hungarian method will select the k perfect matching pairs $(\hat{\mathcal{C}}_{\hat{\mu}_i}, \hat{\mathcal{P}}_{\hat{\nu}_j})$ with minimum cost. In Fig. 5.2, the cluster and the phase of each of the five selected pairs, found by the Hungarian algorithm, are shown at the same level; e.g., cluster *C5* of k-MCMA is assigned to the *Late G1* phase of [22] by our phase assignment approach, and hence they are at the same level in the figure.

Chapter 5

Computational Experiments

We have experimented our approaches on six well-known data sets. First, we normalized the data sets as did by the authors of data sources. Second, we represent profiles to natural cubic spline (NCS) profiles and multiple align them. Third, using these multiple-aligned data set as input in one of our k-MCMA or EMMA approaches in conjunction with validity indices ran to determine the number of clusters. Forth, after obtaining number of clusters we performed the clustering using k-MCMA and EMMA approaches, respectively. Fifth, for comparison, an average classification accuracy was computed by performing a c-Nearest neighbor (c-NN) classifier with clusters to classify the data with a 10-fold cross validation procedure where c is the number of nearest profiles from the centroids. At the end, we also compare our approaches with previously published approaches.

5.1 Data Description

Six groups of data have been used in the experiments: *Saccharomyces cerevisiae* budding yeast data set, *Pseudomonas aeruginosa* transcriptomes data set, Serum data set obtained from [32], *Micrococcus luteus* infection challenge to *Anopheles gambiae* data set, *Escherichia coli* infection challenge to *Anopheles gambiae* data set and *Schizosaccharomyces pombe* data set of cdc25 mutant cell gene expression profiles.

Expression level represents a gene that express to certain protein(s) which make the whole or part of biological process behave in a particular way. This expression is measured in the microarray and is being reflected in the quantification process. The expression ration is the comparison of two expression levels, namely "normal" vs. "abnormal", or "control" vs. experiment. In our experiments, *Saccharomyces cerevisiae* data set and *Schizosaccharomyces pombe* data set contain the expression levels and rest of the data sets are expression ratios. More precisely, typical microarray experiment involves five processing stages target preparation, design DNA chips for targeted genes, hybridization, detecting expression levels or ratios, and data preparation for analysis. Data preparation stage itself involves in certain steps like image processing, quantification, data pre-processing, normalizations and data representation for analysis. In whole data preparation process, every steps does some sort of "corrections". The normalization processes help identify the abnormalities of gene expression levels. To increase the reliability of the experiments, *Saccharomyces cerevisiae* and Serum data set were made in replicates where as *Schizosaccharomyces pombe* data set of cdc25 mutant cell gene raw

expression levels. Rest of the three data sets were made in triplicates.

The detail description and the normalization procedure of each data sets are described below in their specific sub-section.

5.1.1 Saccharomyces cerevisiae Data Set

A data set of pre-clustered genes of budding yeast, *Saccharomyces cerevisiae*, [22]* is discussed in this section. The data set contains time-series gene expression profiles of the complete characterization of mRNA transcript levels during the yeast cell cycle. These experiments measured the expression levels of the 6,220 yeast genes during the cell cycle at seventeen different points, from 0 to 160 minutes, at every 10-minute time-interval. From those gene profiles, 221 profiles were analyzed. We normalized each expression profile as described in [22]; that is, we divided each transcript level by the mean value of each profile with respect to each other.

The data set contains five *known* clusters called *phases*: Early G1 phase (32 genes), Late G1 phase (84 genes), S phase (46 genes), G2 phase (28 genes) and M phase (31 genes).

5.1.2 Pseudomonas aeruginosa Data Set

We have also experimented on another data set of 3315 *Pseudomonas aeruginosa* bacterium gene expression ratios of [34]. These experiments measured the expression ratios of *Pseudomonas aeruginosa* genes during the planktonic cultures at 0, 4, 8, 14, 24 and 48 hours time points. Expressions are averaged

* `http://genomics.stanford.edu/yeast_cell_cycle/cellcycle.html`

values of the three replicates for each condition, and then normalized to zero mean and unit variance.

5.1.3 Serum Data Set

The serum data set contains data on the transcriptional response of cell cycle-synchronized human fibroblasts to serum. These experiments have measured the expression levels of 8,613 human genes after a serum stimulation at twelve different time points, at 0 hr., 15 min., 30 min., 1 hr., 2 hrs., 3 hrs., 4 hrs., 8 hrs., 16 hrs., 20 hrs. and 24 hrs. From these 8,613 gene profiles, 517 profiles were separately analyzed, as their expression ratio has changed substantially at two or more time points. The experiments and analysis have focused on this dataset, which is the same group of 517 genes used in [32].

5.1.4 Micrococcus luteus Data Set

Immune responses of the malaria vector mosquito *Anopheles gambiae* were monitored systematically by the induced expression of five RNA markers after *M. luteus* infection challenge. Bacterial infection of third and fourth instar larvae and adult female mosquitoes were performed by pricking with a needle dipped in a concentrated solution of *M. luteus* at seven different time points, at 1 hr., 4 hrs., 8 hrs., 12 hrs., 18 hrs., and 24 hrs. Expression values were the log-2 transformed, normalized ratios of medians, as described in Materials and Methods in [9].

5.1.5 Escherichia coli Data Set

Immune responses of the malaria vector mosquito *Anopheles gambiae* were monitored systematically by the induced expression of five RNA markers after *Escherichia coli* infection challenge. Bacterial infection of third and fourth instar larvae and adult female mosquitoes were performed by pricking with a needle dipped in a concentrated solution of *E. coli* at seven different time points, at 1 hr., 4 hrs., 8 hrs., 12 hrs., 18 hrs., and 24 hrs. Expression values were the log-2 transformed, normalized ratios of medians, as described in Materials and Methods in [9].

5.1.6 Schizosaccharomyces pombe Data Set

Data set containing the cell cycle progressions of the fission yeast *Schizosaccharomyces pombe* prepared by Peng et al [8]. This data set contains 747 genes and two types of cell, namely, wild-type and cdc25 mutant cells. We have used the cdc25 type mutant cells genes.

5.2 Experimental Results

We performed experiments to show the clustering capabilities of the proposed algorithms combined with presented alignment approaches in this thesis. All programs were written in Matlab (Version 7.7.0.471 (R2008b)) and all tests were run on an Intel $Core^{TM}$ 2 Duo CPU 1.83GHz with 2GB of RAM under Windows $Vista^{TM}$ Home Premium with Service Pack 1.

5.2.1 Experimental Results of k-MCMA

Experiments have been carried out using k-MCMA in conjunction with four validity indices, described in Section 4.3 on all the above five data sets to determine the number of clusters. After determining the number of clusters for each data sets, actual clusterings were performed using k-MCMA algorithm. To see the clusters visually, k-MCMA clusters of each data set were graphically presented.

Table 5.1 shows the results of k-MCMA in conjunction with four cluster validity indices for 2 to $\sqrt{221} = 14$ clusters on the *Saccharomyces cerevisiae* data set.

Clusters	DB	CH	Dunn	I-index
2	2.141	36.371	0.927	0.024
3	1.364	53.554	0.909	0.023
4	1.308	61.814	1.148	0.045
5	1.582	54.142	0.829	0.038
6	1.647	47.213	0.847	0.020
7	1.948	32.041	0.335	0.065
8	1.536	41.560	0.554	0.032
9	1.718	39.859	0.695	0.087
10	1.705	35.074	0.608	0.100
11	1.715	33.985	0.697	0.022
12	1.686	30.799	0.623	0.038
13	1.774	30.175	0.427	0.026
14	1.568	30.984	0.623	0.030

Table 5.1: Validity index values for k-MCMA clusters on the *Saccharomyces cerevisiae* data set.

Table 5.2 shows the results of k-MCMA in conjunction with four cluster validity indices for 2 to $\sqrt{3315} = 57$ clusters on the *Pseudomonas aeruginosa* bacterium data set. But we are only showing up to 50 clusters because it

is not significant to show all the validity indices when actual the number of clusters lies between 7 to 10.

Clusters	DB	CH	Dunn	I-index	Clusters	DB	CH	Dunn	I-index
2	1.331	1094.649	1.239	0.299	27	1.516	618.003	0.290	0.537
3	1.611	783.397	0.912	0.981	28	1.434	571.141	0.227	0.733
4	2.157	578.282	0.579	1.097	29	1.643	535.542	0.359	0.571
5	1.224	1316.982	1.026	0.734	30	1.592	560.812	0.354	0.663
6	1.203	1314.547	0.819	0.633	31	1.487	537.927	0.367	0.719
7	1.245	1362.184	0.953	1.391	32	1.592	554.643	0.336	0.624
8	1.436	1092.089	0.690	0.575	33	1.711	476.496	0.234	0.675
9	1.356	1010.209	0.641	0.912	34	1.573	522.204	0.262	1.134
10	1.454	960.256	0.543	1.080	35	1.681	528.118	0.261	0.790
11	1.484	1007.263	0.571	1.344	36	1.620	528.575	0.233	0.848
12	1.625	880.004	0.475	0.892	37	1.672	516.902	0.300	0.773
13	1.679	830.367	0.454	1.095	38	1.684	491.492	0.195	0.715
14	1.580	907.636	0.462	0.674	39	1.517	511.396	0.321	0.623
15	1.788	687.293	0.454	0.636	40	1.475	531.148	0.280	0.508
16	1.477	726.658	0.443	1.226	41	1.493	488.864	0.339	0.576
17	1.508	793.470	0.484	1.126	42	1.695	439.103	0.291	0.531
18	1.403	773.875	0.435	0.620	43	1.678	483.437	0.283	0.532
19	1.611	580.932	0.470	0.560	44	1.499	487.624	0.387	0.566
20	1.626	711.419	0.405	0.995	45	1.598	445.929	0.306	0.698
21	1.467	646.555	0.566	0.648	46	1.500	489.593	0.289	0.897
22	1.524	708.820	0.521	0.799	47	1.572	460.559	0.315	0.485
23	1.448	713.085	0.439	0.791	48	1.545	431.215	0.290	0.511
24	1.523	628.134	0.403	0.976	49	1.638	425.329	0.259	0.594
25	1.465	653.459	0.374	0.504	50	1.539	444.991	0.259	0.539

Table 5.2: Validity index values for k-MCMA clusters on the *Pseudomonas aeruginosa* data set.

Table 5.3 shows the results of k-MCMA in conjunction with four cluster validity indices for 2 to $\sqrt{517} = 22$ clusters on the *serum* data set. Table 5.4 shows the results of k-MCMA in conjunction with four cluster validity indices for 2 to $\sqrt{652} = 25$ clusters on the *Micrococcus luteus* data set. Table 5.5 shows the results of k-MCMA in conjunction with four cluster validity indices

for 2 to $\sqrt{652} = 25$ clusters on the *Escherichia coli* data set.

Clusters	DB	CH	Dunn	I-index
2	1.052	163.868	1.054	1.805
3	2.334	32.858	0.244	6.558
4	1.538	45.265	0.468	2.183
5	1.502	38.046	0.089	2.897
6	1.749	35.567	0.047	1.819
7	1.475	30.800	0.235	1.100
8	1.621	43.192	0.021	0.531
9	1.620	20.821	0.129	0.084
10	2.007	20.057	0.065	1.283
11	1.852	25.363	0.032	3.554
12	1.988	31.458	0.016	3.795
13	2.001	28.543	0.018	3.998
14	1.783	15.847	0.063	1.283
15	1.864	16.474	0.038	0.398
16	1.675	27.160	0.035	0.624
17	1.867	21.537	0.016	0.234
18	1.693	99.103	0.038	0.872
19	1.725	21.985	0.022	3.360
20	1.960	17.778	0.016	4.104
21	1.710	13.364	0.032	2.483
22	1.649	12.436	0.031	1.573

Table 5.3: Validity index values for k-MCMA clusters on the *serum* data set.

5.2.2 Experimental Results of EMMA

Experiments have been carried out using EMMA in conjunction with four validity indices, stated in Section 4.3 on all the above five data sets to determine the number of clusters. After determining the number of clusters for each data set, actual clusterings were performed using EMMA. To see the clusters visually, EMMA clusters of each data set are graphically presented.

Clusters	DB	CH	Dunn	I-index
2	1.821	88.074	1.052	0.002
3	2.544	73.761	0.617	0.003
4	1.548	93.396	0.670	0.018
5	1.810	76.600	0.710	0.007
6	1.807	60.925	0.509	0.016
7	1.947	60.216	0.608	0.004
8	1.816	61.984	0.651	0.008
9	1.623	60.466	0.381	0.062
10	1.946	50.403	0.451	0.011
11	1.763	53.020	0.426	0.013
12	1.758	53.028	0.356	0.014
13	1.849	49.483	0.353	0.024
14	1.806	42.747	0.418	0.011
15	1.668	47.393	0.419	0.014
16	1.762	47.600	0.358	0.023
17	1.796	43.373	0.343	0.009
18	1.614	42.831	0.325	0.015
19	1.713	41.569	0.367	0.008
20	1.765	42.003	0.273	0.010
21	1.783	37.275	0.334	0.017
22	1.670	40.124	0.323	0.016
23	1.677	38.574	0.289	0.012
24	1.721	35.604	0.322	0.012
25	1.632	39.063	0.391	0.015

Table 5.4: Validity index values for k-MCMA clusters on the *Micrococcus luteus* data set.

Table 5.6 shows the results of EMMA in conjunction with four cluster validity indices for $k = 2$ to $\sqrt{221} = 14$ clusters on the *Saccharomyces cerevisiae* data set.

Table 5.7 shows the results of EMMA in conjunction with four cluster validity indices for 2 to $\sqrt{3315} = 57$ clusters on the *Pseudomonas aeruginosa* data set. But we are only showing up to 50 clusters because it is not significant to show all the validity indices when actual number of clusters lies between 7

Clusters	DB	CH	Dunn	I-index
2	1.637	55.288	1.085	0.007
3	1.678	110.924	0.793	0.014
4	1.880	79.309	0.521	0.010
5	2.223	62.590	0.584	0.002
6	1.747	92.469	0.614	0.006
7	2.084	55.623	0.430	0.018
8	2.003	70.878	0.336	0.008
9	1.763	66.573	0.418	0.014
10	1.752	56.926	0.665	0.002
11	1.751	56.824	0.357	0.006
12	1.995	51.193	0.373	0.004
13	1.911	43.066	0.338	0.001
14	1.638	58.905	0.447	0.006
15	1.659	58.720	0.290	0.009
16	1.787	40.065	0.308	0.004
17	1.829	50.900	0.295	0.027
18	1.841	39.892	0.348	0.001
19	1.715	38.456	0.292	0.007
20	1.964	45.139	0.162	0.012
21	1.747	42.812	0.249	0.006
22	1.767	36.968	0.326	0.006
23	1.613	45.675	0.254	0.012
24	1.876	34.020	0.307	0.002
25	1.642	43.876	0.186	0.006

Table 5.5: Validity index values for k-MCMA clusters on the *Escherichia coli* data set.

and 10.

Table 5.8 shows the results of EMMA in conjunction with four cluster validity indices for $k = 2$ to $\sqrt{517} = 22$ clusters on the *serum* data set. Table 5.9 shows the results of EMMA in conjunction with four cluster validity indices for $k = 2$ to $\sqrt{652} = 25$ clusters on the *Micrococcus luteus* data set. Table 5.10 shows the results of EMMA in conjunction with four cluster validity indices for $k = 2$ to $\sqrt{652} = 25$ clusters on the *Escherichia coli* data set. Table 5.11

Clusters	DB	CH	Dunn	I-index
2	1.365	96.985	1.283	0.024
3	1.461	77.554	1.109	0.035
4	1.218	73.484	1.152	0.070
5	1.616	60.302	0.546	0.101
6	1.376	56.358	0.964	0.029
7	1.339	47.253	0.781	0.053
8	1.571	47.564	0.683	0.057
9	1.872	40.628	0.446	0.078
10	1.479	33.609	0.572	0.042
11	1.514	38.366	0.000	0.071
12	1.213	32.077	0.000	0.039
13	1.773	29.696	0.000	0.028
14	1.785	29.078	0.000	0.052

Table 5.6: Validity index values for EMMA clusters on the *Saccharomyces cerevisiae* data set.

shows the results of EMMA in conjunction with four cluster validity indices for $k = 2$ to $\sqrt{747} = 27$ clusters on the *Schizosaccharomyces pombe* data set.

5.3 Analysis and Discussion

After analyzing all the validity indices tables, taking the majority and the most frequent value and therefore we choose the best number of clusters for their respective data sets using both k-MCMA and EMMA proposed in this thesis. Table 5.12 shows the results.

Setting $k = 5$, we ran both proposed approaches on *Saccharomyces cerevisiae* data set. Once the clusters have been found, to compare the k-MCMA and EMMA clustering with the pre-clustered dataset of [22], the next step is to label the clusters, where the labels are the "phases" in the pre-clustered dataset. Although this can be done in many different ways, we adopted the

Clusters	DB	CH	Dunn	I-index	Clusters	DB	CH	Dunn	I-index
2	1.104	2266.705	1.673	0.526	27	1.572	515.792	0.225	0.206
3	1.489	1521.172	1.020	0.851	28	1.722	571.660	0.117	0.189
4	1.071	1423.356	0.984	0.842	29	1.735	534.723	0.276	0.168
5	1.172	1138.453	0.583	0.641	30	1.730	524.307	0.219	0.194
6	1.695	1200.588	0.451	0.617	31	1.684	492.669	0.240	0.147
7	1.702	1078.647	0.448	0.787	32	1.763	507.150	0.205	0.188
8	1.824	1012.971	0.251	0.514	33	1.755	469.873	0.261	0.164
9	1.924	901.677	0.259	0.710	34	1.783	492.258	0.147	0.130
10	1.742	912.016	0.393	0.314	35	1.721	465.076	0.205	0.121
11	1.748	857.910	0.374	0.431	36	1.746	464.978	0.239	0.140
12	1.464	813.231	0.435	0.468	37	1.592	463.270	0.279	0.153
13	1.472	780.632	0.332	0.410	38	1.808	470.246	0.212	0.464
14	1.662	748.058	0.200	0.356	39	1.737	479.956	0.190	0.157
15	1.700	811.941	0.162	0.263	40	1.839	433.889	0.189	0.144
16	1.810	710.175	0.156	0.391	41	1.793	429.553	0.196	0.144
17	1.655	689.087	0.184	0.321	42	1.721	476.158	0.139	0.161
18	1.813	630.497	0.220	0.247	43	1.670	400.140	0.196	0.118
19	1.767	604.265	0.224	0.349	44	1.662	483.996	0.229	0.094
20	1.663	629.283	0.247	0.281	45	1.708	428.435	0.228	0.246
21	1.691	599.885	0.264	0.202	46	1.769	423.514	0.188	0.118
22	1.839	585.073	0.152	0.238	47	1.782	394.460	0.172	0.119
23	1.803	538.253	0.174	0.308	48	1.677	432.974	0.261	0.155
24	1.916	532.247	0.129	0.205	49	1.581	433.622	0.206	0.122
25	1.777	560.800	0.180	0.224	50	1.670	420.860	0.258	0.108

Table 5.7: Validity index values for EMMA clusters on the *Pseudomonas aeruginosa* data set.

following approach. The five clusters found by EMMA are shown in Fig. 5.2(a) and those found by k-MCMA are shown in Fig 5.2(c), while the corresponding phases of [22] after the phase assignment (assignment is found by *Hungarian algorithm* which procedure can be found in Section 4.4) are shown in Fig. 5.2(b). The horizontal axis represents the time-points in minutes and the vertical axis represents the expression values. Each cluster is vertically shifted by six units up, in order to distinguish them visually. The dashed black

Clusters	DB	CH	Dunn	I-index
2	1.733	42.266	0.664	0.013
3	1.938	50.972	0.166	1.512
4	1.288	66.996	0.182	6.043
5	1.724	56.515	0.089	5.037
6	1.342	80.509	0.091	11.498
7	1.598	49.518	0.109	8.270
8	1.443	42.173	0.049	7.333
9	1.336	39.686	0.048	2.083
10	1.492	33.100	0.048	4.391
11	1.484	40.578	0.026	1.857
12	1.496	29.981	0.032	4.097
13	1.535	27.689	0.022	5.213
14	1.594	36.058	0.000	3.041
15	1.533	26.698	0.000	19.272
16	1.433	32.950	0.019	18.895
17	1.304	51.896	0.017	2.642
18	1.428	20.438	0.000	14.331
19	1.259	68.625	0.016	1.712
20	1.583	19.512	0.000	2.823
21	1.597	18.639	0.000	1.740
22	1.562	21.731	0.000	1.404

Table 5.8: Validity index values for EMMA clusters on the *serum* data set.

lines are the *cluster centroids* learned by EMMA (Fig. 5.2(a)) and the *known phase centroids* of the yeast data (Fig. 5.2(b)). In the figure, each cluster and phase were multiple-aligned using Eq. (3.10) to enhance visualization.

Setting $k = 7$, we applied k-MCMA and EMMA on the *Saccharomyces cerevisiae* data set to see if k-MCMA and EMMA are able to find these clusters correctly.

The clusters found by k-MCMA and EMMA are shown in Fig. 5.2(a) and Fig. 5.2(b), respectively. The horizontal axis represents the time points in hours and the vertical axis represents the expression ratios. The k-MCMA

Clusters	DB	CH	Dunn	I-index
2	2.041	99.820	0.870	0.000
3	3.918	46.920	0.225	0.001
4	1.945	90.408	0.563	0.010
5	3.032	54.024	0.297	0.002
6	3.122	48.059	0.228	0.006
7	2.987	46.685	0.197	0.007
8	2.232	46.035	0.232	0.019
9	2.343	43.697	0.201	0.007
10	1.822	51.275	0.314	0.006
11	1.929	48.783	0.206	0.013
12	2.027	50.673	0.259	0.015
13	1.833	54.198	0.303	0.012
14	2.497	35.252	0.199	0.004
15	1.629	49.106	0.306	0.005
16	1.618	50.318	0.335	0.009
17	1.906	32.792	0.231	0.006
18	1.728	40.035	0.247	0.011
19	1.772	35.893	0.169	0.006
20	1.742	46.854	0.215	0.006
21	1.617	40.290	0.242	0.012
22	1.920	34.470	0.142	0.013
23	1.623	35.566	0.183	0.011
24	1.820	35.006	0.060	0.005
25	1.708	42.872	0.035	0.009

Table 5.9: Validity index values for EMMA clusters on the *Micrococcus luteus* data set.

and EMMA clusters in Fig. 5.2(a) and (b) are vertically shifted by 6 points up to distinguish them visually. The dashed black lines are the *cluster centroids* learned by EMMA (Fig. 3(c)) and the *centroids* of the bacterium data computed using Eq. (3.10).

The performance of the *Multiple-Alignment* method on the *Saccharomyces cerevisiae* data set is discussed in this section. The Multiple alignment method of Section 3.2.2 we have used in k-MCMA and EMMA is based on natural

Clusters	DB	CH	Dunn	I-index
2	7.137	10.629	0.196	0.000
3	5.061	41.568	0.159	0.013
4	3.561	40.357	0.201	0.008
5	3.341	40.395	0.203	0.009
6	2.819	45.872	0.250	0.004
7	2.163	62.821	0.257	0.006
8	3.117	35.060	0.149	0.006
9	2.264	55.137	0.213	0.005
10	2.635	36.678	0.166	0.002
11	2.104	37.027	0.223	0.017
12	3.887	25.563	0.133	0.017
13	2.404	35.654	0.117	0.002
14	2.600	30.571	0.124	0.012
15	1.796	40.118	0.153	0.004
16	2.358	35.756	0.181	0.007
17	2.325	32.344	0.127	0.006
18	2.197	29.198	0.000	0.010
19	1.999	35.955	0.174	0.009
20	2.126	27.307	0.121	0.012
21	1.932	36.178	0.142	0.037
22	1.977	30.745	0.165	0.002
23	1.704	35.493	0.155	0.003
24	2.481	27.217	0.000	0.005
25	2.073	27.884	0.114	0.010

Table 5.10: Validity index values for EMMA clusters on the *Escherichia coli* data set.

cubic spline profiles. We extend the pairwise alignment formulae of [15] for piecewise linear profiles to multiple expression profile alignment. The procedure for extending pairwise profile alignment to multiple profile alignment is similar to the one described in [18], except that in this case we have used piecewise linear profiles instead of natural cubic spline profiles. We combine k-means clustering with our multiple alignment approaches to cluster microarray time-series data.

Clusters	DB	CH	Dunn	I-index
2	2.455	88.186	0.543	0.005
3	2.694	88.873	0.361	0.007
4	2.410	71.351	0.321	0.006
5	2.466	108.008	0.304	1.259
6	1.654	125.466	0.221	0.177
7	1.543	157.105	0.263	0.499
8	1.205	204.634	0.390	0.782
9	1.314	178.000	0.199	0.764
10	1.574	135.519	0.183	1.257
11	2.328	51.883	0.110	0.184
12	1.410	82.840	0.134	0.105
13	1.748	45.277	0.125	0.021
14	1.412	72.565	0.100	0.094
15	1.491	68.239	0.108	0.096
16	1.460	127.589	0.174	1.169
17	1.542	102.795	0.123	0.332
18	1.936	35.091	0.066	0.049
19	1.470	57.766	0.069	0.075
20	1.447	100.789	0.000	0.520
21	2.410	65.073	0.095	0.355
22	1.443	100.122	0.121	0.478
23	1.609	49.927	0.000	0.052
24	3.258	27.278	0.060	0.027
25	1.678	58.370	0.041	0.114
26	1.396	91.524	0.000	0.243
27	2.043	60.000	0.048	0.388

Table 5.11: Validity index values for EMMA clusters on the *Schizosaccharomyces pombe* data set.

Setting $k = 5$, we applied k-MCMA on the yeast data set for both types of profiles (natural cubic spline and piecewise linear profiles) to find those phases as accurately as possible. Once the clusters have been found, we compared the resulting clusters with those of the pre-clustered dataset of [22]. To achieve a better visual representation, we assigned each cluster obtained by k-MCMA to its corresponding phase in [22]. The cluster-phase matching is done by

Data sets	Number of Clusters
Saccharomyces cerevisiae	5 (DB, CH, I)
Pseudomonas aeruginosa	7 (CH, I)
serum	15 (CH, I)
Micrococcus luteus	8 (CH, I)
Escherichia coli	7 (CH, Dunn)
Schizosaccharomyces pombe	8 (DB, CH)

Table 5.12: Best number of clusters for all data sets.

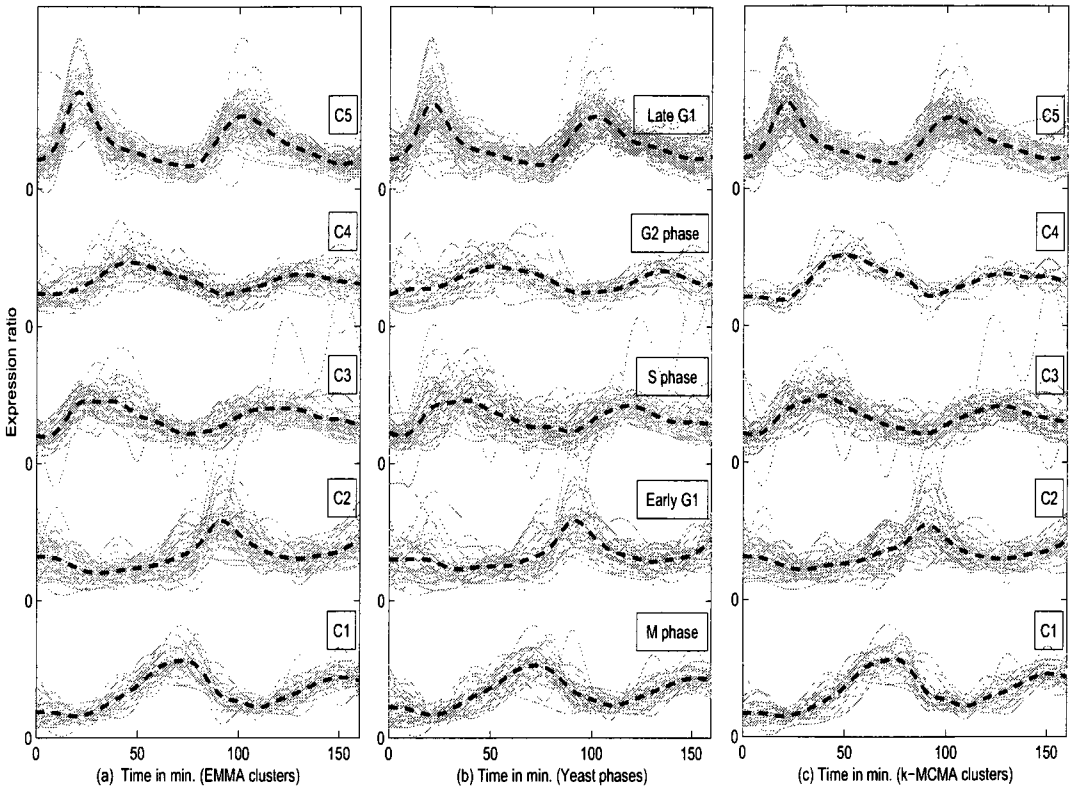

Figure 5.1: (a) EMMA clusters, (b) *Saccharomyce cerevisiae* phases [22] and (c) k-MCMA clusters, with centroids shown.

applying the *Hungarian algorithm* as described in Section 4.4.

The five clusters found by k-MCMA using natural cubic spline profiles are shown in Fig. 5.3(b), while the corresponding phases of [22] after the phase assignment are shown in Fig. 5.3(a). The horizontal axis represents the

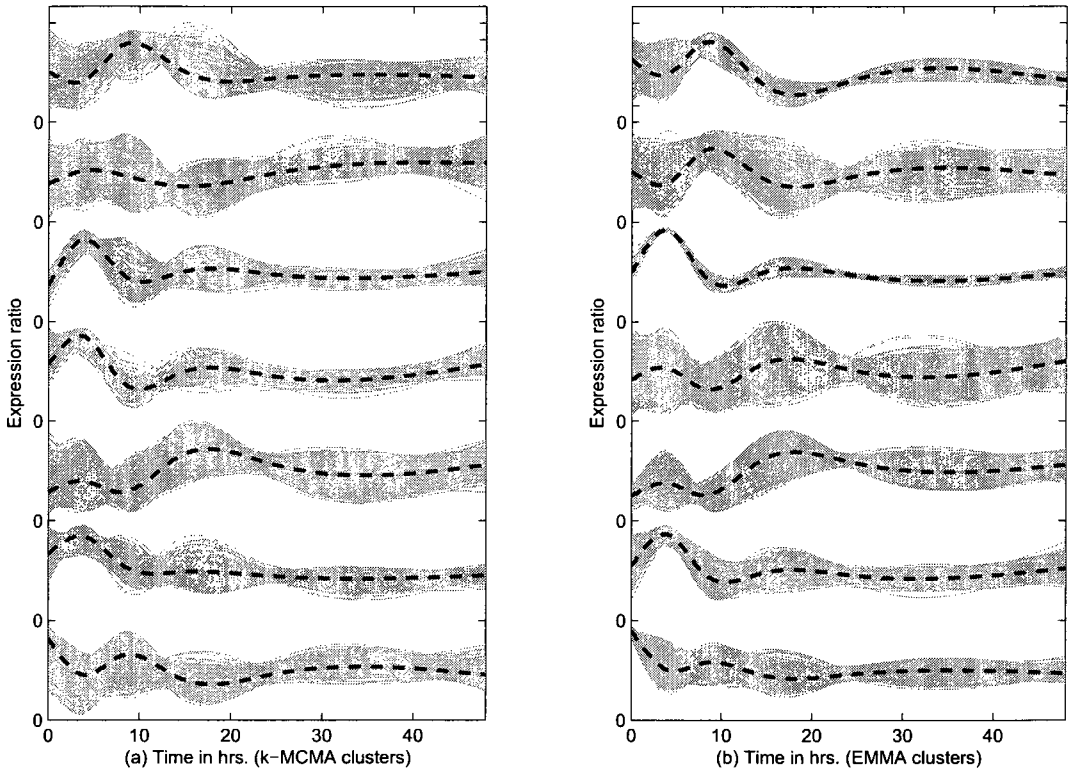

Figure 5.2: (a) k-MCMA clusters and (b) EMMA clusters of *Pseudomonas aeruginosa* data set, with centroids shown.

time points in minutes and the vertical axis represents the expression values. The dashed black lines are the *cluster centroids* learned by k-MCMA (Fig. 5.3(b)), and the *known phase centroids* of the yeast data (Fig. 5.3(a)). Fig. 5.4 shows the clustering on the same data set as Fig. 5.3, where, k-MCMA uses piecewise linear profiles. In the figures, each cluster and phase were vertically shifted by three units up, in order to enhance visualization. After visually representing each cluster in both cases, we see that k-MCMA clusters with both representations are quite similar to exactly one of the yeast phases.

We applied both k-MCMA and EMMA on the *Micrococcus luteus* data set for both types of profiles (natural cubic spline and piecewise linear profiles).

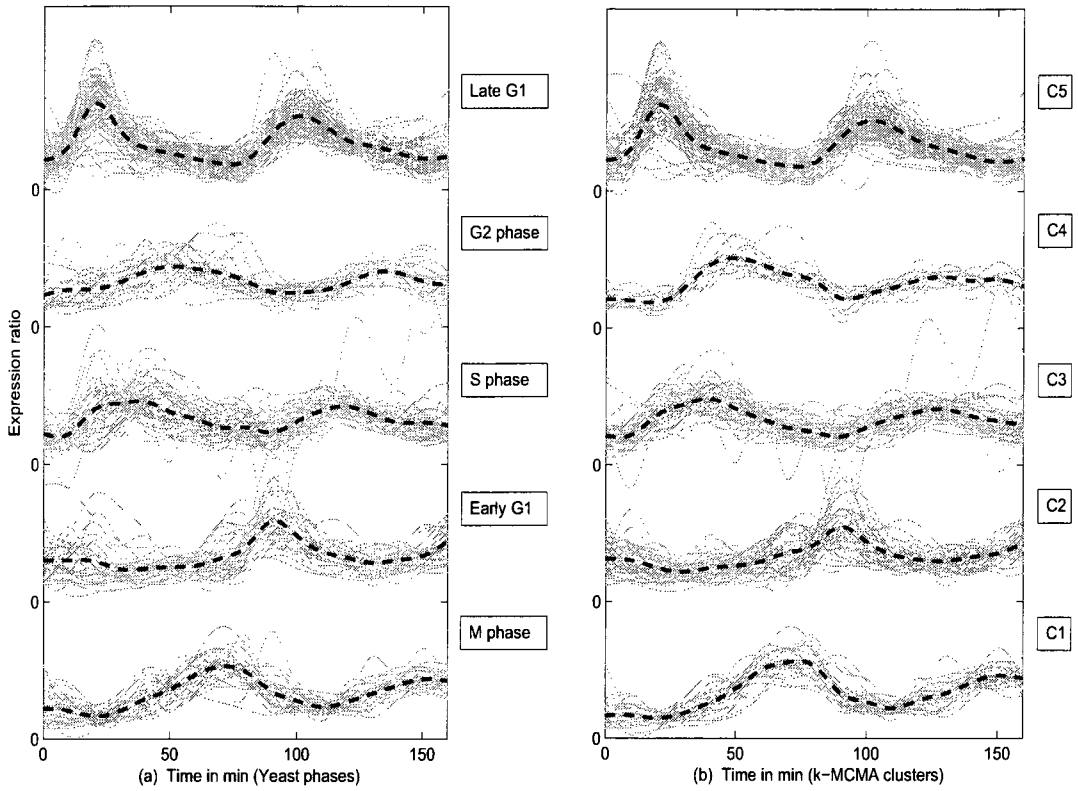

Figure 5.3: (a) *Saccharomyce cerevisiae* phases [22] and (b) k-MCMA clusters using natural cubic spline profiles, with centroids shown.

The clusters found by k-MCMA and EMMA using natural cubic spline profiles are shown in Figs. 5.5(a) and 5.5(c), respectively. The clusters and phases in Fig. 5.5 are vertically shifted by four points up to distinguish them visually. We clearly see that EMMA clusters are similar to exactly one of the luteus phases.

5.4 Comparison with Previous Approaches

We have compared our approaches with the following two previously published approaches: 1) a clustering method that uses piecewise linear profiles which

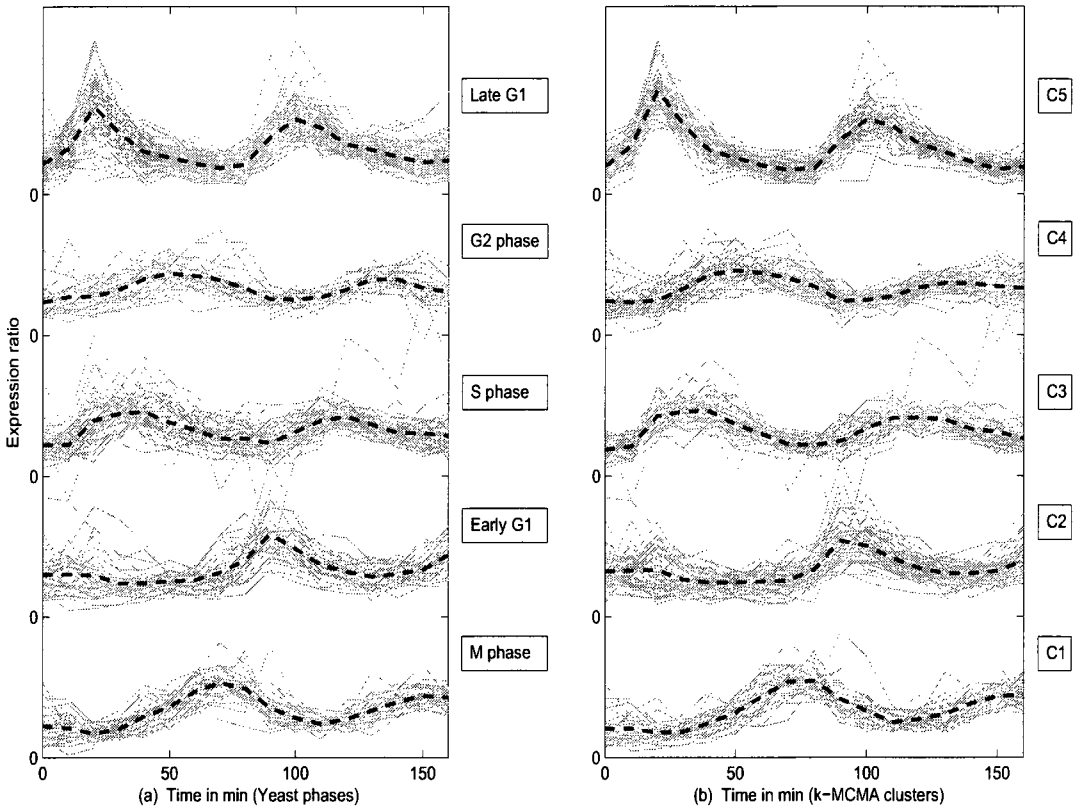

Figure 5.4: (a) *Saccharomyce cerevisiae* phases [22] and (b) k-MCMA clusters using piecewise linear profiles, with centroids shown.

was published in [15], and 2) the Variation-based Coexpression Detection (VCD) algorithm which is described in [36].

We performed an objective measure for comparing the EMMA clusters with the yeast phases. The measurement was computed by taking the average classification accuracy, as the number of genes that EMMA *correctly* assigned to one of the phases. Considering each EMMA cluster as a class, $\hat{\mathcal{C}}_{\hat{\mu}_c}$ $(1 \leq c \leq k = 5)$, we trained a c-Nearest neighbor (c-NN) classifier with clusters to classify the data with a 10-fold cross validation procedure where c is the number of nearest profiles from the centroids. In our scenario, we found that $c = 5$ is the best number of clusters for the data set and we used the

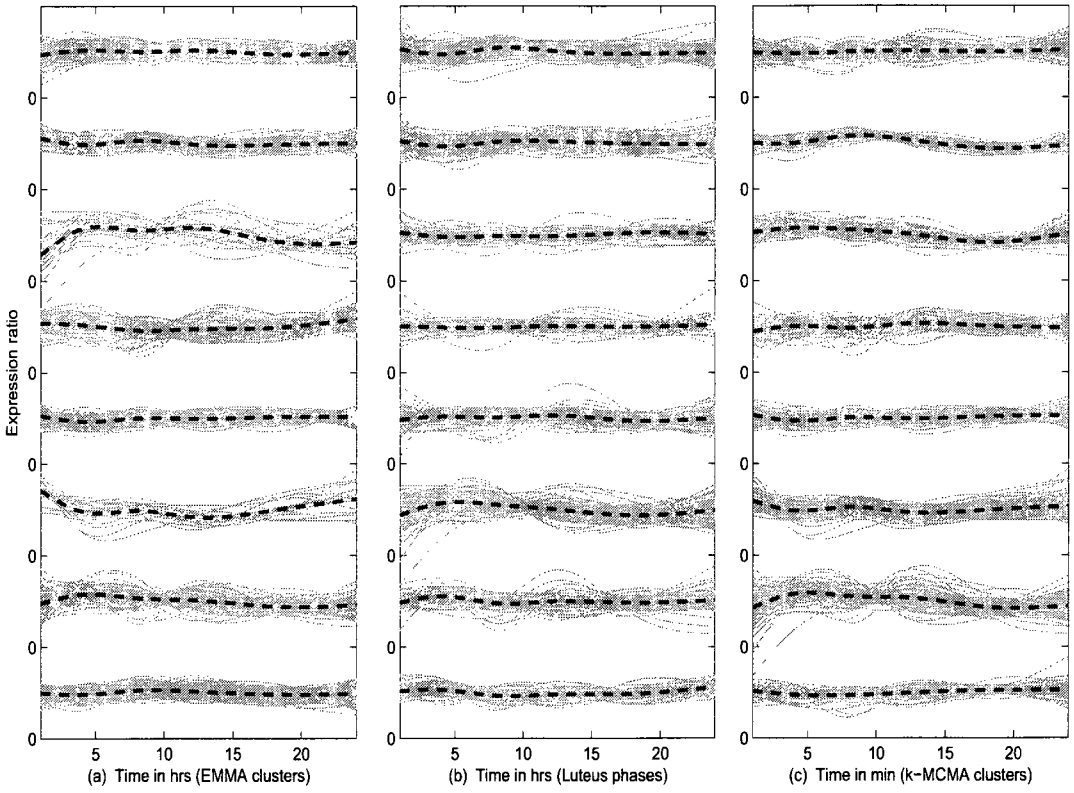

Figure 5.5: (a) EMMA clusters, (b) *Micrococcus luteus* phases [9] and (c) k-MCMA clusters, with centroids shown.

distance function of Eq. 3.16 to measure the distance between the centroids and the nearest profiles. We applied the same procedure for k-MCMA clusters too. This criterion is reasonable, as k-MCMA and EMMA are unsupervised learning approaches that do not know the phases beforehand, and hence the aim is to *"discover"* these phases. In [22], the 5 phases were determined using biological information, including genomic and phenotypic features observed in the yeast cell cycle experiments. EMMA's average classification accuracy is 91.03% whereas k-MCMA is 89.51%.

We also applied the same objective measure as described above for comparing the EMMA clusters with bacterium planktonic clusters and obtained

average classification accuracy of 91.40%. In [34, 15], the correlation coefficient is used as the distance measure between gene profiles while here, we used the distance as defined in Eq. 3.16. Fig. 5.2 shows that EMMA yields the better results than k-MCMA and the methods used in [34].

We also applied the same objective measure as described above for comparing the k-MCMA clusters using piecewise linear profiles (an approach of [15]) with the yeast phases and obtained average classification accuracy of 86.12%. For the bacterium data set, we obtained average classification accuracy of 90.90%. Table 5.13 shows the average classification accuracies of our approaches with the approach of [15].

Profiles	Approaches	*S. cerevisiae*	*P. aerugi-nosa*	*serum*	*M. Luteus*	*E. Coli*
natural cubic spline (NCS)	k-MCMA	89.51%	91.40%	78.47%	85.24%	85.37%
piecewise linear (PL)	k-MCMA	86.12%	90.90%	77.21%	82.73%	81.33%
natural cubic spline (NCS)	EMMA	91.03%	92.71%	85.83%	89.37%	88.36%
piecewise linear (PL)	EMMA	86.43%	89.37%	83.79%	87.76%	86.91%

Table 5.13: Experiment results overview of k-MCMA and EMMA with piecewise linear profile of [15]

From Figs. (5.3 - 5.4) and Table 5.13, we observe that natural cubic spline profiles performed better than piecewise linear profiles. We also see that k-MCMA and EMMA clusters using natural cubic spline profiles on both data sets obtained over 90% classification accuracy, which is very high considering the fact that they are both *unsupervised* learning methods while EMMA yields the better performance results than k-MCMA.

We also performed the same classification comparison for a method presented in [36] which is Variation-based Coexpression Detection (VCD) algorithm. In that approach, gene expressions are translated into gene variation vectors and the cosine values of these vectors are then used to evaluate their similarities over time. We have compared the results of our EMMA approach and VCD approach of [36] on two data sets: *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe* data sets. Results are listed in Table 5.14.

Approaches	*Saccharomyces cerevisiae*	*Schizosaccharomyces pombe*
k-MCMA	89.51%	87.63%
EMMA	91.03%	86.94%
VCD	80.68%	70.46%

Table 5.14: Experiment results overview of EMMA approach and the VCD method of [36]

In Fig. (5.6), cluster no. 5 of EMMA and k-MCMA is similar to the corresponding S phase where as VCD method assigned so many differentially expressed genes and same thing happened for the cluster no. as well. If we carefully see the figure, we can see that EMMA's clusters are better than all other methods. EMMA's clusters are even better than pre-characterized phases, at least visually. In Fig. (5.7), VCD identified three clusters that contain only 2 genes and we can also see visually that there are many genes assigned to the clusters that should not be in those clusters. In this data set, k-MCMA's clusters are better than that of EMMA's.

On *Saccharomyces cerevisiae* data set, both EMMA and VCD found five clusters whereas EMMA clusters obtained over 90% classification accuracy. On *Schizosaccharomyces pombe* data set, we ran EMMA in conjunction with four validity indices. We found there are eight clusters (see 5.12) in this data

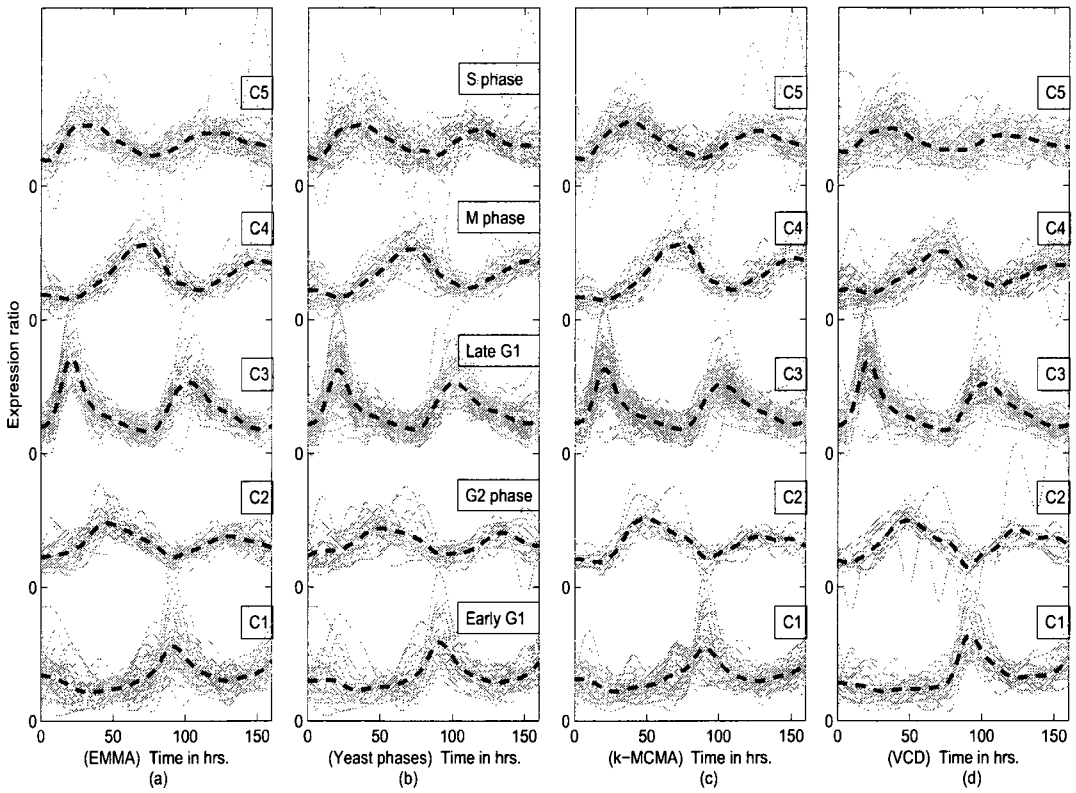

Figure 5.6: (a) EMMA clusters, (b) *Saccharomyce cerevisiae* phases [22], (c) k-MCMA clusters, and (d) VCD clusters, with centroids shown.

set. The EMMA therefore applied with a setting of $k = 8$ and obtained 89.53% classification accuracy. Setting $\lambda = 0.59$ and $z_p = 7$, we applied VCD of [36] on *Schizosaccharomyces pombe* data set as well to find the clusters. VCD identified also eight clusters and obtained 70.46% classification accuracy. VCD identified 33 unique genes which are not belong to any clusters. According to the authors, they obtained 71 clusters in *Schizosaccharomyces pombe* data set when they set the parameters $\lambda = 0.75$ and $z_p = 1.96$. In their method, λ covers the similarity between sets and z_p determines the number of clusters. The eight EMMA and k-MCMA clusters on *Schizosaccharomyces pombe* data set obtained 86.94% and 87.63% classification accuracy, respectively, which

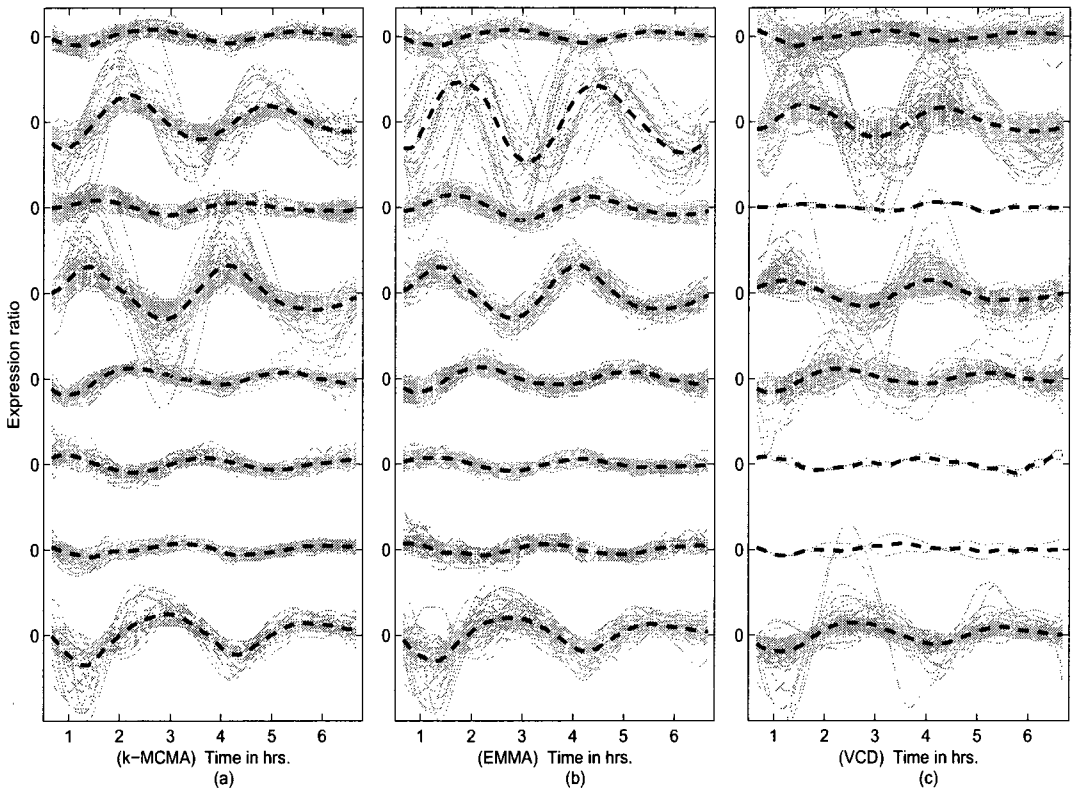

Figure 5.7: (a) k-MCMA clusters, (b) EMMA clusters, and (c) VCD clusters on *Schizosaccharomyces pombe* data set, with centroids shown.

definitely shows that *Schizosaccharomyces pombe* data set contains eight clusters. In fact, EMMA and VCD are both *unsupervised* learning methods while EMMA's performance results better than that of the VCD of [36].

Chapter 6

Conclusion

6.1 Summary of Contributions

In this thesis, we represented the microarray time series profiles using natural cubic spline profiles. We have extended pairwise profile alignment of [25] to use natural cubic spline profile. We also extended the pairwise profile alignment to multiple profile alignment.

We proposed k-MCMA, a method that combines k-means with multiple profile alignment of gene expression profiles to cluster microarray time-series data. The profiles are represented as natural cubic splines functions to compare profiles, where the expression measurements are not necessarily taken at regular time-intervals. Multiple alignment is based on minimizing the sum of integrated squared errors over a time-interval, defined on a set of profiles. A method EMMA is also proposed by combining EM and multiple alignment of gene expression profiles to cluster microarray time-series data. We designed a distance function that suits to the natural cubic spline profiles. Four cluster

validity indices were used in conjunction with the above methods to determine the appropriate number of clusters and also the goodness or the validity of the clusters.

An objective measure for comparing the k-MCMA and EMMA clusters using natural cubic spline profiles with the yeast phases [22] was computed by taking the average classification accuracy, as the number of genes that k-MCMA and EMMA *correctly* assigned to one of the phases. We have used a supervised classification approach (c-nearest neighbor) to consolidate the discriminability of the inferred classes, obtaining accuracy near 90%. This is very high considering that our clustering methods are unsupervised. EMMA performed better than k-MCMA on all data sets. This suggests that EMMA can also be used to correct manual phase assignment errors.

Meanwhile, the results showed that EMMA using natural cubic spline profiles approach for clustering microarray gene expression data, presented in this work, are able to find clusters that are very close to those of biologically characterized phases using natural cubic spline profiles. EMMA using natural cubic spline profiles performed better than the piecewise linear profile method of [15]. EMMA also outperformed the VCD method of [36] especially when the VCD did not assign some of the genes to any clusters.

k-MCMA and EMMA both outperformed some published distance based clustering algorithms (i.e. using piecewise linear profiles of [15] and the VCD method of [36]). Our experiments showed that EMMA algorithm proposed are able to find better clusters than those of biologically characterized phase in [22]. The clustering algorithms proposed in this thesis, can be modified to be used in well-known problems in bioinformatics and computational biology that

are expressed as clustering visualization and supervised pattern recognition for microarray time-series gene expression analysis.

We want to use the benefit of alignment method with the combination of any clustering method in particular k-means and EM. We also analyzed and compared the performance of the proposed methods with other clustering methods and between each other. Initialization is a major issue in k-means and EM methods but we are not interested in improving k-means or EM clustering methods.

6.2 Future Work

In the future, we plan to study other distance-based clustering approaches using our multiple alignment method. Support Vector Clustering (SVC) or Spectral Clustering with multiple alignment approach, different other validity indices (e.g. BAC), phase detection by aligning over a portion of the time series expression can be interesting to investigate. The effect of multiple alignment method on visualizing the clusters can be explored in the field of cluster visualization.

It will be also interesting to study the effectiveness of any clustering methods in dose-response microarray data sets. Cluster validity indices based on multiple alignment can also be investigated. We argue that in real applications data can be very noisy, and the use of cubic spline interpolation could lead to some problems. The use of splines has the advantage of being tractable, however, although we also plan to study interpolation methods that incorporate noise. Though our main focus on clustering, the effect of using different impu-

tation methods rather than natural cubic spline on representing the profiles should also be investigated.

We currently focus on the analysis of gene temporal expression profiles (with the use of spline interpolation and multiple alignment) that can cope with the problem of missing values and non-uniformly sampled data.

Bibliography

[1] A. Ferrer A. Conesa, M.J. Nueda and M. Talon. A method to identify significantly differential expression profiles in time-series microarray experiments. [cited at p. 13, 15]

[2] N. Laird A. Dempster and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977. [cited at p. 34]

[3] C. Bradfield A. Smith1, A. Vollrath and M. Craven. Clustered alignments of gene-expression time series data. *Bioinformatics*. [cited at p. 6, 16]

[4] L. Brehelin. Clustering gene expression series with prior knowledge. *Lecture Notes in Computer Science*, 3692:27–28, 2005. [cited at p. 13, 14]

[5] C.B. Yang C. Lin, H. Chenga and C.N. Yang. Solving satisfiability problems using a novel microarray-based dna computer. *Biosystems*. [cited at p. 5]

[6] K. Cho C. Moller-Levet, F. Klawonn and O. Wolkenhauer. Clustering of unevenly sampled gene expression time-series data. *Fuzzy sets and Systems*, 152(1,16):49–66, 2005. [cited at p. 13, 16]

[7] J. Celis and et al. Gene expression profiling: monitoring transcription and translation products using dna microarrays and proteomics. *Federation of European Biochemical Societies Letters*, 23892:1–15, 2000. [cited at p. 5]

[8] X. Peng et al. Identification of cell cycle-regulated genes in fission yeast. *Molecular Biology of the Cell.* [cited at p. 44]

[9] H. Muller G. Dimopoulos, A. Richman and F. Kafatos. Molecular immune responses of the mosquito anopheles gambiae to bacteria and malaria parasites. *Proc. Natl. Acad. Sci. USA.* [cited at p. 43, 44, 60]

[10] A. Gasch and M. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. [cited at p. 16]

[11] Yury Goltsev and Dmitri Papatsenko. Time warping of evolutionary distant temporal gene expression data based on noise suppression. *BMC Bioinformatics.* [cited at p. 6, 16]

[12] G. Nau J. Ernst and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(supl 1):i159–i168, 2005. [cited at p. 13, 14]

[13] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics.* [cited at p. 39]

[14] S. Kruglyak L. Heyer and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, 9:1106–1115, 1999. [cited at p. 13]

[15] A. Bari L. Rueda and A. Ngom. Clustering time-series gene expression data with unequal time intervals. *Springer Trans. on Computational Systems Biology*, LNBI 5410:100–123, 2008. [cited at p. xiii, 7, 8, 16, 17, 18, 19, 20, 25, 26, 27, 30, 54, 59, 61, 66]

[16] P. Sebastiani M. Ramoni and I. Kohane., editors. *Cluster analysis of gene expression dynamics*, volume 99(14) 9121-9126. Proc. Natl Acad. Sci. USA, 2002. [cited at p. 13, 14]

[17] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transaction Pattern Analysis Machine Intellegence*, 24(12):1650–1654, 2002. [cited at p. 37]

[18] L. Rueda N. Subhani, A. Ngom and C. Burden. Microarray time-series data clustering via multiple alignment of gene expression profiles. *Springer Trans. on Pattern Recognition in Bioinformatics*, LNCS 5780:377–390, 2009. [cited at p. 34, 36, 54]

[19] M. Zhang V. Iyer K. Anders M. Eisen P. Brown D. Botstein P. Spellman, G. Sherlock and B. Futcher. Comprehensive identification of cell cycleregulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998. [cited at p. 6]

[20] J. Mesirov Q. Zhu S. Kitareewan E. Dmitrovsky E. Lander P. Tamayo, D. Slonim and T. Golub, editors. *Interpreting patterns of gene expression with SOMs: Methods and application to hematopoietic differentiation*, volume 96(6):2907-2912. Proc. Natl Acad. Sci. USA, 1999. [cited at p. 13]

[21] N. Weskamp R. Balasubramaniyan, E. Hullermeier and J. Kamper. Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics.* [cited at p. 6, 16]

[22] E. Winzeler L. Steinmetz A. Conway L. Wodicka T. Wolfsberg A. Gareilian D. Lockhart R. Cho, M. Campbell and R. Davis. A genome-wide transactional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, 1998. [cited at p. 39, 42, 50, 51, 55, 56, 58, 59, 60, 63, 66]

[23] S. Rahmann. *Algorithms for Probe Selection and DNA Microarray Design.* PhD thesis, Max Planck Institue for Molecular Genetics, Berlin, 2004. [cited at p. 3]

[24] J. Ramsay and B. Silverman. *Functional Data Analysis.* Springer, New York, USA, 2nd edition., 2005. [cited at p. 15]

[25] A. Bari; L. Rueda. A new profile alignment method for clustering gene expression data. *Springer Trans. on Computational Studies of Intelligence*, LNCS 4013:86–97, 2006. [cited at p. 7, 16, 18, 30, 65]

[26] M. Eisen J. Mulholland D. Botstein P. Brown S. Chu, J. DeRisi and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998. [cited at p. 13]

[27] A. Baccini S. Djean, P. Martin and P. Besse. Clustering time-series gene expression data using smoothing spline derivatives. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007 (70561):705–761., 2007. [cited at p. 13, 15]

[28] L. Li C. Afshari C. Weinberg S. Peddada, E. Lobenhofer and D. Umbach. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, 19(7):834–841, 2003. [cited at p. 13, 14]

[29] M. Campbell R. Cho S. Tavazoie, J. Hughes and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999. [cited at p. 12, 13]

[30] A. Schliep and D.C. Torney, editors. *Group testing with DNA chips: generating desogns and decoding experiments.*, volume 99(14) 9121-9126. Proc. 2nd IEEE Computer Society Bioinformatics Conference, Stanford, CA, USA, 2003. [cited at p. 1]

[31] Naftali Kaminski Tien-ho Lin and Ziv Bar-Joseph. Alignment and classification of time series gene expression in clinical studies. *Bioinformatics.* [cited at p. 6, 16]

[32] D. Ross G. Schuler T. Moore J. Lee J. Trent L. Staudt Jr. J. Hudson V. Iyer, M. Eisen and M. Boguski. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999. [cited at p. 41, 43]

[33] M. Kawanabe V. Roth, J. Laub and J.M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1540–1551, 2003. [cited at p. 34]

[34] R. Waite and et al. Clustering of pseudomonas aeruginosa transcriptomes from planktonic cultures, developing and mature biofilms reveals distinct expression profiles. *Journal: BMC Genomics*, 7(1):162–175, 2006. [cited at p. 42, 61]

[35] R. Xu and D. Wunsch. *Clustering.* Wiley, IEEE Press, 2009. [cited at p. 31, 35]

[36] Zong-Xian Yin and Jung-Hsien Chiang. Novel algorithm for coexpression detection in time-varying microarray data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* [cited at p. xiii, 6, 16, 59, 62, 63, 64, 66]

[37] H. Yue and et al. An evaluation of the performence of cdna micorarrays for detecting changes in global mrna expression. *Federation of European Biochemical Societies Letters*, 29(8):e41, 2001. [cited at p. 5]

[38] T. Jaakkola D. Gifford Z. Bar-Joseph, G. Gerber and I. Simon. Continuous representations of time series gene expresion data. *Journal of Computational Biology*, 10(3-4):341–356, 2003. [cited at p. 15]

Index

Vita Auctoris

NAME:	K M Numanul Hoque Subhani
PLACE OF BIRTH:	Cox'sbazar, Bangladesh
YEAR OF BIRTH:	1979
EDUCATION:	University of Windsor Windsor, Ontario, Canada 2008-2009 M.Sc. 2007-2008 B.C.S. (Honours) 2000-2003 B.C.S. (General)