

COMPUTATIONAL METHODS FOR ANALYSIS AND MODELING
OF TIME-COURSE GENE EXPRESSION DATA

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in the Department of Biomedical Engineering
University of Saskatchewan
Saskatoon, Saskatchewan
Canada

By
Fangxiang Wu

© Copyright Fangxiang Wu, August 2004. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by Professors Chris Zhang and Tony Kusalik who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Biomedical Engineering_____

University of Saskatchewan, Saskatoon, Saskatchewan (S7N 5A9)

ABSTRACT

Genes encode proteins, some of which in turn regulate other genes. Such interactions make up gene regulatory relationships or (dynamic) gene regulatory networks. With advances in the measurement technology for gene expression and in genome sequencing, it has become possible to measure the expression level of thousands of genes simultaneously in a cell at a series of time points over a specific biological process. Such time-course gene expression data may provide a snapshot of most (if not all) of the interesting genes and may lead to a better understanding gene regulatory relationships and networks. However, inferring either gene regulatory relationships or networks puts a high demand on powerful computational methods that are capable of sufficiently mining the large quantities of time-course gene expression data, while reducing the complexity of the data to make them comprehensible. This dissertation presents several computational methods for inferring gene regulatory relationships and gene regulatory networks from time-course gene expression. These methods are the result of the author's doctoral study.

Cluster analysis plays an important role for inferring gene regulatory relationships, for example, uncovering new regulons (sets of co-regulated genes) and their putative *cis*-regulatory elements. Two dynamic model-based clustering methods, namely the Markov

chain model (MCM)-based clustering and the autoregressive model (ARM)-based clustering, are developed for time-course gene expression data. However, gene regulatory relationships based on cluster analysis are static and thus do not describe the dynamic evolution of gene expression over an observation period. The gene regulatory network is believed to be a time-varying system. Consequently, a state-space model for dynamic gene regulatory networks from time-course gene expression data is developed. To account for the complex time-delayed relationships in gene regulatory networks, the state space model is extended to be the one with time delays. Finally, a method based on genetic algorithms is developed to infer the time-delayed relationships in gene regulatory networks. Validations of all these developed methods are based on the experimental data available from well-cited public databases.

Key words: DNA microarray, gene expression, data normalization, gene regulatory relationship, MCM-based clustering, ARM-based clustering, gene regulatory network, state-space model, time delay, genetic algorithm.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my respected supervisors Professor W. J. Zhang and Professor Anthony J. Kusalik, for their invaluable guidance, encouragement, and support during my Ph.D. program as well as the critical reviews and comments on my publications and this dissertation. They are not only insightful advisors, but also good friends. I am happy to meet and to work together with them in past years. In addition, I am very grateful to Professor Zhang for his helps with my living.

I would like to extend my appreciation to my advisory committee members: Professor Glen Watson, Professor Rui Wang, Professor John DeCoteau, and Professor Brian Daku, for their valuable examinations and constructive suggestions to improve the present work. In addition, I am grateful to Professor Watson for his helps beyond my study.

I would also like to acknowledge my colleagues and friends for their friendship, discussion and help to make study life easy and interesting, in particular, ZongXi Zhou, BingChen Wang, JianWu Wang, ChenHong Zhang, PuRen Ouyang, JingXin Li, Yong Zeng, and YingZi Lin.

Finally I would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) for a partial financial support to this research, the College of Graduate Studies and Research at the University of Saskatchewan for funding me through a three-year graduate scholarship award. I would also like to extend my thanks to U. S. Department of Energy and PSB2004 Conference Committee for a travel award to present a part of this work on PSB 2004 conference.

DEDICATION

To

my dear wife Jiping and my lovely son Yichao,

for

their love, encouragement, and devoted support

TABLE OF CONTENTS

PERMISSION TO USE	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
DEDICATION	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiv
1. INTRODUCTION	1
1.1 Background	1
1.2 Overview	4
1.3 Contributions	7
2. GENE EXPRESSION DATA	10
2.1 Measurement Techniques	10
2.1.1 Oligonucleotide and cDNA Microarrays	12
2.1.2 Other Techniques	20
2.2 Gene Expression Datasets.....	25
2.3 Data Pre-processing	30

3. DYNAMIC MODEL-BASED CLUSTERING	34
3.1 Related Work	34
3.1.1 Correlation/Distance-Based Clustering	34
3.1.2 Static Model-Based Clustering	43
3.2 Clustering Validations	45
3.2.1 Internal Indices	45
3.2.2 External Indices	48
3.2.3 A Bootstrapping Method	51
3.3 Markov Chain Model-Based Clustering	52
3.3.1 Gene Expression Dynamics Sequence	53
3.3.2 MCM-Based Clustering Method and EM Algorithms	55
3.3.3 Computational Experiments and Results	59
3.4 Autoregressive Model-Based Clustering	65
3.4.1 Autoregressive Model and Likelihood for a Single Time Series	66
3.4.2 ARM-Based Clustering	69
3.4.3 Computational Experiments and Results	74
3.5 Conclusions	79
4. GENE REGULAORY NETWORKS	81
4.1 Related Work	81
4.1.1 Boolean Network Models.....	82
4.1.2 Differential/Difference Equations Models.....	85
4.2 Evaluations	89

4.3 State-Space Model	92
4.3.1 The Model	94
4.3.2 Model Identification	96
4.3.3 Computational Experiments and Results	100
4.4 State-Space Model with Time Delays	108
4.4.1 The Model	108
4.4.2 Model Identification	110
4.3.3 Computational Experiments and Results	115
4.5 Genetic Algorithm for Inferring Time Delays	122
4.5.1 The Model	123
4.5.2 Genetic Algorithm	125
4.5.3 Computational Experiments and Results	128
4.6 Conclusions	138
5. SUMMARY AND FUTURE WORK	140
5.1 Summary	140
5.2 Future Work	143
5.2.1 Improvement of Dynamic Model-Based Clustering	143
5.2.2 Improvement of the Inference Gene Regulatory Network	145
REFERENCES	149

LIST OF TABLES

2.1 An example of gene expression (ratios of intensities) datasets	26
3.1 Seven hierarchical clustering methods	40
3.2 Contingency table for two partitions of n objects	49
3.3 The parameters in model (3.15) for dataset SYN	59
3.4 The parameters in model (3.15) for dataset BAC	60
4.1 The internal variable expression matrices for datasets CDC15 and BAC	102
4.2 The state transition matrix of internal variables for datasets CDC15 and BAC ...	103
4.3 Comparisons of prediction power between the state-space models with time delays and without time delays for datasets ALP and ELU	119
4.4 Comparisons of prediction power between the state-space models with time delays and without time delays for dataset BAC and CDC28	134

LIST OF FIGURES

2.1 Central Dogma	11
2.2 A schematic diagram for obtaining gene expression data from dual labelling array-based technology	13
2.3 A schematic diagram of SAGE	20
3.1 A dendrogram for hierarchically clustering 7 objects	38
3.2 The procedure for evaluating clustering methods	51
3.3 The EM algorithm for MCM-based clustering	56
3.4 Posterior probability of a gene expression value being in each cluster (state) for dataset SYN	61
3.5 Posterior probability of a gene expression value being in each cluster (state) for dataset BAC	62
3.6 Profile of AARI with respect to the number of clusters for dataset SYN	63
3.7 Profile of AARI with respect to the number of clusters for dataset BAC	64
3.8 Algorithm for ARM-based clustering	72
3.9 Profile of AARI with respect to the number of clusters for dataset SYN	75
3.10 Profile of AARI with respect to the number of clusters for dataset ALP	76
3.11 Profile of AARI with respect to the number of clusters for dataset ELU	77
3.12 Profile of AARI with respect to the number of clusters for dataset ALP	78

4.1 A state-space model for a gene regulatory network	93
4.2 Profiles of BIC with respect of the number of the internal variables for dataset CDC15	100
4.3 Profiles of BIC with respect of the number of the internal variables for dataset BAC	101
4.4 A comparison of the internal variable expression profiles for dataset CDC15	104
4.5 A comparison of the internal variable expression profiles for dataset BAC	105
4.6 The distribution of eigenvalues of gene regulatory system for dataset CDC15 ...	106
4.7 The distribution of eigenvalues of gene regulatory system for dataset BAC	107
4.8 Profiles of BIC with respect to the number of internal variables for dataset ALP	115
4.9 Profiles of BIC with respect to the number of internal variables for dataset ELU	116
4.10 A comparison of the internal variable expression profiles for dataset ALP	117
4.11 A comparison of the internal variable expression profiles for dataset ELU	118
4.12 The distribution of eigenvalues of model with time delays for dataset ALP	120
4.13 The distribution of eigenvalues of model with time delays for dataset ELU	121
4.14 Genetic algorithm for inferring time-delayed relationships	126
4.15 Profiles of BIC with respect to the number of internal variables for dataset BAC	128
4.16 Profiles of BIC with respect to the number of internal variables for dataset CDC28	129
4.17 Plot of prediction error with respect to the number of generations for dataset	

BAC	130
4.18 A comparison of the internal variable expression profiles for dataset BAC	132
4.19 The distribution of eigenvalues of the inferred gene regulatory network with time delays for dataset BAC	133
4.20 Plot of prediction error with respect to the number of generations for dataset CDC28	135
4.21 A comparison of the internal variable expression profiles for dataset CDC28 ...	136
4.22 The distribution of eigenvalues of the inferred gene regulatory network with time delays for dataset CDC28	137

LIST OF ABBREVIATIONS

2D-PAGE	Two-dimensional polyacrylamide gel electrophoresis
AARI	Average adjusted Rand index
AIC	Akaike's information criterion
ALP	A yeast gene expression dataset from the alpha-synchronized experiment of Spellman et al. (1998)
ARI	Adjusted Rand index
ARM	Autoregressive model
BAC	A bacterial gene expression dataset from the experiment of Laub et al. (2000)
BIC	Bayesian information criterion
CDC15	A gene expression data set from the CDC15-synchronized experiment of Spellman et al (1998)
CDC28	A gene expression data set from the CDC28-synchronized experiment of Cho et al (1998)
cDNA	Complementary DNA; complementary single-stranded DNA copy of a message RNA produced by reverse transcription
CF	An operator to calculate the fitness of all individuals in a population and their distribution in Section 4.5.2

Cy3	A fluorescent dye used to label DNA probes for microarray analysis and typically represented in green
Cy5	A fluorescent dye used to label DNA probes and for microarray analysis and typically represented in red
CNS	Central nervous system
DNA	Deoxyribonucleic acid
DP-PCR	Differential display PCR
ELU	A yeast gene expression dataset from the elutriation-synchronized experiment of Spellman et al. (1998)
EM	Expectation maximization algorithm
FA	Factor analysis
GA	Genetic algorithm
GLRT	Generalized likelihood ratio test
HMM	Hidden Markov model
MCM	Markov chain model
MLFA	Maximum likelihood factor analysis
ORF	Open reading frame
PCR	Polymerase chain reaction
PPCA	Probabilistic principal component analysis
RNA	Ribonucleic acid
RT-PCR	Reverse transcription PCR
SAGE	Serial analysis of gene expression

SYN A synthetic gene expression dataset

VEC An operator that transforms a matrix into a vector (Schott, 1997)

Chapter 1

INTRODUCTION

1.1 Background

Advances in genome sequencing and throughput measurement technology (Pease et al., 1994; Schena et al., 1995; Lockhart et al., 1996) for gene expression have enabled investigators to simultaneously measure the expression levels of thousands of genes at a series of time points or under different conditions. Such large-scale data promise informative insights into the regulatory mechanisms of genomes and help enhance the fundamental understanding of life at the molecular level, from gene regulations, to gene functions, or to cellular mechanisms. Such data also have proven useful in genomic disease diagnosis, treatment, and drug design. To realize these promises, analysis of these data requires advanced mathematical tools that are capable of mining large-scale datasets, in particular, capable of inferring gene regulatory relationships and gene regulatory networks.

Gene expression data can typically be divided into two classes: non-time-course and time-course data. In non-time course expression experiments, a snapshot of gene expression levels is taken for cells under varying conditions, for cells from different

categories, or for cells from different tissues; for example, expression levels of tumour cells from different cancer types (Golub et al., 1999). In time-course expression experiments, a temporal cellular process is measured; for example, the response of human fibroblasts to serum (Iyer et al., 1999), response to environmental conditions (Gasch et al., 2000), or the cell division cycle processes (Spellman et al., 1998; Whitfield et al., 2000; Laub et al., 2000). In this dissertation, the term gene expression data refers to time-course gene expression data unless otherwise stated.

Time-course gene expression data can be useful for inferring gene regulatory relationships such as putative functional correlations and gene co-regulated relationships. Cluster analysis has widely been employed and proven useful for this purpose (Eisen et al., 1998). Cluster analysis techniques assign genes with similar expression profiles to the same group (cluster). The intuition behind this is that genes in the same cluster may be co-regulated or functionally similar. The definition of similarity plays an important role in cluster analysis of gene expression. Most cluster analysis techniques employ Euclidean distance or Pearson correlation to measure the similarity among genes and are called distance/correlation-based clustering methods. They include hierarchical clustering (Eisen et al., 1998), k-means clustering (Tavazoie et al., 1999), and self-organizing maps (Toronen et al., 1999). Recently, some static model-based clustering methods (for example, Yeung et al., 2001; Ghosh and Chinnaiyan, 2002; McLachlan et al., 2002) have also been proposed for gene expression data. These methods define genes to be similar if their expression profiles are generated from the same probability distribution. Since arbitrarily permuting time points does not change

the result of the clustering with these distance/correlation- or static model-based clustering methods, the important information about dynamics in time-course gene expression data may be missed, and thus the quality of clustering may not be optimal. Accounting for dynamics of time-course gene expression should improve the quality of clustering for time-course gene expression data.

Time-course gene expression data can also be useful for inferring gene regulatory networks which reflect how genes are regulated in a cell. Inferring gene networks from their expression data is the ultimate goal of time-course gene expression measurements at the large scale. Many models have been proposed for this purpose. For example, Somogyi and Sniegoski (1996) proposed a Boolean network model in which the expression state of a gene is determined by a Boolean function of the states of other genes. Since the Boolean network model views a gene's expression (state) as either completely "on" or "off" (represented by the binary values 1 and 0, respectively), it is a discrete model. Chen et al. (1999) and D'haeseleer et al. (1999) proposed the continuous differential and difference equation models for gene regulatory networks, respectively. In these models, the expression state of a gene is determined by a differential/difference equation. To describe a gene regulatory network with n genes, these aforementioned models need n coupling Boolean equations, or coupling differential equations or coupling difference equations, respectively. Because the number of genes is typically much larger than the number of time points in current gene expression datasets, these models are underdetermined if no further constraints or assumptions are imposed. To make these models identifiable, some assumptions are thus enforced on the structure of

models; for example, the connectivity degree of genes is small (typically 2 or 3). However, not only is the assumption debatable, but the computational complexity of model identification is still expensive. Therefore, it is worthwhile to develop new models to overcome these disadvantages.

1.2 Overview

The dissertation presents research on computational methods for inferring gene regulatory relationships (cluster analysis) and inferring gene regulatory networks from time-course gene expression data. Large portions of this dissertation have been published previously (Wu, 2003; Wu et al., 2004a, b, c, d, e, f). The dissertation consists of five chapters.

Chapter 2 provides background information for this work. Section 2.1 gives a brief overview of the main technologies employed for measuring gene expression levels on the genomic scale. Several real and synthetic time-course gene expression datasets employed in this dissertation are described in Section 2.2. Section 2.3 introduces widely-used pre-processing strategies for analysis of gene expression data, some of which will be employed in the following chapters.

Cluster analysis is a powerful tool for inferring gene regulatory relationships from time-course gene expression data. Many different clustering methods have been proposed to analyze gene expression data, and numerous applications of these methods have been

reported. However, no single one has been accepted as the optimal by the gene expression analysis community. Typical clustering methods ignore information about the dynamics of time-course gene expression, and thus the quality of clustering may be degraded. Section 3.1 reviews this related work, mainly focusing on distance/correlation-based hierarchical clustering, partitional clustering, and static model-based clustering methods. Some internal and external indices for validating cluster analysis are reviewed in Section 3.2. A bootstrapping method and an average adjusted Rand index (AARI) are also proposed to measure the quality of clustering in Section 3.2. In the two subsequent sections, two dynamic model-based clustering methods for time-course gene expression data are presented. Section 3.3 describes a Markov chain model (MCM)-based clustering method in which the Markov chain is employed to account for dynamics of gene expression. To evaluate the proposed method, computational experiments are performed on two gene expression datasets and the results presented. Section 3.4 describes an autoregressive model (ARM)-based clustering method in which an autoregressive equation is used to account for dynamics of gene expression. To investigate the ARM-based clustering method, computational experiments are again performed on several gene expression datasets and the results presented. Section 3.5 concludes the chapter.

Though the study of systems biology has a long history (Bertalanffy, 1968; Wiener, 1948), rooted as much as anywhere in classical physiology (Buchman, 2002), systems biology at the molecular level (e.g., gene regulatory networks) has only recently become feasible with genome sequencing and microarray technology. Many results have been

reported in the literature about gene regulatory networks, but we are far away from a complete understanding of them. Section 4.1 reviews some of these previous reports, mainly focusing on Boolean network models and differential/ difference equation models for gene regulatory networks. Due to limitations in the understanding of real cellular systems, it is difficult to evaluate the models for gene regulatory networks completely by biological experiments. Section 4.2 introduces some indices from the perspective of bioinformatics to evaluate such models. Section 4.3 introduces the state-space model for gene regulatory networks. In this model, genes are viewed as observation variables, whose expression values depend on the current internal state variables and other external inputs, if they exist. The idea behind this view is that genes are regulated by other elements in a cell (Baldi and Hatfield, 2002). Information theory and control system theory are employed to build the state-space model. Some computational experiments on two datasets are performed to evaluate the proposed models. Since the real microarray data example reveals a considerable number of time-delayed interactions (Alter et al., 2000, 2002; Rosenfeld and Alon, 2003; Yildirim and Mackey, 2003) suggesting that time delays are common in gene regulation, a state-space model with time delays for gene regulatory networks is further proposed in Section 4.4. Computational experiments are also performed to evaluate models proposed in this section. In the state-space model with time delay, the identification of time-delayed regulatory relationships are very important. Section 4.5 presents a genetic algorithm (GA) to infer the time-delayed relationships in a gene regulatory network from gene expression data. The results of computational experiments on one dataset are presented. Section 4.6 concludes the chapter.

Chapter 5 summarizes the dissertation and discusses several possible directions for future work.

1.3 Contributions

This dissertation focuses on developing computational methods for inferring gene regulatory relationships and inferring gene regulatory networks from time-course gene expression data. The main contributions are:

- A Markov chain model (MCM)-based clustering method is developed for time-course gene expression data, in which the Markov chain model is employed to account for the dynamics of time-course gene expression. Computational experiments on gene expression datasets are performed to validate the method. The results show that the quality of clustering from the method is improved as compared to the k-means clustering method.
- An autoregressive model (ARM)-based clustering method is developed for time-course gene expression data, in which an autoregressive equation is employed to model the dynamics of time-course gene expression. Computational experiments on gene expression datasets are performed to validate the method. The results show that the quality of clustering from the

method is improved as compared to both the k-means clustering method and the MCM-based clustering method.

- A state-space model is proposed for gene regulatory networks. Unlike Boolean network and differential/difference equation models, the state-space model views genes as the observation variables whose expression values depend on the current internal state variables and other external inputs, if they exist. Maximum likelihood factor analysis (MLFA) and the Bayesian information criterion (BIC) are employed to estimate the number of internal variables and their expression profiles from time-course gene expression data. Computational experiments are performed on two gene expression datasets. The results show that not only may model parameters be unambiguously identified from current time-course gene expression datasets with a modest computational cost, but also the inferred gene regulatory networks have some features of real gene regulatory networks.
- A state-space model with time delays is developed for gene regulatory networks which is an extension of the state-space model. Probabilistic principal component analysis (PPCA) and the BIC are employed to estimate the number of internal variables and their expression profiles from time-course gene expression data. Computational experiments are performed on two gene expression datasets. The results show that the inferred gene regulatory networks have better predication accuracy and demonstrate more features of real gene

regulatory networks as compared to networks using the model without time delays.

- A genetic algorithm is proposed for inferring time delays in gene regulatory networks. Computational experiments on two gene expression datasets are performed to investigate the proposed algorithm. The results show that the algorithm may effectively infer time-delayed relationships among the internal variables in gene regulatory networks.

Chapter 2

GENE EXPRESSION DATA

2.1 Measurement Techniques

It is well known that deoxyribonucleic acids (DNAs) in a living cell encode all genetic information of a living organism. According to the central dogma of genetics (Figure 2.1), by the transcriptional process DNAs are synthesised into a class of cellular ribonucleic acids (RNAs) called messenger ribonucleic acids (mRNAs) and pass on the genetic information to mRNAs. Further, mRNAs carry genetic information from the nucleus to the cytoplasmic protein synthesis machinery (ribosomes), where they are translated into proteins. It has long been recognized that mRNA plays a pivotal role in determining the type and quantity of proteins produced by cells. Indeed, the differences in protein content of different types in cells are a reflection of differences in the mRNA species expressed and of their levels of expression (abundance) during cellular development and maintenance. Once such differences in mRNA populations among types of tissues/cells are appreciated, it becomes important to quantify these differences. Therefore methods for accurate quantification of specific mRNA species in biological samples need to be developed.

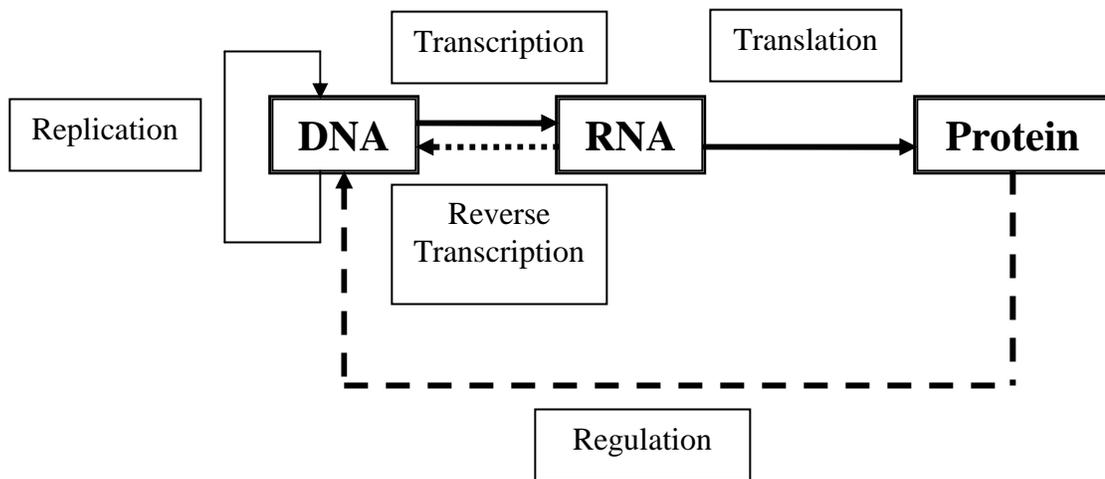


Figure 2.1 Central Dogma (Clark and Russell, 2001, Alberts et al, 1998)

Original hybridization-based assays (e.g., Northern blot) were facilitated by the unique selectivity of nucleic acid base pairing. Applications of such gene-by-gene analysis methods established that some transcripts are abundant in certain tissues whilst absent in others, and that some genes are expressed at relatively consistent levels in many/all tissues. Other techniques for mRNA quantification have since been developed to complement Northern blot such as quantitative reverse transcription polymerase chain reaction (RT-PCR) (Wen et al., 1998). However, these techniques are also limited to relatively few genes per assay.

It should provide an opportunity to gain insights into the total program of genomic activity underlying a biological change (as may be part of growth, a response to stimulus, etc) that several hundreds and thousands of genes as assayed in parallel rather

than one at a time. Technological advances based around polymerase chain reaction (PCR), large-scale cDNA library sequencing, and *de novo* nucleic acid synthesis have contributed to the development of a wide range of techniques for mRNA quantification on a (near) genomic scale. These techniques include differential display PCR (DP-PCR), serial analysis of gene expression (SAGE) and DNA array hybridization. They all have significant benefits over the Northern blot in terms of sensitivity and genes assayed per amount of RNA. This section introduces these techniques and compares them.

2.1.1 Oligonucleotide and cDNA Microarrays

The most commonly used methods for assessing mRNA levels are based on hybridization of a labelled population of nucleic acids (representing an mRNA sample) to an array of individual cDNA sequences or oligonucleotides, printed as spots onto a solid support. After hybridization the intensity of the label associated with each spot represents the expression level for that particular sequence. This intensity is compared to the intensity of the equivalent spot on an identical gridded array hybridized with the material prepared using mRNA isolated from a different source. This provides a measure of differential mRNA expression specific for that spot's DNA sequence (gene). By comparing all equivalent spot intensities between two grids, the changes in expression of many thousands of genes can be monitored simultaneously. The use of distinguishable labels for the two mRNA populations under study permits cohybridization onto one gridded array; a measure of differential hybridization is then obtained by comparing intensities of the two labels at each spot (Figure 2.2). The range

of sequences present on an array limits these methods. Thus if a particular gene sequence is not represented on the array, then obviously it cannot be assayed. For example, higher organisms contain about 100,000 different genes, but only 15% of these genes (i.e. ~15,000) are expressed at one time in any individual cell (Liang and Pardee, 1992). Therefore, the selection of which DNA elements are present on an array is of crucial importance for any study.

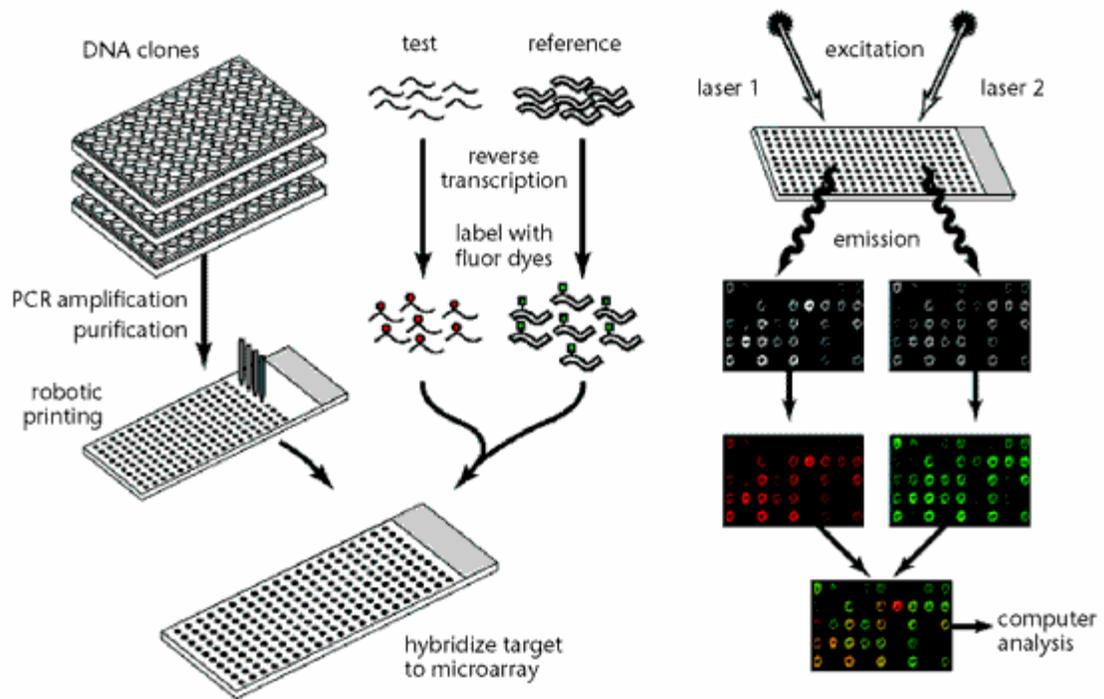


Figure 2.2 A schematic diagram for obtaining gene expression data from the dual labelling array-based technology (Duggan et al., 1999)

Using DNA arrays to measure mRNA abundance is directly analogous to performing multiple reverse Northern blots simultaneously. Instead of RNA being immobilized on a

solid support and being hybridized with gene-specific labelled DNA (Northern blot), the gene-specific DNA elements are attached to a solid support and hybridized to labelled material derived from RNA. The power of array technology is derived from the fact that many thousands of DNA elements (genes), printed as spots, can be assayed in parallel. Typically, thousands of DNA elements are robotically printed onto a nylon membrane or glass slide at high density. Currently, higher spotting densities are achievable on glass, the benefit of which is an increase in the number of genes assayed per amount of input RNA without any compromise in terms of assay sensitivity. DNA arrays can be classified into two types depending on the chemical nature of DNA elements which are used to construct the array: either denatured PCR products (derived from a cDNA library) or oligonucleotides. Therefore, array construction requires access to either cDNA clones or sequence information (for design of representative oligonucleotides). Thus, array design is restricted by our knowledge regarding the genes which make up an organism's genome (hybridizations are performed in a species-specific manner).

When comparing mRNA samples using only one label (single labelling approach), the procedure for identification of differentially expressed genes is relatively simple; labelled RNA (from two or more samples) is applied to identical DNA arrays, and the intensities of spots are compared between the resulting images (after normalization to allow for specific activity differences). Another common technique, the dual labelling approach, uses two distinguishable labels (e.g. Cy3 and Cy5 (Eisen and Brown, 1999)) for two different samples and applying the two differently labelled samples to one glass slide array. The two fluorescent labels, having distinct characteristic emission

wavelengths, can be discriminated from each other, permitting detection of differential hybridization using a single array. The ratio of fluorescent intensities at each spot coordinate gives the result of a sequence-specific competitive hybridization. The principal advantage of using a single array technique over the dual labelling approach is the elimination of artifacts resulting from subtle quality differences between individual gridded arrays. However, a disadvantage of using single-array dual-hybridization is that the resulting ratios are only valid within a particular dataset. For example, for a time course the time-point samples would be labelled with Cy5, and each is hybridized to a separate array alongside the same Cy3-labelled zero-hour mRNA reference samples. As different studies will use different reference samples, the ability to relate results from different studies is compromised, although not impossible (Blakemore et al, 2001). However, this problem can be addressed in principle by proper data processing methods (see Section 2.3).

cDNA arrays: They consist of PCR products, derived from cDNA libraries, robotically spotted onto a solid support. cDNA arrays can be constructed from undefined cDNA libraries (in which the knowledge of cDNA sequences is unknown) or from defined libraries (in which DNA sequences are previously characterized). There are two major advantages of using defined cDNA libraries: (1) the ability to check of the presence of genes of interest for a particular study (i.e., positive/negative controls); and (2) the ability to expedite gene expression data by the mapping of them to gene annotations. The alternative of using undefined cDNA sources requires identification of differentially expressed genes by DNA sequencing of their corresponding clones. This method is

time-consuming and can be inefficient as the same gene may be repeatedly identified (especially when using a standard redundant cDNA library as array source material). Establishment of a facility for cDNA array production, hybridization, and for gene expression data analysis requires considerable bioinformatics support. Accurate and reliable computer-based systems are essential at all stages of the process. This includes choice of cDNA sequences for an array, sample tracking to ensure the correct cDNA is located to the correct array spot coordinate, measurement of hybridization signal intensities of all spot coordinate, retention of hybridization results in a database (linking the hybridization data to gene annotations), and subsequent data-mining.

cDNA arrays can be used to perform transcript profiling in any species (provided that an appropriate cDNA library is accessible), enabling the simultaneous monitoring of tens of thousands of gene sequences, facilitating thorough data analysis and databasing of results. The utility of cDNA arrays for generating novel biological information has been demonstrated by the increasing number of publications in this field; examples include cancer gene expression profile studies (DeRisi et al., 1996; Golub et al., 1999; Moch et al., 1999; Perou et al., 1999), identification of a key insulin-resistance gene (Aitman et al., 1999), detection of neural gene expression changes during the circadian cycle (Patten et al., 1998), comprehensive identification of cell cycle-regulated genes of the Yeast *Saccharomyces cerevisiae* (Spellman et al., 1998), the identification of altered lymphocytic gene expression in asthma (Syed et al., 1999), and identification of genes periodically expressed in the human cell cycle and their expression in tumors (Whitfield, et al., 2002). A demonstration of the novel biological insight with this technology is the

study on the response of quiescent cultured human fibroblasts to serum (a well characterized model of cell cycle control) using an 8613 gene array (Iyer et al., 1999). However, cDNA arrays can only provide truly genome wide assays for those species whose whole genome sequences are known. The costs of production of arrays and establishment of bioinformatics tend to be prohibitive for the majority of laboratories, resulting in limited access to the technology.

Oligonucleotide arrays: In terms of utility and general performance, oligonucleotide arrays are similar to cDNA arrays. Both consist of DNA elements arrayed at high density on a solid matrix (e.g., a glass slide), which are used for hybridization-based gene expression profiling. Instead of arraying PCR products from cDNA clones, oligonucleotide arrays are made up of synthetic gene-specific oligo-deoxynucleotides.

The basic principal of “oligo” arrays is that short oligodeoxynucleotides (usually 20-25-mers) can contain sufficient sequence complexity to selectively hybridize a single transcript. In practice, for one gene several different component oligonucleotide sequences are usually placed on an array. Obviously, the construction of an oligo array requires prior knowledge of the expressed sequences, limiting their usefulness to those species whose expressed genomes have been extensively characterized. However, the use of oligos means that there is no need to retain and to carefully take care of physical collections of cDNA clones and PCR products, which simplifies the logistics of accurate array assembly. Indeed, the use of “on-chip” oligo synthesis (Baldi and Hatfield, 2002) can minimize the risk of array error. Methods are available that facilitate oligo design in

order to provide unique sequence-specificity to a single transcript (Wodicka et al., 1997). These methods offer the optimal choice of sequences based on available genome sequence information to reduce the possibility of artifactual results caused by cross-hybridization. Effects of cross-hybridization cannot easily be ruled out when using cDNA arrays.

Once oligonucleotide sequences have been chosen to represent a required set of genes there are two basic methods for array production: robotic spotting of pre-synthesized oligos similar to that for cDNA arrays; or direct photolithographic DNA synthesis on the surface of the array as developed by a US biotechnology company, Affymetrix (<http://www.affymetrix.com>; Lockhart et al., 1996). The Affymetrix method can produce arrays of higher spot density (10^6 elements/cm²; Mcgall et al., 1996) than robotic spotting of oligonucleotides (3×10^5 elements/cm²; Yershov et al., 1996). Currently, designing and constructing arrays by the direct photolithographic DNA synthesis method is considerably more expensive, limiting its availability to potential users, and involves generating a preset array-specific masks for the photolithography process, causing the technology to be relatively inflexible. However, Singh-Gasson et al. (1999) describes a method that may reduce the cost of on-slide oligonucleotide synthesis as it does not require production of array specific masks.

One feature of oligonucleotide arrays which differentiates them from cDNA arrays is that it is possible to include imperfectly matched oligos for the represented gene set, as well as perfectly matched ones. This practice is employed on Affymetrix arrays, which

include imperfect oligos different a single mismatch nucleotide from perfect ones. In this way hybridization specificity of each oligo is reported and accommodated, providing an increased level of quantitative accuracy and possibility of noise estimation for each gene assayed. It should also be noted that oligonucleotides are single-stranded templates ready for hybridization, whereas arrayed PCR products must first be denatured to supply single-stranded hybridization templates. It is important that the denaturation procedure is consistent to optimize reproducibility between cDNA arrays — this is not a consideration for oligonucleotide arrays. For converting mRNA or total RNA into labeled material oligonucleotide arrays use the same methods as cDNA arrays, among which the most common one is fluorescent labeling engaging some form of linear amplification to supply sufficient material of high specific activity. Data analysis and databasing issues are similar to those for cDNA arrays.

Oligo arrays offer a technology which has similar attributes to cDNA arrays but which can achieve higher gene-specific hybridization accuracy than cDNA arrays, albeit at a higher cost. The requirement of prior gene sequence knowledge to design oligo arrays will become a lesser consideration as genome sequencing projects mature. Examples of oligo array application include simultaneous monitoring of expression of all yeast genes (Wodicka et al., 1997; Holstege et al., 1998; Cho et al., 1998), identification of redundancy in mouse receptor tyrosine kinase-activated signaling pathways (Famborough et al., 1999), and analysis of effect of calorific restriction on mouse skeletal; muscle aging (Lee et al., 1999).

2.1.2 Other Techniques

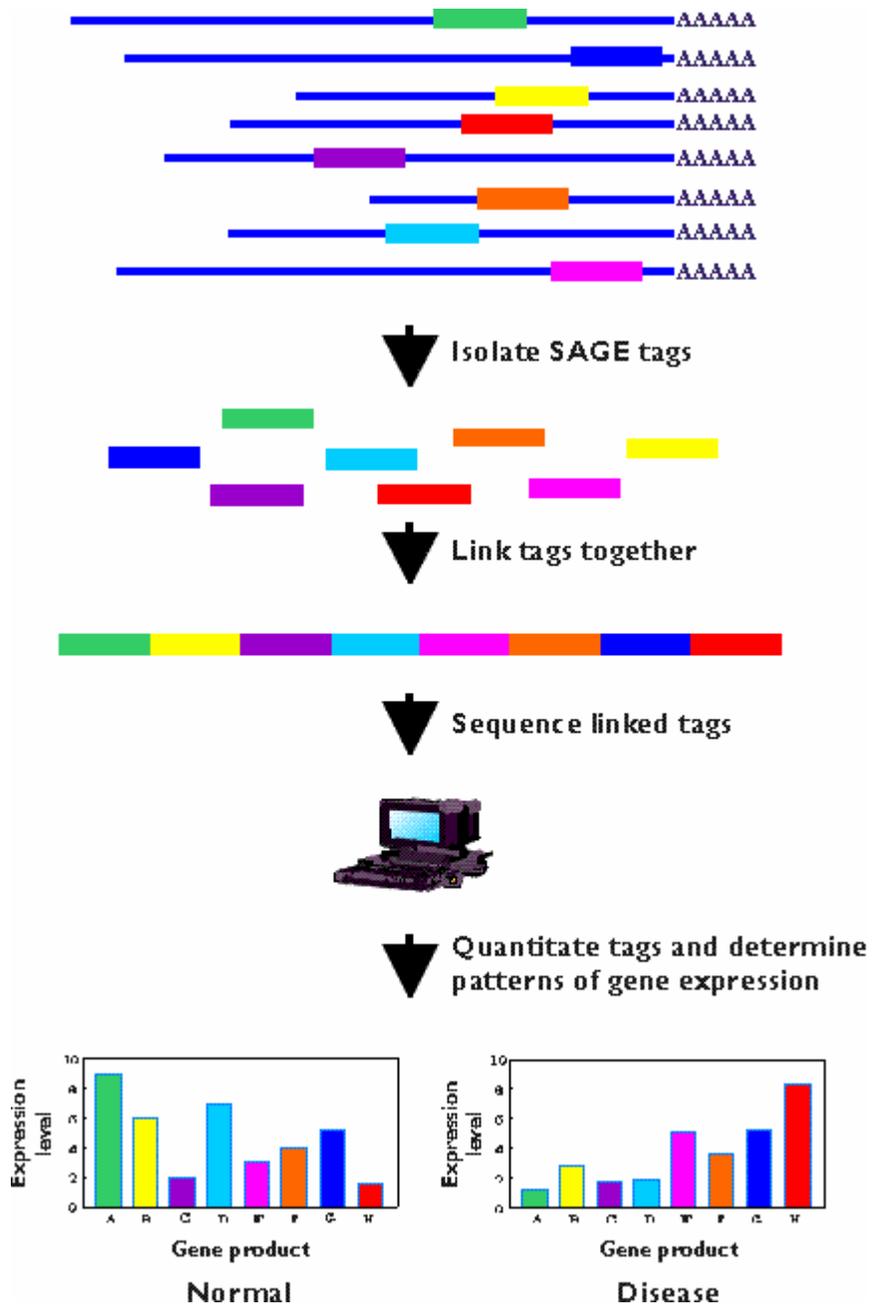


Figure 2.3 A schematic diagram of SAGE (Velculescu et al., 1995)

In addition to DNA array-based methods, there are several other methods for mRNA quantitation (Pennington and Dunn, 2001). This section gives a brief introduction to two of these methods: serial analysis of gene expression (SAGE) and differential display PCR (DD-PCR).

Serial analysis of gene expression – SAGE: Serial analysis of gene expression (SAGE) is a DNA sequence-based technology for quantifying mRNA abundance first published in 1995 from the laboratory of Bert Vogelstein (Velculescu et al., 1995). SAGE sequences cDNA inserts of cloned cDNA libraries—the necessity to sequence tens of thousands of cloned cDNA inserts to provide quantitative accuracy and may overcome some shortcomings of gene expression analysis (e.g., cross-hybridization). The fundamentals of SAGE involve isolation of short, unique sequence tags (9-14 bases) representing a defined region of each individual transcript, followed by their concatenation, cloning of the tag concatenates, sequencing of the cloned concatenates, and then quantitating of the tags (Figure 2.3). The frequency of representation of a particular sequence tag within the total number of tags is then a measure of the frequency of its mRNA in the original population. Theoretically, a tag length of 9-14 bases provides sufficient sequence information to unequivocally identify specific mRNA transcripts (Velculescu et al., 1995). For example, if one assumes a random distribution then all possible permutations of 10 bases (4^{10}) yield 1,048,576 possible combinations, which is about thirty times greater than the estimated number of genes constituting the human genome (Lander et al., 2001). Therefore, by reducing the DNA sequence to a minimum informative length there is a gain in efficiency over the cDNA

library sequencing method: for each “sequence tag concatenate” clone sequenced, 30-50 fold more gene information is acquired (Bertelsen and Veculescu, 1998).

Just as for sequencing cDNA library clones, SAGE data is digital and its range is theoretically limitless. Methods have been developed for assessing the significance of differences in sequence tag abundance derived from two biological samples based on simulations (Zhang et al., 1997) and on statistical methods (Audic and Claverie, 1997). Using independent methods, it has been demonstrated (Madden et al., 1997; Veculescu et al., 1995, 1997) that SAGE sequence tag frequencies are an accurate measure of transcript abundance for mRNAs. For example, expression levels ranging from 0.3 to over 200 transcripts per cell (containing totally 60,633 transcripts) can comfortably be analyzed by SAGE (Veculescu et al., 1997). SAGE applications have included the identification of p53-induced genes prior to apoptosis in a human colorectal cancer cell line (Polyak et al., 1997), the analysis of gene expression profiles of normal versus cancer cells (Zhang et al., 1997), and annotation of the human genome (Saha et al., 2002). Further examples of SAGE applications can be found at SAGE (<http://www.sagenet.org/findings/index.html>) and NCBI (<http://www.ncbi.nlm.nih.gov/SAGE/>).

There are two major drawbacks to SAGE technology. Firstly, as a short sequence tag distinguishes each transcript, high-quality sequence data is essential for its accurate identification. Secondly, the ability to successfully identify the originating transcript for a tag is directly related to the number of sequences (in particular, 3' end sequences) deposited in databases for each species. In conclusion, SAGE is broadly applicable to

any biology system and, in conjunction with automated DNA sequencing and sufficient bioinformatics supports, it is an efficient and accurate method for quantifying mRNA abundance.

Differential display PCR (DD-PCR): Differential display PCR is a method for identifying cDNA fragments that are differentially expressed between two biological samples (Liang and Pardee, 1992). It is based on generating cDNA fragments from mRNA using two oligonucleotide primers, one being complementary to the polyA tail of transcripts (e.g., oligo-d(T)₁₁VN), and the other a short random nucleotide sequence (e.g., 10-mer). DD-PCR has the potential to identify all transcripts present in a biological sample when sufficient primer combinations are applied. After cDNA synthesis, the fragments are labeled (radiolabel or fluorescent) during PCR amplification. The products are then separated by electrophoresis on a sequencing gel, and the pattern of amplified cDNA fragments is visualized. The intensity of a labeled band reflects the relative abundance of its mRNA transcript within the original mRNA population. Major differences in the cDNA band patterns generated from two biological samples, when using the same set of primers, indicate the presence of differentially expressed transcripts. Cloning and sequencing of the eluted cDNA bands enables the identity of the genes from which these cDNAs originate to be defined.

DD-PCR has been applied in many diverse areas: for example, identifying novel drug targets (Shiue, 1997; Wang and Feuerstein, 1997), uncovering differentially regulated genes in rheumatoid arthritis (White and Petkovich, 1998), and assessing effects of

environmental stimuli on bacterial gene expression (Fislage, 1998). Indeed, DD-PCR is currently the most widely published technique for the identification of differentially expressed genes. However, this probably reflects the fact that DD-PCR does not require either expensive specialized equipment or sophisticated bioinformatics analysis tools, facilitating its introduction into many laboratories (rather than any intrinsic superiority over other differential expression techniques). A significant advantage of DD-PCR is the relatively small quantity of input RNA required. Originally, the order of several hundred nanograms of RNA was required (Liang and Pardee, 1992). Recently an adaptation of the technique has claimed to need only the RNA derived from a single cell (Renner et al., 1998). Therefore, RNAs from a biological sample should not preclude application of DD-PCR. However, DD-PCR also has disadvantages. As mentioned earlier by Liang et al. (1994), the technique does suffer from false positives. Also, a band identified as differentially expressed may not always be a single molecular species, as more than one mRNA transcript could generate fragments of similar size commigrating on the electrophoresis gel. This can result in difficulties identifying the “gene” which give rise to the observed “band”. Nevertheless, more improvements were addressed such difficulties later (Prasher and Weissman, 1996)

In summary, DD-PCR can be successfully used to identify differentially expressed genes in any tissue from any species from which high-quality RNA can be isolated (e.g., Shiue, 1997; White and Petkovich, 1998; Renner et al., 1998). The simplicity of this technique and its relative low cost has led to its widespread use. However, results are

often more qualitative rather than quantitative and tend to be more error prone than data generated from other techniques such as array-based data and SAGE-based data.

It should be noted that all current techniques for mRNA quantification provide relative rather than absolute mRNA steady-state level information; however, this is sufficient for detection of changes in levels of mRNA. All methods require significant experimental input and subsequent work to validate findings. For this reason there are currently no published examples where results from two techniques, for example cDNA array and SAGE, have been comprehensively compared. Therefore, it is difficult to provide absolute comparisons of utility of each technique. However, without doubt the utility of array-based hybridization, SAGE, and DD-PCR for mRNA quantitation has been demonstrated by confirmation of findings by independent methods (RT-PCT or Northern blot).

2.2 Gene Expression Datasets

In order to investigate the methods for either clustering gene expression data or modelling gene regulatory networks presented in the dissertation, several gene expression datasets are employed. There are five real-life datasets and one synthetic dataset. This section gives a brief description of all these datasets. Numerical gene expression data is usually collected in a data matrix, where each row is the expression values for a single gene on all microarrays (conditions, time points), and each column is the expression values for all genes on a single microarray (Table 2.1).

Table 2.1 An example of gene expression datasets from the bacterial gene expression experiments (Laub et al., 2000), where each row except for the first one is the expression values for a single gene at 11 time points over 150 minutes, and each column except for the first one is the expression values for all genes on a single microarray at one time points. The first row shows that one mRNA sample is taken each 15 minutes during bacterial development while the first column lists a part of the transcript (mRNA) names expressed by ORF number in the original dataset.

ORF	0m	15m	30m	45m	60m	75m	90m	105m	120m	135m	150m
ORF06244	0.37	0.97	1.93	1.2	1.38	1.22	0.66	0.45	0.5	0.98	1.39
ORF03152	0.19	0.13	0.09	0.07	0.07	0.41	0.93	1.45	1.17	0.99	0.46
ORF03156	0.22	0.12	0.08	0.07	0.07	0.38	0.92	1.27	1.34	0.85	0.5
ORF03161	0.25	0.18	0.15	0.06	0.1	0.42	0.97	1.57	1.43	1.14	0.67
ORF00509	0.35	0.18	0.16	0.16	0.21	0.99	2.3	1.58	1.65	1.12	0.62
ORF02752	0.14	0.14	0.24	0.14	0.25	0.85	1.81	1.7	1.2	0.68	0.4
ORF00082	0.83	0.81	1.2	0.82	1.05	0.99	0.56	1.2	1.01	0.6	1.2
ORF00076	0.27	0.12	0.09	0.09	0.14	0.89	1.46	1.92	1.45	1.08	0.66
ORF00072	0.76	0.61	0.88	0.76	0.93	1.22	1.04	0.99	0.79	1.11	0.83
ORF02312	3.01	1.15	0.84	0.66	0.54	0.65	0.77	0.69	0.78	0.84	1.01
ORF02316	1.18	1	0.81	0.97	0.72	0.94	0.95	0.87	1.03	1.33	1.13
ORF02318	0.81	1.08	0.97	0.88	0.8	0.95	0.89	0.73	0.89	0.93	1.11
ORF04267	0.86	0.63	0.55	0.42	0.41	0.67	0.42	0.36	0.58	0.52	0.55
ORF04260	1.23	0.87	0.9	1.1	0.79	1.19	1.21	0.98	0.86	1.17	0.96
ORF01334	0.91	0.75	0.57	0.57	0.63	0.85	0.66	0.6	0.64	0.85	0.79
ORF00097	1.04	0.98	0.72	0.9	0.58	0.82	0.76	0.7	0.68	0.71	0.73
ORF02592	0.63	0.51	0.67	0.63	0.81	0.75	0.5	0.46	0.54	0.46	0.52
ORF01741	1.23	2.52	2.44	2.27	1.15	1.13	0.63	0.45	0.58	0.98	1.32
ORF01745	0.67	0.8	0.85	0.62	1.25	1.29	1.29	0.85	0.62	0.94	0.98
ORF01751	1.25	1.27	1.25	1.26	1.26	1.44	1.42	0.88	1.05	1.3	1.12
ORF01754	2.46	1.41	1.31	1.53	1.31	1.61	1.68	0.88	1.04	1.32	1.14
ORF03868	1.62	1.6	1.34	1.62	0.95	1.62	1.62	1.08	1.27	1.38	1.62
ORF05200	1.91	1.79	1.77	1.36	1.12	1.6	1.29	1.04	1.03	1.44	1.28
ORF05206	7.37	36.3	21.7	36.3	21.2	36.3	18.2	7	3.42	11.9	36.3
ORF05213	0.95	1.16	1.14	1.24	0.73	0.88	1.11	1.13	1.33	1.33	1.12
ORF01317	2.18	1.27	1.6	1.64	1.94	1.74	1.12	0.99	0.97	1.87	1.3
ORF01315	1.56	1.76	1.41	0.63	0.99	1.09	0.84	0.63	0.84	1.03	0.99
ORF01312	1.06	2.13	1.67	1.71	1.31	1.84	1.57	1.02	0.85	1.52	1.41

Dataset CDC15: The dataset CDC15 is from the CDC15-synchronized experiment for yeast gene expression (Spellman et al., 1998) and consists of the expression data of 799 cell-cycle regulated genes for the first 12 equally-spaced time points representing the first two cycles. The dataset is available at <http://cellcycle-www.stanford.edu>, and missing data were imputed by the mean of gene expression values on the same microarray. This dataset is used to investigate the state-space model for gene regulatory networks in Section 4.3.

Dataset ALP: The dataset ALP is from the alpha-factor synchronized experiment for yeast gene expression (Spellman et al., 1998) and consists of expression levels of 701 cell-cycle regulated genes at 18 equally-spaced time points with no missing data. The dataset is available at <http://cellcycle-www.stanford.edu>. This dataset is used to investigate the MCM- based clustering methods in Section 3.3 and to investigate the state-space model with time delays for gene regulatory networks in Section 4.4.

Dataset ELU: The dataset ELU comes from the elutriation-synchronized experiment for yeast gene expression (Spellman et al., 1998) and consists of expression levels of 789 cell-cycle regulated genes at 14 equally-spaced time points with no missing data. The dataset is available from the website <http://cellcycle-www.stanford.edu>. This dataset is used to investigate the ARM-based clustering methods in Section 3.4 and to investigate the state-space model with time delays for gene regulatory networks in Section 4.4.

Dataset BAC: The dataset BAC comes from the experiment for bacterium gene expression (Laub et al., 2000) and consists of expression levels of 1590 cell-cycle regulated genes at 11 equally-spaced time points with no missing data. The dataset is available from the website <http://caulobacter.stanford.edu/CellCycle>. This dataset is used to investigate both the MCM- and the ARM-based clustering methods in Sections 3.3 and 3.4, respectively, and also to investigate the state-space model for gene regulatory networks in Section 4.4 and the genetic algorithm for inferring time delays in gene regulatory networks in Section 4.5, respectively.

Dataset CDC28: The dataset CDC28 is a subdataset from yeast gene expression experiment (Cho et al., 1998) and consists of expression levels of 237 genes at 17 equally-spaced time points selected by Yeung et al. (2001). The dataset is available from the website <http://faculty.washington.edu/kayee/model/>. This dataset is used to investigate the genetic algorithm for inferring time delays in gene regulatory networks in Section 4.5.

Of these five real-life datasets above, CDC15, ELU, ALP and BAC are obtained from dual labelling approaches, while CDC28 is from single labelling approaches. In this chapter, gene expression data from dual labelling approaches are called to be intensity-ratio-type while those from single labelling approaches are intensity-type. However, in our study the types of data in these datasets have no essential difference after proper data pre-processing methods are taken (see Section 2.3).

SYN dataset: this is a synthetic dataset generated by the sine function modeling cyclic behaviour of genes employed by Yeung et al. (2001). Let x_{ij} be the synthesized expression level of gene i and time point j in the dataset and be modeled by $x_{ij} = \delta_j + \lambda_j * (\alpha_i + \beta_i \phi(i, j))$, where $\phi(i, j) = \sin(2\pi j/8 - w_{k(i)} + \varepsilon)$. α_i represents the average expression level of gene i , which is chosen according to the standard normal distribution. β_i is the amplitude control for gene i , which is chosen according to the normal distribution with mean 3 and standard deviation 0.5. λ_j is the amplitude control at time j , which is chosen according to the normal distribution with mean 3 and standard deviation 0.5. δ_j represents the additive experimental error at time point j , which is chosen according to the normal distribution with mean 0 and standard deviation 2. $\phi(i, j)$ models the cyclic behaviour of genes. Each cycle is assumed to span eight time points. There are a total of five clusters, and k is the cluster label. The sizes of different clusters are chosen according to the uniform distribution on the interval [100, 300]. Different clusters are represented by different phase shifts, and $w_{k(i)}$ represents a phase shift for gene i in cluster k , which is chosen according to the uniform distribution on the interval $[0, 2\pi]$. The random variable ε represents the noise of gene synchronization, which is chosen according to the standard normal distribution. Using the model above, a synthetic dataset is generated consisting of expression levels of 900 genes at 24 equally-spaced time points. These 900 genes belong to five clusters, which contain 127, 200, 194, 152, 227 genes, respectively. This dataset is used to investigate both the MCM- and ARM-based clustering methods in Sections 3.3 and 3.4, respectively.

2.3 Data Pre-processing

After numerical gene expression data (either intensity-type or intensity-ratio-type) have been obtained and before any further analysis is done, proper data pre-processing methods must be applied to the raw gene expression data. There are a series of operations that transform the data into a format that is suitable for specific analysis methods and reduce the possibility of statistical artefacts. These operations mainly include (Eisen et al., 1998):

- Log transformation: replace each value in the data matrix X by $\log_2(X)$. The operation is often applied to intensity-ratio-type gene expression data (from dual labelling approaches, Eisen et al., 1998) where induction and repression are values of different magnitude although they have an equal weight in nature. For example, a twofold induction will have more weight in any comparison than a one half repression. To treat them as values of identical magnitude, ratios are log transformed (the base 2 is a common choice). In this case, after a \log_2 transformation, a twofold induction would have a numerical value 1, and a one half repression a value -1. Now, both cases will present an equal weight in any comparison operation. This operation can also be applied to intensity-type gene expression data (from single labelling approaches, Yeung et al., 2001).

- Adjustment of mean centers for genes and/or arrays: subtract the row-wise and/or the column-wise mean from the values in each row and/or column of the data matrix such that mean value of each row and/or column is 0. The operation corrects the difference in overall gene and/or array expression values. For example, differences in the efficiency of probe labelling may be responsible for a non-homogeneity in the overall signal intensity between the different genes and /or arrays. In order to correct this effect data needs to be adjusted in respect to a quantity that is considered as a constant across the genes and /or arrays.
- Adjustment of median center for genes and/or arrays: subtract the row-wise and/or column-wise median from the values in each row and/or column of the data matrix such that median value of each row and/or column is 0. The purpose of this operation is exactly the same as that of immediately previous operation while their statistical meanings and robustness are different. Eisen et al. (1998) recommended the use of median rather than mean to adjust for genes and/or arrays because the former is more robust to noise than the latter. It is also noticed that the use of median for genes does not reduce the independent dimensions of gene expression profile.
- Adjustment of deviation for genes and/or arrays: Multiply all values in each row and/or column by a row-wise and/or a column-wise scale factor such that the standard deviation (from the mean or the median) of each row and/or column is 1.0. The operation makes distance measure such as the Euclidean distance more

sensible, yet does not affect correlation coefficient measures where similar genes are identified on the basis of their expression waveforms rather than on the basis of the geometry of their profiles. This is the case in most methods for cluster analysis techniques of gene expression data. This operation compresses or expands the profiles to the same scale although the shapes of the profiles are maintained.

The operations introduced above are not associative, so the order in which they are applied is very important and should be considered carefully in advance. For the intensity-ratio-type expression data from glass-based cDNA microarrays, Eisen et al. (1998) recommended that after \log_2 transformation is performed, median/mean centers for genes and arrays be adjusted alternatively for five to ten times, followed by five to ten applications of deviation adjustments for genes and arrays. However, Eisen's recommendation is debatable. In our study, the data pre-processing operations employed will be described in the context of the use of the data.

It should be noted that distinction between the intensity-ratio-type data and the intensity-type data of gene expression can be eliminated by proper application of the operations above. For example, assume there are two data matrices of gene expression from the same biological process, one in which the elements are intensity-ratio-type data (from a dual labelling approach) and one where they are intensity-type data (from a single labelling approach). Corresponding elements of the two matrices stand for expression values of the same gene at the same time point. After log transformation, the intensity-

ratio-type data becomes the difference of the log-intensity values between the treatment samples and the reference sample while the intensity-type data becomes the log-intensity values of the treatment samples. If the reference samples are the same one in gene expression experiments for the dual labelling approach, two corresponding rows in these two data matrices differ only by a constant (i.e. the log-intensity expression value of the corresponding gene as the reference sample). Therefore, after adjustment of mean (median) center of rows (the operations above) is taken, there will be no difference in principle between these two matrices (i.e., between the intensity-type data and intensity-ratio-type data of gene expression).

DYNAMIC MODEL-BASED CLUSTERING

3.1 Related Work

Clustering could be defined as a process of classifying a set of objects into a set of disjointed groups (clusters) of objects. Its goal is to reduce the amount of data by categorizing or grouping similar data items together and therefore help discover new knowledge in the underlying data (for example, gene regulatory relationships in gene expression data). There have been many clustering methods proposed for analyzing time course gene expression data to infer gene relationships and annotate gene functions. They can be divided into two groups: distance/correlation-based clustering and static model-based clustering. In this section, a brief review of these clustering methods is given.

3.1.1 Correlation/Distance-Based Clustering

The correlation/distance-based clustering algorithms may be further divided into two groups: hierarchical clustering and partitional clustering methods. They all need a

similarity measure to quantify the (dis)similarity in features between any two objects. In the case of clustering time-course gene expression, the objects are the genes while their features are a time-series of gene expression values (gene expression profiles). In these methods, gene expression profiles with m expression values are viewed as m -dimensional vectors. Thus, a gene corresponds to a point in the m -dimensional vector space. Clustering of genes simply becomes clustering of these points based on a similarity measure of the m -dimensional vectors. The choice of similarity measures may be as important as the choice of clustering algorithms

Similarity measures: Given two genes with their corresponding expression profiles $g_1 = (g_{11}, g_{12}, \dots, g_{1m})$ and $g_2 = (g_{21}, g_{22}, \dots, g_{2m})$, where m is the number of time points (conditions) at which gene expression levels are collected, and g_{ij} represents the expression value of gene i at time point (condition) j . Two types of similarity measures are extensively used in clustering of gene expression profiles: the correlation coefficient and the squared Euclidean distance.

The *correlation coefficient* is defined by

$$r(g_1, g_2) = \frac{\sum_{j=1}^m (g_{1j} - g_{1offset})(g_{2j} - g_{2offset})}{\sqrt{\sum_{j=1}^m (g_{1j} - g_{1offset})^2} \sqrt{\sum_{j=1}^m (g_{2j} - g_{2offset})^2}} \quad (3.1)$$

where $g_{i\text{offset}}$ ($i = 1, 2$) are two constants. It is obvious that $|r(g_1, g_2)| \leq 1$ and "=" holds if and only if there exists a real number λ such that $(g_1 - g_{1\text{offset}}) = \lambda(g_2 - g_{2\text{offset}})$ or

$$\frac{g_1 - g_{1\text{offset}}}{\sqrt{\sum_{j=1}^m (g_{1j} - g_{1\text{offset}})^2}} = \pm \frac{g_2 - g_{2\text{offset}}}{\sqrt{\sum_{j=1}^m (g_{2j} - g_{2\text{offset}})^2}} . r(g_1, g_2) = 1 \text{ means that genes } g_1 \text{ and}$$

g_2 have identical expression profiles subject to an affine transformation and therefore may represent a co-regulated response to a biological process or a series of stimuli in the same direction. On the other hand, $r(g_1, g_2) = -1$ means that genes g_1 and g_2 have opposite expression profiles subject to an affine transformation and therefore may represent a co-regulated response to a biological process or a series of stimuli but in the opposite direction. In practice $|r(g_1, g_2)|$ will not be exactly one owing to the presence of noise in gene expression data. A value close to one is taken as indicating the co-regulated relationships among genes.

When $g_{i\text{offset}}$ ($i = 1, 2$) are set to the means of gene expression profiles g_1 and g_2 , respectively, $r(g_1, g_2)$ is exactly the Pearson correlation coefficient of genes g_1 and g_2 .

When $g_{i\text{offset}}$ ($i = 1, 2$) are set to zeros, $r(g_1, g_2)$ is exactly the so-called "cosine" correlation coefficient of genes g_1 and g_2 (Kohonen, 1997). Besides these two cases, other settings of $g_{i\text{offset}}$ ($i = 1, 2$) are possible. For example, $g_{i\text{offset}}$ ($i = 1, 2$) may be set to the medians of gene expression profiles g_1 and g_2 to get a more robust measure than the Pearson correlation measure (Eisen et al., 1998).

The *squared Euclidean* distance is defined:

$$d(g_1, g_2) = \sum_{j=1}^m (g_{1j} - g_{2j})^2 \quad (3.2)$$

This distance measures the absolute distance between two genes in the m – dimensional gene expression space, which in this case is defined by all gene expression profiles. If used directly with non-transformed data one is considering similar gene expression profiles with similar magnitude of expression. Although this property may be significant in some cases, usually it is biologically more interesting to search for genes expressed at different levels but with the same overall profiles. That is, one identifies similar genes on the basis of their expression waveforms rather than their similarity in the geometry of the profiles. Therefore, some data pre-processing methods mentioned in Section 2.3 should be applied to gene expression data. Actually, when the gene expression data is normalized to have for each gene the mean 0 and the variance 1, the squared Euclidean distance is equivalent to the Pearson correlation measure because of the identity $d(g_1, g_2) = 2(1 - r(g_1, g_2))$. Therefore, $1 - r(g_1, g_2)$ is considered as a kind of distance for the similarity measure (Eisen et al., 1998).

Although the similarity measures, as described above, are popular in cluster analysis of the gene expression data, there are other similarity measures that have been reported, for example, the standard Euclidean distance (Wen et al., 1998), the squared Pearson correlation (D’haeseleer et al., 1998), the Spearman rank correlation (D’haeseleer et al., 1998), and the mutual information measure (D’haeseleer et al., 1998; Michaels et al., 1998; Laurie et al., 1999).

Hierarchical clustering: Hierarchical clustering proceeds successively either by merging smaller clusters into larger ones, or by splitting larger clusters into smaller clusters. The former is called *agglomerative* (bottom-up, clumping) while the latter is *divisive* (top-down, splitting). The more details about the agglomerative hierarchical clustering will be reviewed in this section.

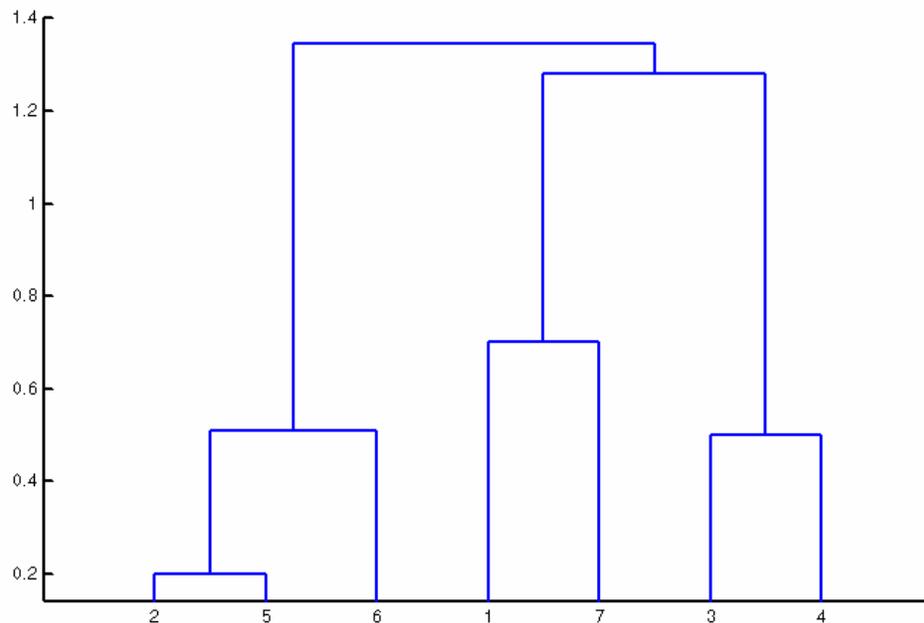


Figure 3.1 A dendrogram for hierarchically clustering 7 objects. The numbers on the horizon axis represent the indices of objects, and the numbers on the vertical axis represent the distance between the two objects (clusters) being connected.

The agglomerative hierarchical clustering begins with a similarity matrix of objects, whose (i, j) -th element is the similarity between objects i and j . Based on the

similarity matrix, another matrix called *cophenetic* matrix is formed. The (i, j) -th element of the cophenetic matrix represents the emerging similarity level at which a pair of objects i and j appears together in the same cluster for the first time during the process of a hierarchical clustering. A derived hierarchical structure is fully described by its cophenetic matrix (Legendre et al., 1998) and often visualized by using a binary tree of clusters called a *dendrogram* (Figure 3.1). A dendrogram shows how the clusters are related to each other. By cutting the dendrogram at a desired level (threshold), a partition of objects in a dataset into disjoint groups is obtained. For example, if 0.7 is chosen as the desired distance level at which two objects are considered to be in different clusters in Figure 3.1, three clusters (2,5,6), (1,7) and (3,4) can be obtained. One of the advantages with hierarchical clustering is that it allows detection of higher order relationships between clusters (Duda et al., 2001).

The hierarchical clustering methods differ in the rules used to decide which two clusters are merged. There are seven different hierarchical clustering methods which are defined in terms of the general agglomerative algorithm (Lance and Williams, 1967) by the following unified formulae:

$$d_{r(s,t)} = \alpha_s d_{rs} + \alpha_t d_{rt} + \beta d_{ts} + \gamma |d_{rs} - d_{rt}| \quad (3.3)$$

where d_{rs} denotes the distance between objects or clusters r and s , $d_{r(s,t)}$ is the distance between object or cluster r and the combined cluster (s,t) , and n_s is the

number of objects in cluster s . Different combinations of coefficients $(\alpha_s, \alpha_t, \beta, \gamma)$ in Equation (3.3) lead to the different hierarchical clustering methods (see Table 3.1).

Table 3.1 Seven agglomerative hierarchical clustering methods specified by parameters to the general agglomerative formulae of Lance and Willams (1967), given in Equation (3.3)

Methods\Parameters	α_i	β	γ
Single link	$1/2$	0	$-1/2$
Complete link	$1/2$	0	$1/2$
Average link	$n_s / (n_s + n_t)$	0	0
Median	$1/2$	$-1/4$	0
Weighted average	$1/2$	0	0
Centroid	$n_s / (n_s + n_t)$	$-n_s n_t / (n_s + n_t)^2$	0
Ward	$(n_s + n_r) / (n_s + n_t + n_r)$	$-n_r / (n_s + n_t + n_r)$	0

Although applications of divisive hierarchical clustering to gene expression data can be found (Alon et al., 1999), the agglomerative hierarchical clustering methods have become more popular in part due to the availability of implementations either in standard statistical packages or as specifically designed programs for gene expression data (Eisen et al., 1998). Currently much gene expression data from organisms,

including yeast (Chu et al., 1998; Spellman et al., 1998) and human cells (Wen et al., 1998; Iyer et al., 1999; Whitfield et al., 2002) has been analyzed by means of agglomerative hierarchical clustering. A problem with this clustering technique is that it is the highly demanding of computing resource when the large number of genes is large.

Partitional clustering: In general, any partitional clustering method requires that the number of clusters, k , is given a priori. For each object i , there is a corresponding set D_i which describe features of the object. D_i is also called the feature vector of object i . Hereafter, an object and its feature vector are not distinguishable. A k -partitional algorithm takes as input n objects and an pre-specified integer k , and partition objects into a set of disjoint subsets P_1, P_2, \dots, P_k . Each of the subsets is a cluster; objects in the same cluster are more similar to each other than they are to objects in other different clusters. One of the difficulties with the partitional clustering technique is how to define the concept of similarity, or more generally how to define quality of a particular partition. One way to address this difficulty is to define a cost function that measures the quality of any partition. Beginning with an initial partition, algorithms minimize the cost function by iterative reallocation of cluster members. Well-know partitional clustering methods includes k-means, fuzzy k-means, self-organizing maps, and so on.

A common criticism of this type of algorithm is its requirement of a pre-defined number of clusters. One response to this criticism is the so-called leader algorithm (Hartigan, 1975) that finds the number of different clusters from the data itself. However, this algorithm needs two pre-specified threshold parameters α and β . Parameter α is the

minimum admissible similarity between the representatives of two clusters for the merging of these two clusters. That is, when the similarity between the representatives of two clusters is greater than or equal to α , the two clusters are merged into a new cluster. Parameter β corresponds to the maximum admissible similarity between an object, say x , and the representatives of all existing clusters for determining whether to create a new cluster with object x as its representative. That is, when similarity between object x and the representatives of all existing clusters is less than β , a new cluster is created with object x as its representative. Another way to address the issue is to estimate a statistically reasonable number of clusters in a dataset (Calinski and Harabasz, 1974; Hartigan, 1975, 1985; Krzanowski and Lai, 1985; Kaufman and Rousseeuw, 1990; Tibshirani et al., 2000; Duoit and Fridlyand, 2002). In addition, for some specific problems, expert knowledge may be helpful to estimate an appropriate number of clusters in a dataset.

K-means algorithms have been used (Tavazoie et al., 1999) to discover distinct clusters of genes based on gene expression data and then identify *cis*-regulatory elements through which co-regulation of the genes within the cluster is achieved. Herwig et al. (1999) proposed the application of a progressive *k*-means procedure, which essentially is a variation of the leader algorithm (Hartigan, 1975) that finds the number of the different clusters from the data itself and is independent of an a priori specified number of clusters. Fuzzy *k*-means algorithms (Gasch et al., 2002; Dembele et al., 2003) and self-organizing maps (Tamayo et al., 1999; Toronen, 1999; Torkkola, 2001) have also

been used to analyze gene expression data. These partitional clustering techniques perform quite well for problems with a larger number of genes.

In a conclusion, as applied to time-course gene expression data, the distance/correlation-based clustering methods can not take the dynamics of time-course gene expression into consideration. Therefore, the quality of distance/correlation-based clustering may be degraded.

3.1.2 Static Model-Based Clustering

Clustering methods in the second group are model-based. Instead of defining a similarity measure in terms of the distance or the correlation, these methods assume that observed datasets are generated by an unknown number of probabilistic models. Each model represents a cluster. As such, the clustering process is driven by maximizing the likelihood that observed data are generated by the given models. Cluster membership in this case is decided by posterior probabilities that a gene expression profile is generated by a specific model. Model-based methods may further be divided into the static model-based (time dependence of gene expression data is not considered) and the dynamic model-based (time dependence of gene expression data is taken into account). The static model-based clustering methods typically use multivariate normal distributions as models describing clusters (for example, Yeung et al., 2001; Ghosh and Chinnaiyan, 2002; McLachlan et al., 2002). Static model-based clustering methods have the same problems

as the distance/correlation based clustering methods. That is, they do not take the valuable time-dependence information of time-course gene expression into account.

In contrast, the dynamic model-based models try to account for the dynamics of time-course gene expression data. Ramoni et al. (2002a) recently presented a Bayesian method for model-based clustering of gene expression dynamics, where the dynamics were represented by the autoregressive models, and an agglomerative procedure was used to search for the most probable set of clusters. As the number of possible sets grows exponentially with the number of observed time-course gene expression profiles, a distance-based heuristic search procedure was devised to render the search feasible. Their method has several disadvantages. First, the use of an agglomerative procedure results in an expensive computational complexity, while the use of the distance-based heuristic search procedure means that their method has all inherent disadvantages with the distance-based hierarchical clustering. Second, both a set of autoregressive coefficients and p initial values are needed to determine a curve (gene expression profile) described by the autoregressive model with order p (see Section 3.4 for more details). How the p initial values are determined was not discussed in their method.

Other model-based clustering methods for (state) sequences (Smyth 1999; Cadez et al., 2000; Ramoni et al., 2002b) may be employed to cluster time-course gene expression data. The state sequences in these methods are modeled by hidden Markov models (HMMs) or Markov chain models (MCMs). In clustering genes based on gene expression dynamics, it is not necessary to model the state sequence by sophisticated

HMMs if one does not consider either missing data or time delay in a gene regulatory process. Ramoni et al. (2002b) discussed the advantages of modeling sequences as MCMs over HMMs. By modelling sequences as MCMs, they proposed a clustering method under the setting of a hierarchical clustering technique. Their method builds MCM for each sequence and computes the distances of the pair-wise MCMs as the distances of the corresponding sequences. As a result, the complexity of their algorithm is $O(N^4M^2)$. Therefore, it is considerably expensive to apply their algorithm to a large-scale dataset such as gene expression datasets considered in this study.

3.2 Clustering Validations

The term clustering validation usually refers to the ability of a given method to recover true clusters in a dataset. There have been several attempts to evaluate a clustering method on theoretical grounds (Theodoridis and Koutroumbas, 1999). This section describes three kinds of methods for validating clustering: internal index methods, external index methods, and a bootstrapping method.

3.2.1 Internal Indices

Hierarchical clustering validation: Although many measures for validating hierarchical clusterings have been developed (Hartigan, 1967; Sokal, 1962), one common measure is the cophenetic correlation coefficient (Everitt and Dunn, 1992; Duda et al., 2001) of the similarity matrix S (consisting of the similarity measures of all pair-wise objects) and its

cophenetic matrix C (induced from a hierarchical clustering); and it is defined as follows:

$$r(C, S) = \frac{\sum_{i=2}^n \sum_{j=1}^{i-1} (c_{ij} - \bar{c})(s_{ij} - \bar{s})}{\sqrt{\sum_{i=2}^n \sum_{j=1}^{i-1} (c_{ij} - \bar{c})^2} \sqrt{\sum_{i=2}^n \sum_{j=1}^{i-1} (s_{ij} - \bar{s})^2}} \quad (3.4)$$

where c_{ij} and s_{ij} are the values of the (i, j) -th entry in matrices C and S , respectively. \bar{c} and \bar{s} are the average values of all elements below the main diagonals of matrices C and S , respectively. The value $|r(C, S)|$ has range $[-1, 1]$. The closer the value of the cophenetic correlation coefficient is to 1, the better the clustering is. The cophenetic correlation coefficient is essentially the Pearson correlation coefficient of the similarity matrix and its induced cophenetic matrix. Accordingly, comparing the orders (ranks) of the elements in matrices C and S is more objective than directly comparing the elements in matrices C and S for validating the hierarchical clustering (Sokal and Rohlf, 1962; Baker, 1974; Cunningham and Ogilvie, 1972). Thus, to validate a hierarchical clustering, one can also compute a Spearman rank correlation coefficient (Hays, 1973) between the similarity matrix and its induced cophenetic matrix. However, from a statistical point of view a product moment correlation coefficient is not well suited to this case due to the ordinal data.

With ties existing in ranks of the similarity matrix and its induced cophenetic matrix, the gamma index of Goodman and Kruskal (1954) is well suited to this case (Baker, 1974; Hays, 1973). The gamma index can be defined as the difference between two

conditional probabilities for two object pairs selected at random from all possible object pairs and untied on both ranking (Hays, 1973), i.e.,

$$\gamma = p(\text{same ordering} \mid \text{untied pairs}) - p(\text{different ordering} \mid \text{untied pairs}) \quad (3.5)$$

where a same ordering for two object pairs means that both rankings give a higher rank to one of the two object pairs. The value of the Gamma index has range $[-1, 1]$ where the larger value indicates perfect agreement between the two rank orderings. Hays (1973) introduced a good method to calculate the gamma index.

Partitional clustering validation: For the validation of partitional clustering, there are three aspects that require attention. The first is concerned with the quality of a clustering, given the number of clusters, k . One usual way to evaluate the quality of a partitional clustering is by use of *silhouette index* (Rosseeuw, 1987; Chen et al., 2001). The silhouette index assesses the ratio of inter-cluster separation and intra-cluster similarity. For each object i in a dataset containing n objects, its *silhouette width* $s(i)$ is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.6)$$

where $a(i)$ is the average distance of object i to other objects in the same cluster. $b(i)$ is the average distance of object i to other objects in its nearest “neighbour” cluster and

can be seen as the dissimilarity between object i and the nearest neighbour cluster to which it does not belong. Objects with a large $s(i)$ (almost 1) are very well clustered, a small $s(i)$ (around 0) means that the object lies between two clusters, and objects with a negative $s(i)$ are probably placed in the wrong cluster. The average $s(i)$ across all objects for a clustering

$$S = \frac{1}{n} \sum_{i=1}^n s(i) \quad (3.7)$$

is called the silhouette index of the clustering. The value of S reflects the overall quality of the clustering and has range $[-1, 1]$. A larger value of S indicates a better overall quality of the clustering.

3.2.2 External Indices

Since a clustering result can be considered as a partition of objects into a number of groups, one possible way to validate a clustering result is to compare the cluster labels from it with known cluster labels. The generic problem is thus to define a measure of the agreement between two partitions of the same dataset. In the clustering literature, measures of agreement between two partitions are referred to as external indices. Several such indices have been proposed (Theodoridis and Koutroumbas, 1999; Dudoit and Fridlyland, 2002). In the following, some of these indices are introduced.

Table 3.2 Contingency table for two partitions of n objects

	v_1	v_2	...	v_s	Total
u_1	m_{11}	m_{12}	...	m_{1s}	$m_{1.}$
u_2	m_{21}	m_{22}	...	m_{2s}	$m_{2.}$
\vdots	\vdots	\vdots		\vdots	\vdots
u_r	m_{r1}	m_{r2}	...	m_{rs}	$m_{r.}$
Total	$m_{.1}$	$m_{.2}$...	$m_{.s}$	$m_{..} = n$

Consider two partitions of n objects $\mathbf{x}_1, \dots, \mathbf{x}_n$: the r -partition $U = \{u_1, \dots, u_r\}$ and the s -partition $V = \{v_1, \dots, v_s\}$. One can define the matrix (or the contingency table)

$M = [m_{ij}]$, where entry m_{ij} is the number of objects that are both in clusters u_i and v_j ,

$i = 1, \dots, r$, $j = 1, \dots, s$. Let $m_{i.} = \sum_{j=1}^s m_{ij}$ and $m_{.j} = \sum_{i=1}^r m_{ij}$ denote the sums of row

i ($i = 1, \dots, r$) and column j ($j = 1, \dots, s$) in matrix $M = [m_{ij}]$, respectively, and let

$$Z = \sum_{i=1}^r \sum_{j=1}^s m_{ij}^2 \text{ and } T = \binom{n}{2} = n(n-1)/2 \text{ (the number of pairs of } n \text{ objects).}$$

Based on matrix $M = [m_{ij}]$, the following indices to measure the agreement between

two partitions $U = \{u_1, \dots, u_r\}$ and $V = \{v_1, \dots, v_s\}$ have been proposed :

- Rand (Rand, 1971):

$$Rand = 1 + [Z - \frac{1}{2}(\sum_{i=1}^r m_i^2 + \sum_{j=1}^s m_j^2)] / T \quad (3.8)$$

- FM (Fowlkes and Mallows, 1983) :

$$FM = \frac{1}{2}(Z - n) / [\sum_{i=1}^r \binom{n_i}{2} + \sum_{j=1}^s \binom{n_j}{2}] \quad (3.9)$$

- Jaccard (Jain and Dubes, 1988):

$$Jac = (Z - n) / (\sum_{i=1}^r m_i^2 + \sum_{j=1}^s m_j^2 - Z - n) \quad (3.10)$$

An external index is often adjusted in such a way that its expected value is 0 when the two partitions are selected at random and 1 when they match perfectly. One of the commonly used indices is the adjusted Rand index defined as

- Adjusted Rand Index (ARI) (Kreiger and Green, 1999):

$$ARI = \frac{\sum_{i=1}^r \sum_{j=1}^s \binom{n_{ij}}{2} - \frac{1}{T} \sum_{i=1}^r \binom{n_i}{2} \sum_{j=1}^s \binom{n_j}{2}}{\frac{1}{2} \left[\sum_{i=1}^r \binom{n_i}{2} + \sum_{j=1}^s \binom{n_j}{2} \right] - \frac{1}{T} \sum_{i=1}^r \binom{n_i}{2} \sum_{j=1}^s \binom{n_j}{2}} \quad (3.11)$$

3.3.3 A Bootstrapping Method

1. Repeat the following B times (where B is a preset integer number)
 - (a) Randomly divide the original dataset into two non-overlapping sets, a learning set L and a test set T .
 - (b) Apply the target method to the learning set L to obtain a partition $P(\circ, L)$.
 - (c) Construct a predictor (classifier) $C(\circ, L)$ using the cluster labels from the partition $P(\circ, L)$.
 - (d) Apply the predictor $C(\circ, L)$ to the test set T to get the predicted partition $\tilde{P}(\circ, T)$.
 - (e) Apply the target method to the test set T to obtain a partition $P(\circ, T)$.
 - (f) Calculate the ARI of partitions $\tilde{P}(\circ, T)$ and $P(\circ, T)$.
2. Calculate the average ARI (AARI) over the B times as the measure index of the evaluated clustering method.
3. For the various number of clusters, K , repeat the procedure described in steps (1) and (2) above to get $\text{AARI}(K)$, and then plot $\text{AARI}(K)$ with respect to K .

Figure 3.2 The procedure for validating clustering methods

In many studies, the performance of clustering methods is evaluated upon datasets where true cluster labels are known (e.g., Yeung et al., 2001; Dougherty et al., 2002). However, for real-life gene expression datasets the true cluster labels are typically unknown. Furthermore, since microarray technology is still in its infancy, data created with this technology may be noisy. There may also be information in real data which is

not known to biologists. Even though clustering algorithms may determine a cluster label for each profile in a dataset, many of these cluster labels may be false negatives or false positives. Therefore, it is worthwhile to have an alternative approach to evaluate the clustering methods without either resorting to seed clustering methods or requiring *a priori* known cluster labels. At this point, a procedure is proposed in this study as shown in Figure 3.2. This procedure is primarily based on a procedure proposed by Breckenridge (1989) under the name of replicating cluster analysis and was designed to evaluate the stability of a clustering. Recently Dudoit and Fridlyand (2002) employed this approach to estimate the number of clusters in a dataset. For the given number of clusters, K , the average ARI (AARI) reports the quality of the clustering result obtained from the clustering methods under consideration (Kreiger and Green, 1999). The ARI ranges from -1 to 1, and so does AARI. Accordingly, the larger AARI, the better the quality of the clustering, i.e., the better the performance of the clustering method (Kreiger and Green, 1999).

Note that this procedure will be applied to validate dynamic model-based clustering methods to be discussed in the next two sections, especially for the real-life gene expression datasets. For the synthetic dataset ARI (Equation 3.11) will be applied.

3.3 Markov Chain Model-Based Clustering

A Markov chain model (MCM)-based clustering method for time-course gene expression data is proposed and described in this section (Wu et al., 2004b). After a

transformation of gene expression profiles into gene expression state sequences, the dynamics of gene expression is represented by MCMs. All gene expression state sequences are assumedly created by a mixture model of MCMs in which MCM corresponds to a different cluster. For the given number of clusters, the proposed method finds distinguished cluster MCMs using the EM algorithm and assigns each gene to one specific cluster if its posterior probability being in this cluster is the largest.

Compared to the method proposed by Ramoni et al. (2002b), the method proposed in this study does not need either to build an MCM for all individual genes or to compute the distances between pair-wise genes. Instead the proposed method views a cluster as a MCM and computes the probability that an individual gene fits a MCM. This reduces the computational complexity of the proposed method to $O(NM)$ (see Section 3.3.2). Further, this study employs AARI to evaluate the quality of clustering. The superior performance of the proposed method is demonstrated by comparing to the k-means method on datasets SYN and BAC as described in Section 2.2.

3.3.1 Gene Expression Dynamics Sequence

The aim of clustering genes based on their expression profiles is to group the co-regulated genes in an underlying biological process into the same cluster. Three types of regulatory states, induction (I), repression (R), and constant (C), are considered here. In order to convert log-transformed gene expression values into gene expression states, the following steps are taken:

- (1) Normalize gene expression profiles to each have a median of zero and a standard deviation (from the median) of one;
- (2) Convert the normalized expression profiles into gene expression state sequences. The assignment of the normalized gene expression values to these states is based on two threshold parameters: a positive number d_1 and a negative number d_2 . The expression values between these two parameters are classified as C. The expression values greater than the threshold parameter d_1 are classified as I, while the expression values less than the threshold parameter d_2 are classified as R.

Once the parameters d_1 and d_2 are given (see Section 3.3.3), using the procedure above, a set of time-course gene expression profiles can be converted into a set of gene expression state sequences over the alphabet $S = \{I, R, C\}$, in which each sequence corresponds to one gene. If one is hypothesizing that two genes are co-regulated, one would expect to see that these two expression state sequences are similar (i.e., have similar dynamics). To account for the time dependence of gene expression in cluster analysis, this study employs MCMs for modelling dynamics of gene expression state sequences. Genes in the same cluster were assumed to be generated by the same cluster model. Each cluster model is described by an initial state probability distribution represented by a 3-dimensional vector $\pi(s)$ and a 3×3 state transition probability

matrix T , whose (i, j) -th element is the probability of the transition from states s_j to s_i , represented by $p(s_i | s_j)$, where s, s_i , and s_j come from the alphabet $S = \{\text{I, R, C}\}$.

3.3.2 MCM-Based Clustering Method and EM Algorithm

Let K be the number of the components in the mixture model and $p(D_n | \pi_k, T_k)$ be the probability that the k -th MCM generates the expression state sequence of gene n , D_n , where (π_k, T_k) ($1 \leq k \leq K$) are the parameters of the k -th MCM. With these notations, the likelihood that the set of gene expression state sequences $D = \{D_1, \dots, D_N\}$ is generated by a mixture model of K MCMs can be written as:

$$p(D | \Theta) = \prod_{n=1}^N p(D_n | \Theta). \quad (3.12)$$

where $p(D_n | \Theta)$ is the likelihood that state expression sequence D_n is generated by the mixture model and can further be written as

$$p(D_n | \Theta) = \sum_{k=1}^K p(D_n | \pi_k, T_k) \alpha_k \quad (3.13)$$

where α_k ($1 \leq k \leq K$) is the probability that a gene belongs to the k -th cluster, the

parameters Θ consist of $\{(\pi_k, T_k, \alpha_k), k = 1, \dots, K\}$, and $0 \leq \alpha_k \leq 1$ and $\sum_{k=1}^K \alpha_k = 1$.

Initialize α_{nk} given the number of clusters K

Repeat M-step: compute maximum-likelihood parameter estimates given α_{nk}

$$\alpha_k = \frac{1}{N} \sum_{n=1}^N \alpha_{nk}$$

$$\pi_k(s) = \frac{\sum_{n=1}^N \alpha_{nk} \delta(s, D_{i1})}{\sum_{n=1}^N \alpha_{nk}}$$

$$T_k(s_i | s_j) = \frac{\sum_{n=1}^N \alpha_{nk} r_n(s_i \rightarrow s_j)}{\sum_{n=1}^N \alpha_{nk}}$$

E-step: compute α_{nk} given the parameter estimates from the M-step

$$\alpha_{nk} = \frac{p(D_n | \pi_k, T_k) \alpha_k}{\sum_{k=1}^K p(D_n | \pi_k, T_k) \alpha_k}, 1 \leq k \leq K$$

Until a convergence criterion is satisfied

Figure 3.3 The EM algorithm for MCM-based clustering. The term α_{nk} represents the probability that object n belongs to the k -th cluster and describes a partition of N objects. The term $\pi_k(s)$ represents the probability that the MCM for the k -th cluster starts with state s . The term $T_k(s_i | s_j)$ represents the transition probability from s_i to state s_j in the MCM for the k -th cluster. The term $r_n(s_i \rightarrow s_j)$ represents the number of transitions from state s_i to state s_j in expression dynamics of gene n , and the term $\delta(s, D_{i1})$ is equal to 1 when $s = D_{i1}$ and 0 otherwise.

Now, the task of MCM-based clustering is to estimate the parameters in model (3.12), and then to use the posterior probabilities to assign each gene to an appropriate cluster. In this study, the EM algorithm (Dempster et al., 1977) is employed to estimate the parameters in model (3.12). Given observation data (gene expression state sequences) D_1, \dots, D_N of N genes, the EM algorithm maximizes the log-likelihood:

$$L(\Theta) = \log(p(D | \Theta)) = \sum_{n=1}^N \log(p(D_n | \Theta)) \quad (3.14)$$

to obtain the maximum likelihood estimates of the parameters in model (3.12). Figure 3.3 shows the EM algorithm for MCM-based clustering.

The EM algorithm, as shown in Figure 3.3, iterates between an E-step in which the values of α_{nk} are computed from the data with the current model parameter estimates and an M-step in which the values of the maximum-likelihood model parameters are computed with the current values of α_{nk} . At convergence, the maximum likelihood estimates $\hat{\Theta}$ of the parameters Θ in model (3.12) are obtained. In this study, the hat “^” over a letter stands for the estimate of the corresponding parameter as the EM algorithm converges. With the estimates of model parameters $\hat{\Theta}$, the posterior probabilities are calculated (see Section 3.3.3). The classification rule is that a gene is assigned to a cluster if its posterior probability of being in that cluster is the largest. Accordingly,

such a classification rule has the least misclassification rate under the Bayesian meaning (Fraley and Raftery, 1998).

There are two issues to address before the EM algorithm runs. The first issue is the initialization of α_{nk} for the given number of clusters, K . There are several ways to do this. The simplest way is to randomly assign objects to one of the K clusters. Another way is to employ the partition from either the hierarchical clustering techniques or the partitional clustering techniques using the edit distance to measure the dissimilarity between two gene expression state sequences (Duda et al., 2001; Kohonen, 1997). In the context of clustering gene expression, one may also use a partition based on gene expression profiles to get the initialization of α_{nk} . By these ways, the initialized value of α_{nk} is 1 if gene n belongs to the k -cluster and 0 otherwise. The second issue is to choose a suitable convergence criterion. Unfortunately, there are no standard methods to do this. However, there are two heuristics which are often used to judge the convergence of the EM algorithm. One is to set a maximum number of iterations. Another is to set a cut-off value for (relative) differences between two consecutive iterations. In the following computational experiments, the initial partitions will be randomly selected and the algorithm considered to have convergence if the relative differences between two consecutive iterations is less than 10^{-6} .

The computational complexity of the algorithm at a high level is the same as the EM algorithm for the standard multivariate normal mixtures (Fraley and Raftery, 1998), i.e., linear in the total number of objects, N , in the total number of samples, M , and in the

number of iterations of the EM algorithm. For the mixture of MCMs, the complexity of computing the E-step and the M-step is linear in the total number of discrete symbols, NM , and the overall complexity of the algorithm retains its linearity.

3.3.3 Computational Experiments and Results

Computational experiments in this section use the synthetic dataset (SYN) and the real-life gene expression dataset (BAC) described in Section 2.2. These datasets are normalized to have a median of 0 and a standard deviation (from the median) of 1 for each gene and further normalized as so to have a mean of 0 and a standard deviation of 1 for all genes at each time point (array).

Table 3.3 The parameters in model (3.15) for dataset SYN

k (states)	$\hat{\beta}_k$	$\hat{\mu}_k$	$\hat{\sigma}_k^2$
1 (I)	0.3071	1.1961	0.0668
2 (C)	0.3255	0.1136	0.1906
3 (R)	0.3674	-1.1007	0.0754

To convert the normalized expression profiles into state sequences, all gene expression values are classified into three groups standing for three states I, C, and R, respectively. It is assumed that values in these three clusters come from three distinguished normal

distributions. That is, each expression value x comes from a normal mixture distribution with the probability density function:

$$f(x; \Phi) = \sum_{k=1}^3 \beta_k N(x; \mu_k, \sigma_k^2) \quad (3.15)$$

where $N(x; \mu_k, \sigma_k^2)$ stands for the normal density function with mean μ_k and variance σ_k^2 , β_k 's stands for the mixing portions, and their sum is 1, and Φ stands for all parameters in (3.15) consisting of $\{(\beta_k, \mu_k, \sigma_k^2), k = 1, 2, 3\}$.

Table 3.4 The parameters in model (3.15) for dataset BAC

k (states)	$\hat{\beta}_k$	$\hat{\mu}_k$	$\hat{\sigma}_k^2$
1 (I)	0.2647	1.1703	0.4290
2 (C)	0.4869	-0.0454	0.1989
3 (R)	0.2485	-1.1578	0.3719

Again by applying the EM algorithm (Dempster et al., 1977; Jain and Dubes, 1988), the parameters in (3.15) are estimated and listed in Tables 3.3 and 3.4 for datasets SYN and BAC, respectively. For both datasets, the means of the cluster standing for state C is close to zero, the means of the cluster standing for state I are close to 1, and the means of the cluster standing for state R are close to -1. These results are in agreement with intuition.

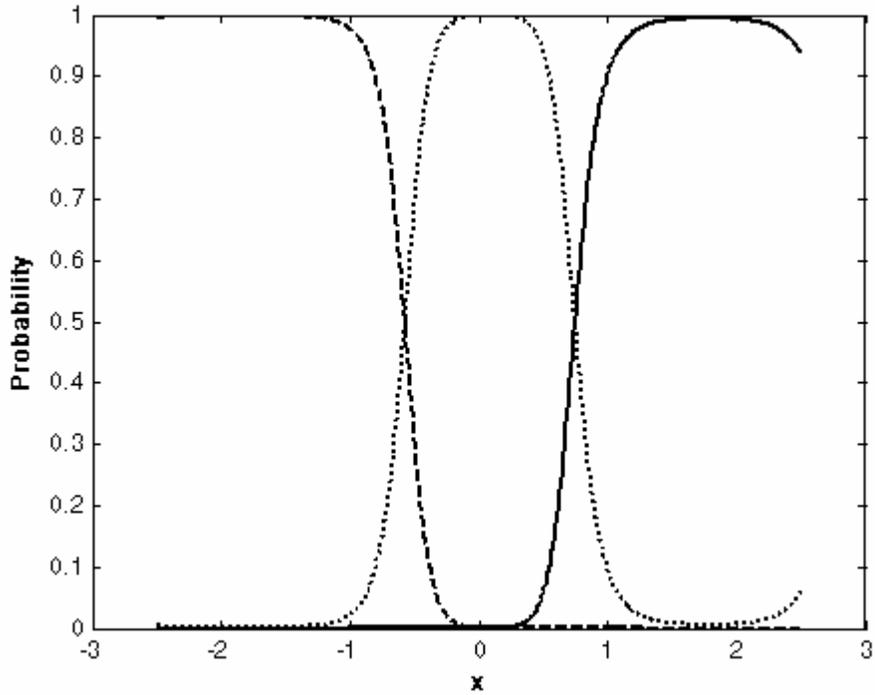


Figure 3.4 Posterior probability of a gene expression value being in each cluster (state) for dataset SYN

The posterior probabilities are calculated using the following equation:

$$\beta_{xk} = \frac{N(x; \hat{\mu}_k, \hat{\sigma}_k^2) \hat{\beta}_k}{\sum_{k=1}^3 N(x; \hat{\mu}_k, \hat{\sigma}_k^2) \hat{\beta}_k} \quad (3.16)$$

and Figures 3.4 and 3.5 show the posterior probability for datasets SYN and BAC, respectively, where solid lines stand for the probability distribution of state I, dotted lines for the probability distribution of state C, and dashed lines for the probability

distribution of state R. A gene expression value is classified as a state if its posterior probability of being in this state is the largest.

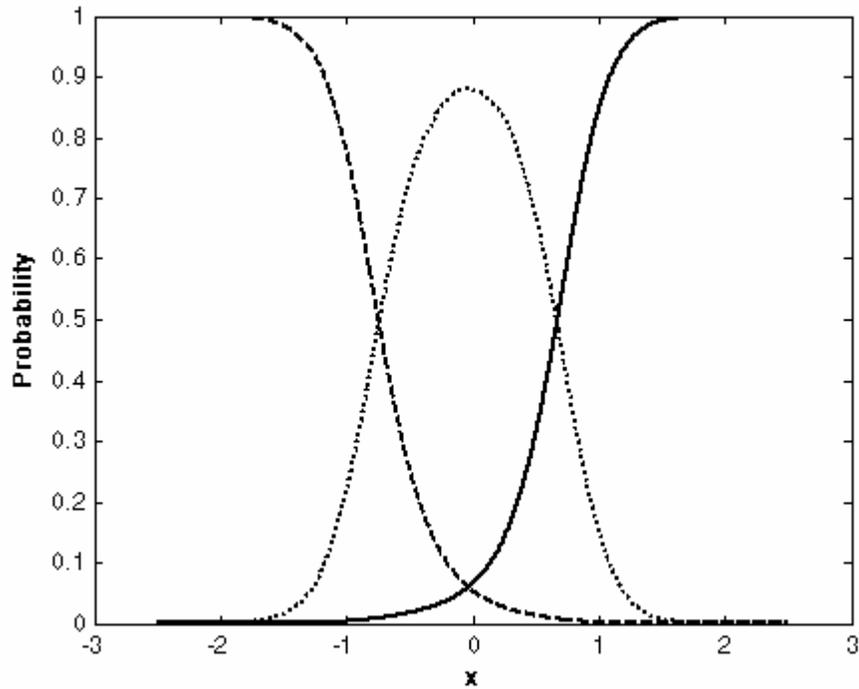


Figure 3.5 Posterior probability of a gene expression value being in each cluster (state) for dataset BAC

From Figures 3.4 and 3.5, it can be seen that if an expression value has a large positive number, then it will be classified as state I, that if it has a large negative number, then it will be classified as state R, and otherwise it will be classified as state C. Specifically, from Figure 3.4, two threshold parameters $d_1 = 0.74$ and $d_2 = -0.58$ are obtained for dataset SYN; and from figure 3.5 two threshold parameters $d_1 = 0.67$ and $d_2 = -0.74$ are obtained for dataset BAC.

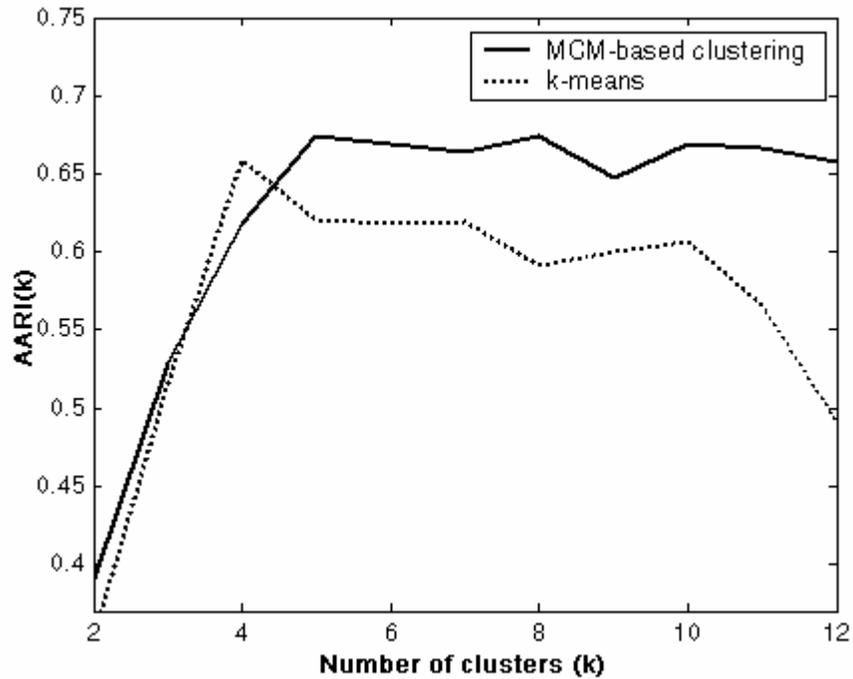


Figure 3.6 Profile of AARI with respect to the number of clusters for dataset SYN

Figure 3.6 shows the results (AARI) of the calculation for dataset SYN over a variety of the numbers of clusters, where for each number of clusters, AARI is calculated based on 20 runs of the algorithms in Figure 3.3. To compare the proposed clustering methods with the k-means clustering methods, AARI (based on 20 runs of the k-means clustering method) is also calculated for the same dataset. Figure 3.6 shows that AARI's with the proposed method are bigger than those with k-means for all numbers of clusters except 4. In particular, at $k = 5$ the AARI with the proposed MCM-based clustering is 0.67, and bigger than 0.62 (the AARI with k-means clustering). Recall that dataset SYN does contain 5 clusters. Therefore, the quality of clustering with the MCM-based clustering method is better than that with the k-means method.

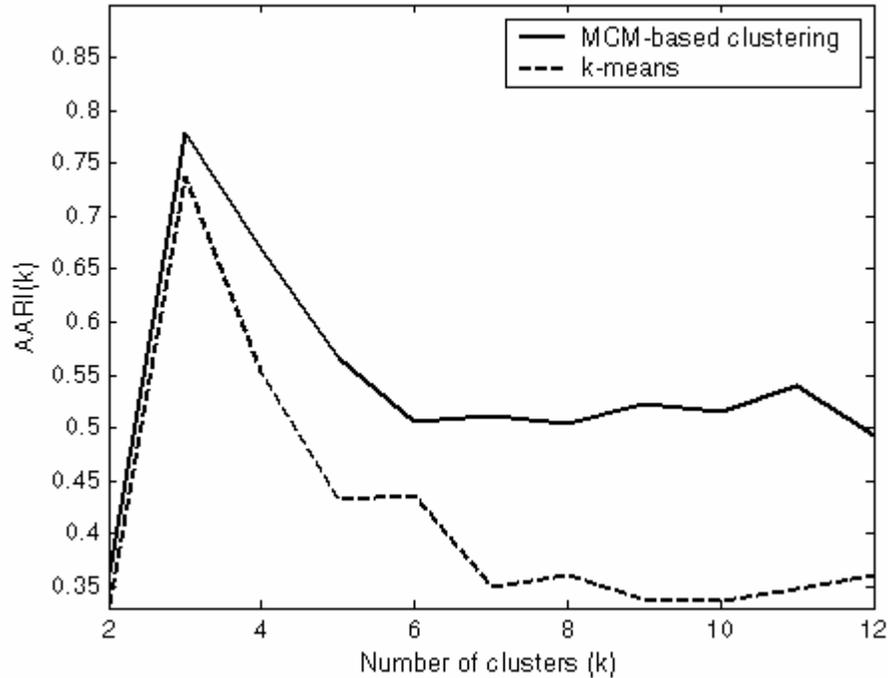


Figure 3.7 Profile of AARI with respect to the number of clusters for dataset BAC

Similarly, the EM algorithm for clustering with dynamics in Figure 3.3 and the procedure for clustering evaluation in Figure 3.2 with parameter $B = 20$ are run on dataset BAC. To compare the proposed clustering methods with the k-means clustering methods, the k-means clustering method and the procedure of clustering evaluation in Figure 3.2 with parameter $B = 20$ are also run on the same dataset. The results are depicted in Figure 3.7. The quality of clustering from MCM-based clustering is always better than that of clustering from k-means according to figure 3.7. This indicates that MCM-based clustering method outperforms the k-means method.

Note that the k-means method is a kind of model-based clustering method (Yeung et al., 2001), yet it employs a multivariate normal distribution for each component, and thus a

static model. The proposed method (i.e., MCM-based clustering) is a kind of dynamic model-based clustering since it accounts for the time dependence of gene expression. The above discussion, especially the two computational experiments, implies that accounting for the time-dependence of time-course gene expression data can improve the quality of clustering.

3.4 Autoregressive Model-Based Clustering

This study also proposes an autoregressive model (ARM)-based clustering method for time-course gene expression data, which regards a set of time-course gene expression profiles as a set of observed time series X , generated by a preset number of autoregressive models. In the ARM-based clustering methods (Wu et al., 2004c), each cluster is represented by an autoregressive model of order p (Harvey, 1993), and p initial values are modelled by a p -variate normal distribution. Thus two genes are considered as similar if they are generated by the same autoregressive model.

The task of the ARM-based clustering is to divide a given set of time-course gene expression profiles into the number of disjoint subsets (clusters) such that time-course profiles in the same cluster are generated by the same autoregressive model, and the likelihood that all profiles are generated by a mixture model with a number of autoregressive models is maximized. The cluster membership of a specific time-course gene expression profile is determined by the posterior probabilities that this gene expression profile is generated by the autoregressive models.

A bootstrapping method and an average adjusted Rand index (AARI) are used to measure the quality of clustering. Datasets SYN, ALP, ELU and BAC described in Section 2.2 are used to investigate the performance of the proposed method in this section. A comparison of the proposed method with the k-means methods is presented to highlight the performance of the proposed method.

3.4.1 Autoregressive Model and Likelihood for a Single Time Series

Let $x = \{x_1, \dots, x_m, \dots, x_M\}$ be a time series of continuous values with M equally time-spaced observations. The time series follows an autoregressive model of order p , denoted by $AR(p)$. Assuming that the current observed value x_m ($m > p$) is a linear combination of the observed values at the previous p steps plus a term representing the errors. More formally, an autoregressive model of order p may be written as,

$$x_m = a_1 x_{m-1} + \dots + a_p x_{m-p} + \varepsilon_m, \quad m = p+1, \dots, M \quad (3.17)$$

where a_i ($i = 1, \dots, p$) are the autoregressive coefficients, and ε_m ($m = p+1, \dots, M$) represent the errors. It is assumed here that the errors are subject to a normal distribution independent of time with mean 0 and variance σ^2 . Thus x_m ($m = p+1, \dots, M$), conditional on $(x_{m-1}, \dots, x_{m-p})$, are subject to a normal distribution with mean $a_1 x_{m-1} + \dots + a_p x_{m-p}$ and variance σ^2 , i.e.,

$$\begin{aligned}
& p(x_m \mid \sigma^2, x_{m-1}, \dots, x_{m-p}) \\
& \qquad \qquad \qquad m = p+1, \dots, M \qquad (3.18) \\
& = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x_m - a_1 x_{m-1} - \dots - a_p x_{m-p})^2}{2\sigma^2},
\end{aligned}$$

Further, the log-likelihood function that time series x is generated by an autoregressive model of order p with coefficients a_i ($i = 1, \dots, p$) can be written as:

$$\begin{aligned}
L(x \mid a, \sigma^2, x_1, \dots, x_p) &= \log p(x_1, \dots, x_p) + \sum_{m=p+1}^M \log p(x_m \mid \sigma^2, x_{m-1}, \dots, x_{m-p}) \\
& \qquad \qquad \qquad (3.19) \\
&= \log p(x_1, \dots, x_p) - \frac{M-p}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{m=p+1}^M (x_m - a_1 x_{m-1} - \dots - a_p x_{m-p})^2
\end{aligned}$$

where $a = [a_1, \dots, a_p]^T$, and $p(x_1, \dots, x_p)$ is the joint probability distribution of the first p observations of time series x .

The distribution of the first p observations remains to be discussed. Note that Ramoni et al. (2002b) did not address the distribution of the first p observations when they presented time-course gene expression profiles by autoregressive models. Time series (gene expression profiles) are fully determined only by both p autoregressive coefficients a_i ($i = 1, \dots, p$), and p initial observation values. Indeed, two time series that are generated by the same order p autoregressive model but different initial p

observations may have very different behaviours (Harvey, 1993; Kedem and Fokianos, 2002). This study assumes the first p observations have a multivariate normal distribution with mean $\mu = (\mu_1, \dots, \mu_p)$ and covariance matrix $\Sigma = \sigma_0^2 I_p$ (I_p represents the $p \times p$ identity matrix), i.e.,

$$p(x_1, \dots, x_p | \sigma_0^2, \mu) = \left(\frac{1}{\sqrt{2\pi\sigma_0^2}} \right)^p \exp \left[\frac{-\sum_{i=1}^p (x_i - \mu_i)^2}{2\sigma_0^2} \right] \quad (3.20)$$

This assumption is inspired by the k-means method which assumes that all the M observations have a multivariate normal distribution with mean $\mu = (\mu_1, \dots, \mu_M)$ and covariance matrix $\Sigma = \sigma_0^2 I_M$ (McLachlan and Basford, 1988; Duda et al., 2001).

At this point, the log-likelihood function that time series x is generated by an autoregressive model of order p with coefficients a_i ($i=1, \dots, p$) and initial p observations following the normal distribution (3.20) can be written:

$$\begin{aligned} L(x | \mu, \sigma_0^2, a, \sigma^2) &= \log p(x | \mu, \sigma_0^2, a, \sigma^2) \\ &= -\frac{M-p}{2} \log(2\pi\sigma^2) + \frac{p}{2} \log(2\pi\sigma_0^2) - \frac{1}{2\sigma_0^2} \sum_{i=1}^p (x_i - \mu_i)^2 \\ &\quad - \frac{1}{2\sigma^2} \sum_{m=p+1}^M (x_m - a_1 x_{m-1} - \dots - a_p x_{m-p})^2 \end{aligned} \quad (3.21)$$

Let x_0 be the vector $[x_1, \dots, x_m]^T$, y be the vector $[x_{p+1}, \dots, x_M]^T$, and X be the $(M - p) \times p$ regression matrix whose m -th row is $[x_{m-1}, \dots, x_{m-p}]$ for $m = p + 1, \dots, M$.

Then equation (3.21) may be rewritten in a vector-matrix form as:

$$\begin{aligned}
L(x | \mu, \sigma_0^2, a, \sigma^2) &= \log p(x | \mu, \sigma_0^2, a, \sigma^2) \\
&= -\frac{M-p}{2} \log(2\pi\sigma^2) - \frac{p}{2} \log(2\pi\sigma_0^2) \\
&\quad - \frac{1}{2\sigma_0^2} (x_0 - \mu)^T (x_0 - \mu) - \frac{1}{2\sigma^2} (y - Xa)^T (y - Xa)
\end{aligned} \tag{3.22}$$

3.4.2 ARM-Based Clustering

The mixture model: Let K be the number of clusters in a given set of observed time series $X = \{x_1, \dots, x_n, \dots, x_N\}$, where x_n ($n = 1, \dots, N$) stand for time series. Let $\mu_k = [\mu_{k1}, \dots, \mu_{kp}]^T$, σ_{0k}^2 , $a_k = [a_{k1}, \dots, a_{kp}]^T$, and σ_k^2 be the mean vector and the variance of the first p values, and the autoregressive coefficient vector and the variance of autoregressive model for the k -th cluster, respectively. With these notations, the task of dynamic model-based clustering is to compute a partition $C = \{C_1, \dots, C_k, \dots, C_K\}$ of the set X and $AR(p)$ models $(\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)$ ($k = 1, \dots, K$) by maximizing the likelihood function

$$f(C) = p(X | \Theta) = \prod_{k=1}^K \prod_{x \in C_k} p(x | \mu_k, \sigma_{0k}^2, a_k, \sigma_k^2) \quad (3.23)$$

or the log-likelihood function

$$\log f(C) = \log p(X | \Theta) = \sum_{k=1}^K \sum_{x \in C_k} \log p(x | \mu_k, \sigma_{0k}^2, a_k, \sigma_k^2) \quad (3.24)$$

where the parameters Θ of the mixture model (3.23) or (3.24) consist of $\{(\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2), k = 1, \dots, K\}$.

Estimation of model parameters: Following the log-likelihood function for a single time series (3.22), the log-likelihood function for multiple time series in cluster C_k , generated by the same $AR(p)$ model $(\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)$, can be written as:

$$\begin{aligned} L_k(C_k | \mu_k, \sigma_{0k}^2, a_k, \sigma_k^2) &= \sum_{x \in C_k} \log p(x | \mu_k, \sigma_{0k}^2, a_k, \sigma_k^2) \\ &= -\frac{(M-p)|C_k|}{2} \log(2\pi\sigma_k^2) - \frac{p|C_k|}{2} \log(2\pi\sigma_{0k}^2) \\ &\quad - \frac{1}{2\sigma_{0k}^2} \sum_{x \in C_k} (x_0 - \mu_k)^T (x_0 - \mu_k) - \frac{1}{2\sigma_k^2} \sum_{x \in C_k} (y - Xa_k)^T (y - Xa_k) \end{aligned} \quad (3.25)$$

where $|C_k|$ represents the number of time series in cluster C_k , $\sum_{k=1}^K |C_k| = N$.

Substituting (3.25) into (3.24) yields:

$$\begin{aligned} \log f(C) &= \log p(S | \Theta) = \\ & - \sum_{k=1}^K |C_k| \left(\frac{(M-p)}{2} \log(2\pi\sigma_k^2) + \frac{p}{2} \log(2\pi\sigma_{0k}^2) \right) \\ & - \sum_{k=1}^K \frac{1}{2\sigma_{0k}^2} \sum_{x \in C_k} (x_0 - \mu_k)^T (x_0 - \mu_k) + \sum_{k=1}^K \frac{1}{2\sigma_k^2} \sum_{x \in C_k} (y - Xa_k)^T (y - Xa_k) \end{aligned} \quad (3.26)$$

For a given partition $C = \{C_1, \dots, C_k, \dots, C_K\}$ of the set X , the maximum likelihood estimates of the parameters in the model (3.23) can be found by maximizing the log-likelihood function (3.26). This leads to:

$$\hat{\mu}_k = \frac{1}{|C_k|} \sum_{x \in C_k} x_0 \quad (3.27)$$

$$\hat{\sigma}_{0k}^2 = \frac{1}{|C_k|p} \sum_{x \in C_k} (x_0 - \hat{\mu}_k)^T (x_0 - \hat{\mu}_k) \quad (3.28)$$

$$\hat{a}_k = \left(\sum_{x \in C_k} X^T X \right)^{-1} \sum_{x \in C_k} X^T y \quad (3.29)$$

$$\hat{\sigma}_k^2 = \frac{1}{|C_k|(M-p)} \sum_{x \in C_k} (y - X\hat{a}_k)^T (y - X\hat{a}_k) \quad (3.30)$$

for $k = 1, \dots, K$.

Algorithm: This study employs a relocation-iteration algorithm (e.g., k-means), as shown in Figure 3.8, to estimate the model parameters in (3.23) such that the log-likelihood (3.24) is maximized. In 2(a) of Figure 3.8, Θ^t represents the parameters of Equations (3.24) or (3.25) at iteration t , while in 2(b), $(\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^t$ represents the parameters of model k at iteration t .

1. Select randomly an initial partition for the given number of clusters, K ;
2. Iterate ($t = 1, 2, \dots$):
 - (a) Estimate the parameter Θ^t based on the present partition by using Equation (3.27) and (3.28);
 - (b) Generate a new partition by assigning each sequence x to cluster k for which the log-likelihood $\log p(x | (\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^t)$ is maximal;
3. Stop if the improvement of the log-likelihood function (3.24) is below a given threshold, the cluster memberships of time series do not significantly change or a given iteration number is reached.

Figure 3.8 Algorithm for ARM-based clustering

Convergence: The following theorem establishes the convergence of the algorithm above.

Theorem 3.1 The log-likelihood function (3.24) is non-decreasing as the number of iterations increases.

Proof: For the log-likelihood function (3.24), denote the partition after iteration t by C^t and the corresponding parameters by $\Theta^t = \{(\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^t, k = 1, \dots, K\}$. There will be

$$\begin{aligned} \log f(C^t) &= \log p(X | \Theta^t) = \sum_{k=1}^K \sum_{x \in C_k^t} \log p(x | (\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^t) \\ &\leq \sum_{k=1}^K \sum_{x \in C_k^t} \max_{1 \leq l \leq K} \log p(x | (\mu_l, \sigma_{0l}^2, a_l, \sigma_l^2)^t) = \sum_{k=1}^K \sum_{x \in C_k^{t+1}} \log p(x | (\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^t) \\ &\leq \sum_{k=1}^K \sum_{x \in C_k^{t+1}} \log p(x | (\mu_k, \sigma_{0k}^2, a_k, \sigma_k^2)^{t+1}) = \log p(X | \Theta^{t+1}) = \log f(C^{t+1}) \end{aligned}$$

The last inequality above holds because Equations (3.27)-(3.30), which give the maximum likelihood estimates of the parameters in Equation (3.26) for a given partition C^{t+1} . QED.

Note that the above algorithm may converge to a local maximum. A common approach to deal with local maxima is to run the algorithm a number of times and select the best

result among these runs. Such an approach is often used with the k-means and the EM algorithms. It is used for the proposed method here.

3.4.3 Computational Experiments and Results

Four datasets are employed, including SYN, ALP, ELU and BAC described in Section 2.2, to investigate the proposed method. Before cluster analysis, data pre-processing strategies are applied to these four datasets. The expression profile of each gene is first normalized to have a median of 0 and a standard deviation (from the median) of 1. Further, the expression data of all genes at each time point is normalized as so to have a mean of 0 and a standard deviation of 1. Different order ARM-based clusterings may result in different qualities of clusterings. In the following experiments, ARM-based clusterings with different orders ($p = 1,2,3$) (in Figure 3.8) and k-means are applied.

Dataset SYN: ARM-based clusterings with different orders ($p = 1,2,3$) and k-means are applied to dataset SYN over a variety of numbers of clusters. Since the cluster label of each gene in the dataset is known, the ARI between the known cluster labels and the computed cluster labels using both the ARM-based clustering methods and the k-means methods are calculated. AARI over 20 runs are employed to measure the quality of the clustering. The results (in Figure 3.9) show that the quality of clustering using the k-means methods is lower than those using the ARM-based methods with the three different orders. In particular, for the intrinsic number of clusters, $k = 5$, the AARI's of three ARM-based clusterings are 0.76, 0.68 and 0.70 for $p = 1,2,3$, respectively, and are

bigger than 0.62, the AARI using the k-means method. Further, comparing Figure 3.6 (the results of MCM-based clustering for the same data set) to Figure 3.9 shows that the quality of ARM-based clustering with the first order and the second order is also better than that of MCM-based clustering, while the quality of ARM-based clustering with the third order is comparable with that of MCM-based clustering.

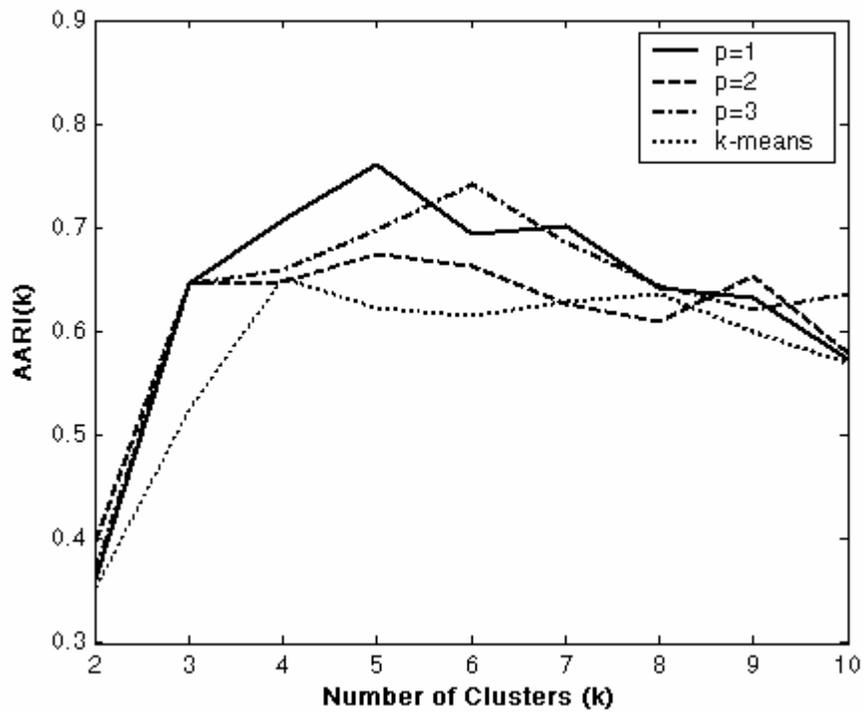


Figure 3.9 Profile of AARI with respect to the number of clusters for dataset SYN

Further comparisons of the ARM-based clustering methods with the k-means methods on datasets ALP, ELU and BAC are attempted. Again for the ARM-based clustering methods, three different orders ($p = 1, 2, 3$) are considered. The number of runs in Figure 3.2 is set to be $B = 20$ in the following experiments.

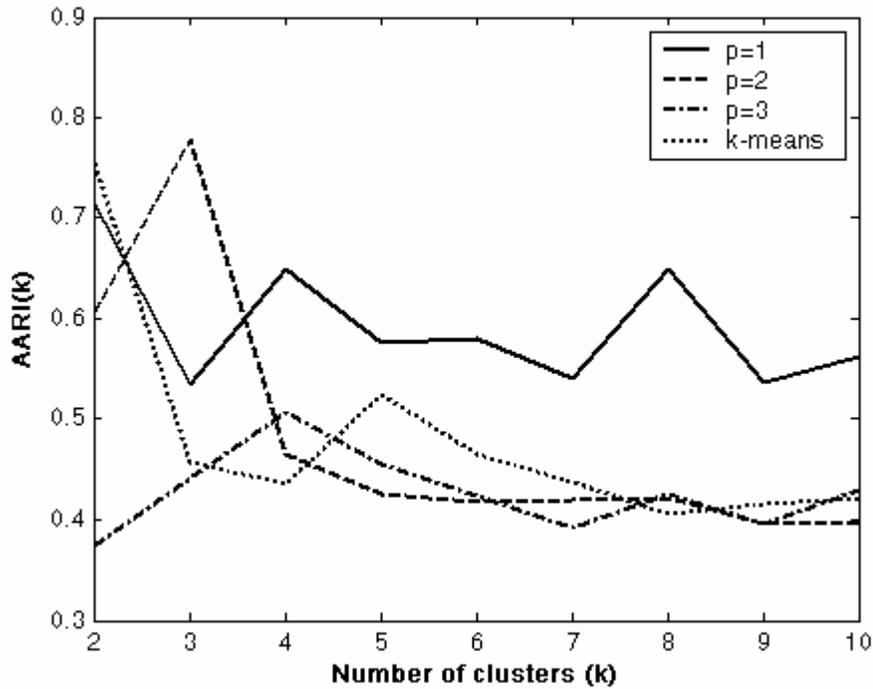


Figure 3.10 Profile of AARI with respect to the number of clusters for dataset ALP

Dataset ALP: Figure 3.10 shows that the AARI's of ARM-based clustering with the first order autoregressive model are the highest among the four methods tested. The results from ARM-based clustering with the second and third order autoregressive models are comparable to those of k-means. This means that with respect to dataset ALP the quality of clustering using the ARM-based method with the first order autoregressive model is the best one, and in particular, better than that using k-means method.

Dataset ELU: Figure 3.11 shows that the AARI's of ARM-based clusterings with the first and second order autoregressive models are always bigger than those using the k-means methods. The result for ARM-based clustering with $p = 3$ is comparable to

those using k-means method. This means that with respect to dataset ELU, the quality of clustering using the ARM-based clustering method with the first and second order autoregressive models is better than that using the k-means method.

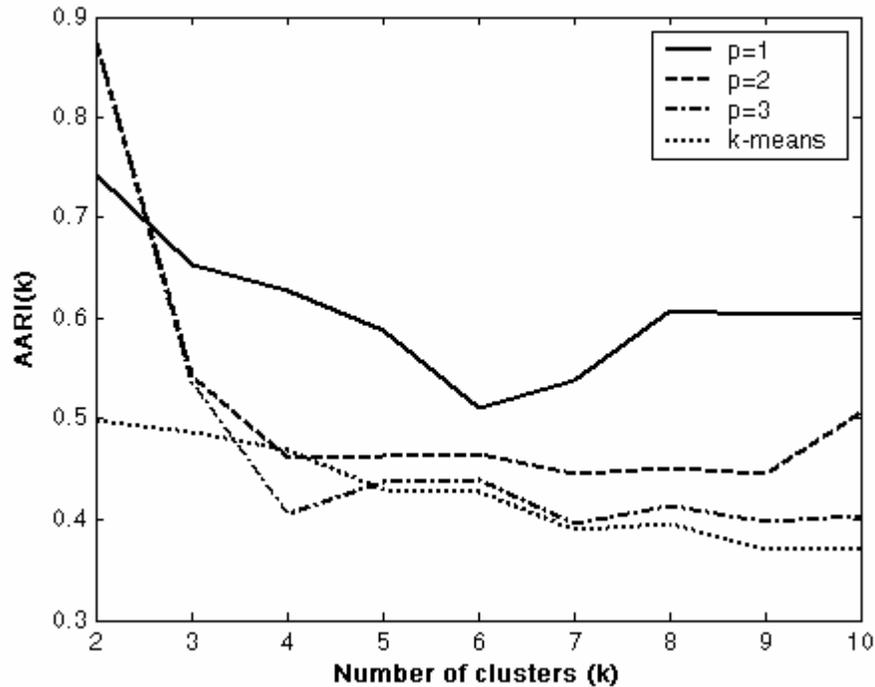


Figure 3.11 Profile of AARI with respect to the number of clusters for dataset ELU

BAC Dataset: Figure 3.12 shows that the AARI's of ARM-based clustering with the three different orders are also higher than those using k-means method over all different numbers of clusters except for $k = 4$ (in the case of $p = 2$ and 3). This means that with respect to dataset BAC, the quality of clustering from ARM-based clustering with the three different orders is better than that using k-means. In addition, comparing Figure 3.7 (the results of MCM-based clustering for the same data set) to Figure 3.12, shows

that the quality of ARM-based clustering with the three different orders is also better than that using the MCM-based clustering method.

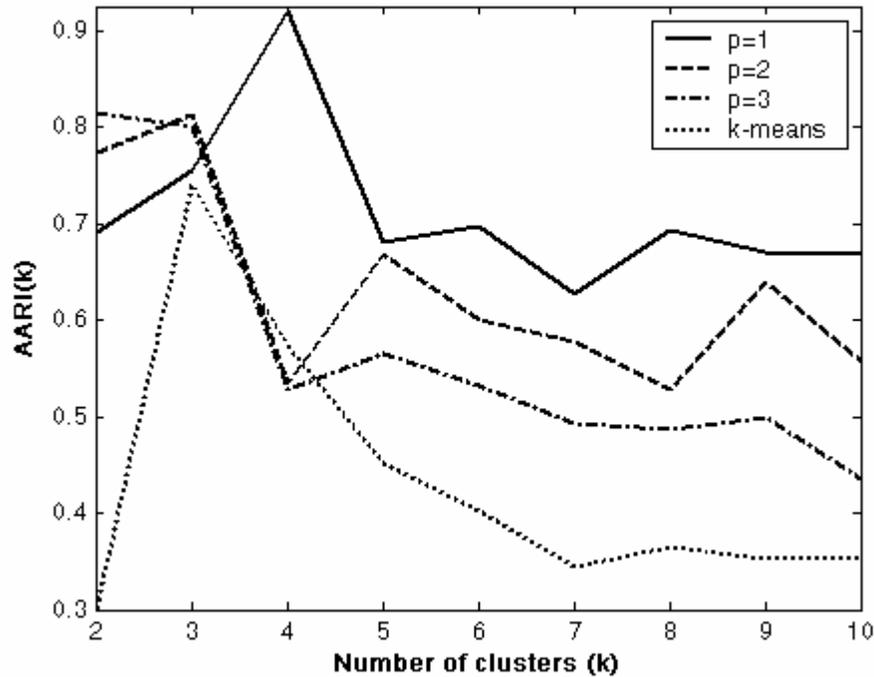


Figure 3.12 Profile of AARI with respect to the number of clusters for dataset BAC

Two general observations are evident from Figure 3.9 through Figure 3.12. First, with increasing the order of autoregressive models, the quality of the ARM-based clustering methods tends to decrease. Thus the best quality with this method is provided by the first-order autoregressive model. One of the possible reasons for this result is that the amount of current gene expression data is insufficient to train higher-order models (Ramoni et al., 2002b). Second, it is always possible to choose an integer p such that the ARM-based clustering methods with the p -th order autoregressive model gives better results than the k-means clustering method. The latter is not surprising given that

the ARM-based clustering method takes into consideration the time dependence of time course gene expression data while k-means methods (a static model-based clustering method) does not.

3.5 Conclusions

Cluster analysis is an important tool to infer gene regulatory relationships. The clustering of time-course gene expression data (where pattern features are time-dependent) differs from the clustering of other kinds of data (where pattern features are independent) and thus is more difficult. The clustering methods in which dynamics of gene expression are not accounted for can not perform as well for time-course gene expression data.

In this chapter, two dynamic model-based clustering methods for time-course gene expression data were proposed. The first one is the MCM-based clustering method in which MCM is employed to account for the dynamic of gene expression. To do that, the whole gene expression dataset is discretized and assigned to one of three gene regulatory states: induction (I), repression (R), and constant (C). Although the discretization may result in information loss, gene expression state sequences retain information about the dynamics of gene expression. The results of computational experiments on two datasets show that the MCM-based clustering outperforms k-means. The second method is the ARM-based clustering method in which ARM is employed to account for the dynamics of gene expression. The results of computational experiments

on four datasets show that the ARM-based clustering outperforms not only k-means but also MCM-based clustering.

In conclusion, since the dynamics of gene expression are accounted for, the proposed dynamic model-based clustering methods for time-course gene expression data are able to improve the quality of the clustering. The most important feature of the proposed methods is that they take the inherent time dependence (dynamics) of time-course gene expression patterns into consideration. In addition, the proposed methods are flexible in the sense that they can incorporate a priori information into the models as they have a solid probabilistic foundation.

Chapter 4

GENE REGULATORY NETWORK

4.1 Related Work

A gene regulatory network is a dynamic system to describe interactions among a large number of different substances (such as mRNA, proteins) in a living cell. The understanding and unravelling of such cellular systems has been proven useful in genomic disease diagnosis and genomic drug design. Recently the advent of microarray technology and other high throughput gene expression measurement technologies have provided the opportunity to model gene regulatory networks with large-scale gene expression data. Since then, a wide variety of different models have been proposed for genetic regulatory networks.

This section gives a survey of computational models for large gene regulatory networks and discusses their advantages and disadvantages. The survey is not meant to be on all models, and instead focusing on Boolean network models and differential/difference models. Several reviews on modelling gene regulatory networks have been published (De Jong, 2002; Wessels et al., 2001; D'haeseleer et al., 2000; and their references).

Besides the emphasis on large gene networks, this section reviews the existing models to focus on mathematical methods, computational cost, evaluating their relative advantages and disadvantages, and the biological concepts on which models based, rather on the biological results obtained through their applications.

4.1.1 Boolean Network Models

One of the earliest models for large gene regulatory networks is Boolean network model, where a gene can be in one of two states, either active or inactive, described as completely “on” or “off”. These two states are often represented by the binary values 1 and 0, respectively. The binary state varies with respect to time and depends on the states of other genes in the network through a Boolean variable equation:

$$x_i(t+1) = F_i[x_1(t), \dots, x_n(t)], \quad i = 1, \dots, n \quad (4.1)$$

where $x_i(t)$ ($i = 1, \dots, n$) stands for the state of the i -th element (genes or proteins) in the network, n is the number of genes in the network, and the function F_i ($i = 1, \dots, n$) is a Boolean function in the states of the network at time t for updating the state of element i at $t+1$. For example, if $x_1(t)$ is ‘on’ AND either $x_2(t)$ OR $x_3(t)$ is ‘off’ at time t , then $x_4(t)$ is ‘on’ at time $t+1$. In this case, Equation (4.1) can be written as: $x_4(t+1) = x_1(t) \wedge (\neg x_2(t) \vee \neg x_3(t))$, where \wedge , \vee , and \neg are standard logic operation symbols. Let $\mathbf{x}(t) = [x_1(t) \ \dots \ x_n(t)]^T$ denote the vector state and $\mathbf{F} = [F_1 \ \dots \ F_n]^T$

denote the vector-valued function of the system where the superscript “T” stands for the transposition of a vector. Equations (4.1) can be rewritten in the concise form as follows:

$$\mathbf{x}(t+1) = \mathbf{F}(\mathbf{x}(t)) \quad (4.2)$$

The most important problem with the identification of Boolean networks is to determine the connectivity degree of genes in the networks. The term connectivity degree refers to the number of input variables which appear in the function F_i ($i = 1, \dots, n$) in the right side of Equation (4.1). If it is simply thought that one gene is regulated by all n genes in the network, there are 2^{2^n} Boolean functions that have to be checked to determine a regulatory relationship as in Equation (4.1) between this gene and other genes. This is infeasible in practice even if n is moderate. For example, when $n = 10$, $2^{2^n} \approx 2^{1000} \approx 10^{300}$. Indeed, some biological knowledge shows that one gene does not need to be regulated by all n genes in the network. That is, the function F_i ($i = 1, \dots, n$) in Equation (4.1) may depend on h ($\ll n$) genes only (Baldi and Hatfield, 2002).

Unfortunately, there has been no objective approach available to determine the connectivity degree of genes in a Boolean network. In the recent studies on Boolean network models for gene regulatory networks (Akutsu et al., 1999; Wuensche 1998; Liang et al., 1998; Akutsu et al., 2000), there tends to assume that the connectivity degrees of genes are smaller than a constant h (typically less than 3). Under this assumption, Liang et al. (1998) described an algorithm for inferring gene network

architectures from the rule table of a Boolean network model. Their computational experiments and theoretical analysis have shown that a small number of state transition pairs are sufficient to infer the original observations. Furthermore, Akutsu et al. (1999) devised a much simpler algorithm for the same problem and proved that only $O(\log n)$ state transition pairs (from 2^n pairs) are necessary and sufficient to identify the original Boolean network of n nodes (genes) correctly with high probability. More exactly, this number is $O(2^{2^h} (2h + \alpha) \log n)$ where α is a positive constant (Wu, 2003). Their algorithms were claimed to have time complexity $O(mn^{h+1})$ where m is the number of examples (Akutsu et al., 1999 and 2000). More precisely, this number is $O(h \cdot 2^{2^h} \cdot n^{h+1} \cdot m)$. Therefore when $h = 2$ or 3 , the authors' claims about algorithmic complexity or the number transition pairs are acceptable. However, when $h = 10$, for example, their claims do not make sense because the symbol "big O " hides a very large coefficient ($2^{2^{10}} \approx 10^{300}$) in $O(mn^{h+1})$.

Somogyi and Sniegowski (1996) showed that such Boolean networks have features similar to those in the biological systems, such as global complex behaviour, self-organization, stability, redundancy, and periodicity. However, the Boolean network models have several disadvantages. For example, they treat gene expression as either completely "on" or "off", and thus ignore those genes that have a range of expression levels and can have regulatory effects at intermediate expression levels. Furthermore, they do not address those regulatory genes that influence the transcription of various genes to differing degrees. Finally, such networks are designed such that all genes have

a fixed maximum connectivity degree. In biology, some genes are known to have many regulatory inputs, while others are not known to have more than a few (Weaver et al., 1999).

4.1.2 Differential/difference Equation Models

An alternative to the Boolean network models (discrete variable) is a continuous dynamic model, where the state variables theoretically have range $[-\infty, \infty]$ rather than $\{0,1\}$. A continuous dynamic model may be described by a system of differential equations in the generic form as follows:

$$\frac{dx_i}{dt} = F_i[x_1(t), \dots, x_n(t), I(t)], \quad i = 1, \dots, n \quad (4.3)$$

or by a system of difference equations in the generic form as follows:

$$x_i(t + \Delta t) = F_i[x_1(t), \dots, x_n(t), I(t)], \quad i = 1, \dots, n \quad (4.4)$$

where $x_i(t)$ ($i = 1, \dots, n$) stands for the state of the i -th element (genes or proteins) in the network, n is the number of genes in the network, the vector $I(t)$ represents some external inputs to the system, and F_i ($i = 1, \dots, n$) is a multivariable nonlinear function. It is impossible to identify the nonlinear systems (4.3) and (4.4) because of the limitations of data obtained from microarray experiments and of identification

techniques for nonlinear systems. Therefore it often assumed that F_i ($i = 1, \dots, n$) is a multivariable linear function. Such an assumption has two advantages. Firstly, it is mathematically simple. Secondly, a linear system may be a satisfying approximation of a nonlinear system in a certain neighbour of a working state.

Chen et al. (1999) proposed a theoretical model for gene regulatory networks described by the following linear differential equations:

$$\frac{d}{dt} \mathbf{x}(t) = \Lambda \cdot \mathbf{x}(t) \quad (4.5)$$

where Λ is a constant matrix and represents the extent or degree of regulatory relationships among genes and/or proteins, the vector $\mathbf{x}(t) = [x_1(t) \ \dots \ x_n(t)]^T$ contains the mRNA and/or protein concentrations as a function of time t with $x_i(t)$ ($i = 1, \dots, n$) standing for the state of the i -th element (genes or proteins), and n is the number of genes and/or proteins in the model.

D'haeseleer et al. (1999) proposed the following linear difference equation model for gene regulatory networks:

$$\mathbf{x}(t + \Delta t) = \mathbf{W} \cdot \mathbf{x}(t) \quad (4.6)$$

where the constant matrix $\mathbf{W} = [w_{ij}]_{n \times n}$ represents regulatory relationships and degrees among genes, $x_i(t + \Delta t)$ is the expression level of gene i at time $t + \Delta t$, and w_{ij} indicates how much the level of gene j influences gene i when time goes from t to $t + \Delta t$. Furthermore, an extra term indicating the influence of kainate and two bias terms are added to Equation (4.6), and the final equation becomes

$$\mathbf{x}(t + \Delta t) = \mathbf{W} \cdot \mathbf{x}(t) + K \cdot \text{kainate}(t) + C + T \quad (4.7)$$

where $\text{kainate}(t)$ is the kainate level at time t , and K , C , and T are three n -dimensional vectors, where the i -th components of K , C , and T are the influence of kainate on gene i , a constant bias factor for gene i , and the difference in bias between tissue types, respectively (D'haeseleer et al., 1999).

Models (4.5) and (4.6) are equivalent. When Δt tends to zero, model (4.6) may be transformed into model (4.5). On the other hand, to identify the parameters in model (4.5), one must discretize it into the formalism of model (4.6) (Chen et al., 1999). Since gene expression data from the DNA microarrays can only be obtained at a series of discrete time points with the current experimental technologies, the difference equations are more suitable to model gene expression data. Due to the lack of gene expression data, models (4.5) and (4.6) are usually underdetermined. Similar to Boolean network models (Akutsu et al., 1999; Wuensche 1998; Liang et al., 1998; Akutsu et al., 2000), the assumption that the connectivity degree of all genes in the network is smaller than a fixed constant h is also used to make models (4.5 and (4.6) identifiable. Chen et al.

(1999) showed that model (4.5) can be constructed in $O(n^{h+1})$ time. It is clear that the models constructed as such contradict the fact that some genes are known to have many regulatory inputs, while others are not known to have more than a few as the Boolean networks (Liang et al., 1998; Akutsu et al., 1999).

Furthermore, the fixed maximum connectivity degree h of Chen et al. (1999) is chosen in an *ad hoc* manner. De Hoon et al. (2003) considered Chen's differential model and used Akaike's Information Criterion (AIC) to determine the connectivity degree h for each gene. In their method, not all genes must have a fixed connectivity. However, they do not present an efficient algorithm to identify the parameters of their differential equation model; the brute-force algorithm used in their paper (De Hoon et al., 2003) has a computational complexity of $O(2^{n^2})$, where n is the number of genes in the model. The authors claimed that their method could be applied to find a regulatory network among individual genes. However, for biologically realistic regularity networks, the computational complexity is prohibitive. Actually, De Hoon et al. (2003) did not build any gene expression models among individual genes, and instead chose to group the genes into several clusters and only studied the interrelationships among the clusters.

D'haeseleer et al. (1999) applied the linear difference equation model (4.7) to mRNA expression data during CNS (Central Nervous System) development and injury. The dataset includes $65 \times 28 = 1820$ gene expression values for 65 genes and 28 expression values for each gene. However, there are $65 \times 68 = 4420$ parameters in model (4.7). To cope with the lack of gene expression data, D'haeseleer et al. (1999) used a nonlinear

interpolation scheme to guess the shapes of gene expression profiles between the measured time points. Such an interpolation scheme is *ad hoc*. Therefore, the soundness of the model built from such interpolated data is suspicious. In addition, while they built a linear model for 65 measured mRNA species, there exists a problem of dimensional disaster even when the number of genes in a model is moderate, for example, around 1000 (typically, the number of genes in a gene regulatory network under considerations in this study).

4.2 Evaluations

Due to limitations of the understanding of real gene regulatory networks, it is difficult (if not impossible) to evaluate the models for gene regulatory networks completely by biological experiments. Wesseles et al. (2001) proposed six indices to evaluate the models for gene regulatory networks from the viewpoint of bioinformatics. Some of these indices are inapplicable to evaluation of gene regulatory network models on real-life gene expression datasets because the real gene regulatory networks creating these data are unknown. In the following, five indices are introduced, including the computational cost, the prediction power, and the stability (Wesseles et al., 2001), the robustness, and the periodicity (Kauffman, 1993).

The *computational cost*: The demand on computational resource is always a concern in the analysis of gene expression data given a large-scale dataset. The computational complexity to build gene regulatory networks is used for evaluating the models.

The *stability*: Due to limited energy and storage within a living cell, concentrations of gene expression products such as mRNA should remain bounded. All real gene regulatory networks are therefore stable. Therefore, inferred gene regulatory networks should also be (almost) stable in order to be realistic. The dynamics part of all proposed models in this study may be written in a unified equation:

$$\mathbf{z}(t+1) = \sum_{\tau=0}^{\tau_{\max}} \mathbf{A}_{\tau} \cdot \mathbf{z}(t-\tau) \quad (4.8)$$

where $\mathbf{z}(t)=[z_1(t) \ \dots \ z_p(t)]^T$ is the state vector, $\mathbf{A}_{\tau}=[a_{ij\tau}]_{p \times p}$ ($\tau=0, \dots, \tau_{\max}$) are the state transition matrices with time delay τ , and the integer parameter τ_{\max} denotes the maximum time delay accounted for. As such, the stability of inferred gene regulatory networks is equivalent to the stability of Equation (4.8). It can be proven that Equation (4.8) is stable if and only if all eigenvalues of the following $(\tau_{\max} + 1) \times (\tau_{\max} + 1)$ block matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{0}_p & \mathbf{I}_p & \dots & \mathbf{0}_p \\ \vdots & \vdots & \ddots & \mathbf{0}_p \\ \mathbf{0}_p & \mathbf{0}_p & \dots & \mathbf{I}_p \\ \mathbf{A}_{\tau_{\max}} & \mathbf{A}_{\tau_{\max}-1} & \dots & \mathbf{A}_0 \end{bmatrix} \quad (4.9)$$

lie inside the unit circle in the complex plane, where \mathbf{I}_p is a $p \times p$ identity matrix and $\mathbf{0}_p$ is a $p \times p$ zero matrix, and \mathbf{A}_{τ} ($\tau=0, \dots, \tau_{\max}$) are state transition matrices with time delay τ in Equation (4.8).

The *periodicity*: Certain biological processes are periodic. The cell-cycle and circadian clock, for example, repeat at well-defined and reliable intervals. Studies have shown that gene regulatory networks associated with these periodic biological processes are themselves rhythmic (Kauffman, 1993; Baldi and Hatfield, 2002). Therefore, the inferred gene regulatory networks associated with these periodic biological processes should be periodic over their stable states. Accordingly, the periodicity of system (4.8) at its stable state is determined by its dominant eigenvalues (the eigenvalues of matrix T in (4.9) whose modulus is the largest).

The *robustness*: The robustness of a gene regulatory network is understood as its insensitivity to noises or disturbances. It is known that a real gene regulatory network has robustness (Kauffman, 1993). Therefore, the inferred gene regulatory network should be robust. In general, the stability of a linear system implies some of its robustness (Chen, 1999). Note that the stability, the robustness, and the periodicity of the system (4.8) are all related to the eigenvalues of matrix T in (4.9).

The *prediction power (error)*: Let $\hat{\mathbf{X}}$ be a matrix with the same size as the original data matrix \mathbf{X} , which is computed from an initial state and the model derived from the data matrix \mathbf{X} . The prediction error reflects how well $\hat{\mathbf{X}}$ approximates \mathbf{X} . The prediction error (P_E) may be defined as:

$$P_E = \frac{1}{n} \sum_{i=1}^n \frac{\|\mathbf{X}(i,:) - \hat{\mathbf{X}}(i,:)\|^2}{\|\mathbf{X}(i,:)\|^2} \quad (4.10)$$

where $\mathbf{X}(i,:)$ is the i -th row vector of gene expression data matrix \mathbf{X} (i.e., the expression profile of the i -th gene). $\|\mathbf{X}(i,:)\|$ is the Euclidean norm of the vector $\mathbf{X}(i,:)$. Intuitively, the smaller the prediction error, the greater the prediction power. Wesseles et al. (2001) defined the prediction power as:

$$P_p = 1/(1 + E_{MSE}) \quad (4.11)$$

where $E_{MSE} = \frac{1}{nm} \sum_{i=1}^n \|\mathbf{X}(i,:) - \hat{\mathbf{X}}(i,:)\|^2$. Obviously, the scale of \mathbf{X} 's elements influences the value of E_{MSE} and further influences the value of P_p . For example, one may always multiply by a small constant to decrease E_{MSE} and thus increase P_p while the model is actually not improved. On the other hand, P_E in Equation (4.10) is invariant to the scale of \mathbf{X} . Therefore, it is more reasonable for evaluation of the models using P_E in Equation (4.10) than using P_p in Equation (4.11).

4.3 State-Space Model

The state-space model is one of the most powerful methods to describe a dynamic system and has been widely employed for engineering control systems (Chen, 1999). A state-space model consists of the internal variables, external (input) variables, and

observation (output) variables. Figure 4.1 shows a typical state-space model of gene regulatory networks. Typically the observation variables depend on the internal variables while the change of the internal variables is completely determined by the current internal variables plus any external inputs, if they exist.

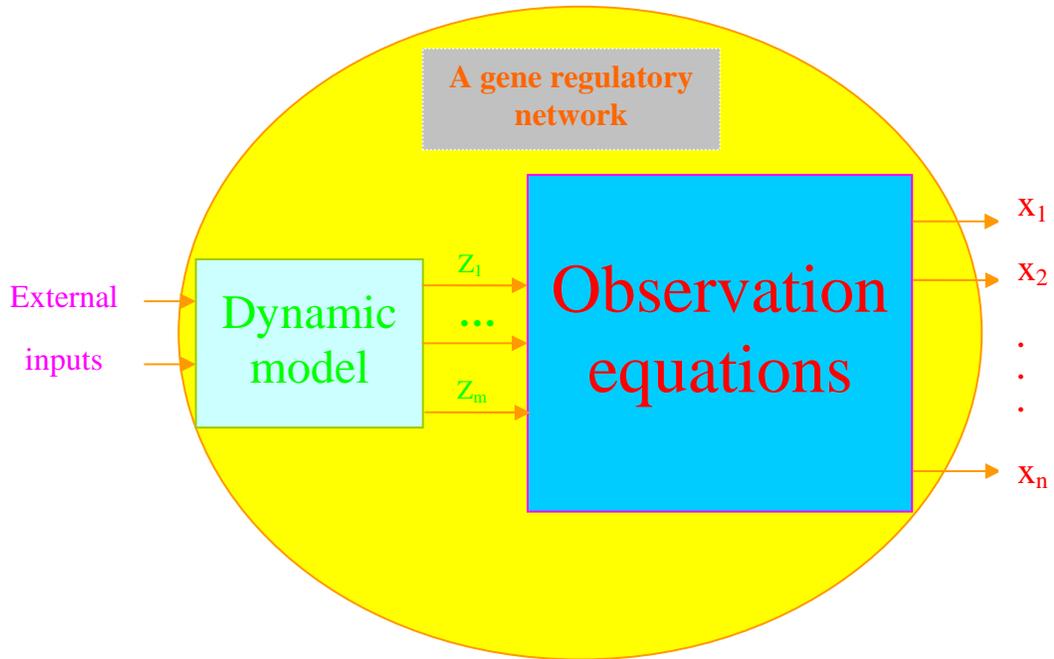


Figure 4.1 A state-space model for a gene regulatory network, where x_i ($i = 1, \dots, n$) is an observation variable while z_i ($i = 1, \dots, p$) is an state variable.

In fact, the Boolean network models and the differential/difference models (Equations 4.1- 4.7) are variations of the state-space model. However, in these models, genes were viewed as the internal state variables as well as observation variables of a cellular system and their expression levels were the values of both the internal state variables and the observation values. This viewpoint has led to an underestimation of model

parameters, as pointed out previously. In addition, these models assume that regulatory relationships among genes are “direct”; for example, gene j directly regulates gene i with the weight w_{ij} in model (4.6). In fact, genes may not be regulated in such a direct way in a cell, and instead they may be regulated by some internal regulatory elements (Spellman et al., 1998; Zhang, 1999; Baldi and Hatfield, 2002).

This section proposes a state-space model for gene regulatory networks, in which genes are viewed as the observation variables and their expression levels are observation values and gene expression dynamics are governed by the internal variables with their linear combinations. The number of internal variables and the expression profiles of internal variables will be determined by Bayesian information criterion (BIC) and factor analysis (FA) from the observation values of a cellular system, i.e., gene expression data.

4.3.1 The Model

The state-space model for gene regulatory network (Figure 4.1) may be mathematically described as follows (Wu et al., 2004a):

$$\begin{cases} \mathbf{z}(t + \Delta t) = \mathbf{A} \cdot \mathbf{z}(t) + \mathbf{n}_1(t) \\ \mathbf{x}(t) = \mathbf{C} \cdot \mathbf{z}(t) + \mathbf{n}_2(t) \end{cases} \quad (4.12)$$

In terms of the linear system theory (Chen, 1999), Equation (4.12) are called the state-space description of a system. The vector $\mathbf{x}(t)=[x_1(t) \ \cdots \ x_n(t)]^T$ consists of the observation variables of the system (4.12), and $x_i(t)$ ($i=1, \dots, n$) represents the expression level of gene i at time t , where n is the number of genes in the model. The vector $\mathbf{z}(t)=[z_1(t) \ \cdots \ z_p(t)]^T$ consists of the internal state variables of the system (4.12) and $z_i(t)$ ($i=1, \dots, p$) represents the expression value of internal element i at time t which directly regulates gene expression, where p is the number of the internal state variables. $\mathbf{A}=[a_{ij}]_{p \times p}$ is a time translation matrix of the internal state variables, called the state transition matrix, which provides key information on the influences of the internal variables on each other. $\mathbf{C}=[c_{ik}]_{n \times p}$ is a transformation matrix between the observation variables and the internal state variables, which provide key information on the influences of the internal regulatory elements on the genes. Finally, the vectors $\mathbf{n}_1(t)$ and $\mathbf{n}_2(t)$ stand for the system noises and the observation noises, respectively.

Comparing to previous models (4.5-4.7), the state-space model (4.12) has the following characteristics. First, genes are the observation variables rather than the internal state variables. Second, from a biological angle, the model (4.12) can capture the fact that genes may be regulated by other internal regulatory elements (Alberts et al., 1998; Zhang, 1999; Baldi and Hatfield, 2002). Finally, although it contains two equations (one is a group of difference equations, and the other is a group of algebraic equations), the parameters in model (4.12) are identifiable from the current volume of gene expression datasets without any objective assumptions on the connectivity degrees of genes (Liang

et al., 1998; Akutsu et al., 1999; Chen et al., 1999; Akutsu et al., 2000) and the computational complexity to identify them is simple (see the next section).

4.3.2 Model Identification

The task of parameter identification in model (4.12) is to estimate the elements of matrices $\mathbf{A} = [a_{ij}]_{p \times p}$ and $\mathbf{C} = [c_{ik}]_{n \times p}$ such that both the system error and the observation error are minimized with certain senses. Let \mathbf{X} be the gene expression data matrix with n rows and m columns, where n and m are the numbers of the genes and the measuring time points, respectively. The building of model (4.12) from microarray gene expression data \mathbf{X} can be divided into two phases. Phase one identifies the internal state variables and their expression matrix \mathbf{Z} from the data matrix \mathbf{X} and computes the transformation matrix \mathbf{C} such that

$$\mathbf{X} = \mathbf{C} \cdot \mathbf{Z}. \quad (4.13)$$

Phase two builds the dynamic equations of the internal states; i.e., determine the state transition matrix \mathbf{A} from the expression matrix \mathbf{Z} . Phase one minimizes the observation error (i.e., maximize the data likelihood) with BIC, while Phase two minimizes the system error.

In the process of building model (4.12), Phase one, i.e., establishing Equation (4.13), is key. There are many methods that may be used to get decomposition (4.13) of gene

expression data \mathbf{X} . For example, one may employ the singular value decomposition (Alter et al., 2000; Holter et al., 2001), where some of the so-called characteristic modes or eigengenes may be viewed as the internal variables. However, the number of such internal variables is chosen *ad hoc* in those studies since the matrix \mathbf{C} and the expression data matrix of the internal variables \mathbf{Z} are decided by subjectivity rather than by the data themselves. Note that the matrices \mathbf{C} and \mathbf{Z} are dependent. After \mathbf{Z} is identified, \mathbf{C} may be calculated by formulae $\mathbf{C} = \mathbf{X} \cdot \mathbf{Z}^+$, where \mathbf{Z}^+ is a unique Moore-Penrose generalized inverse of the matrix \mathbf{Z} .

This study employs maximum likelihood factor analysis (MLFA) (Lawley and Maxwell, 1971; Bubin and Thayer, 1982; Everitt and Dunn, 1992) to identify the internal state variables and employs Bayesian Information Criterion (BIC) (Schwarz, 1978) to determine the number of the internal state variables, where \mathbf{X} is the $n \times m$ observed data matrix, \mathbf{C} is the $n \times p$ unobserved factor-score matrix, and \mathbf{Z} is the $p \times m$ loaded matrix. In fact, both the generalized likelihood ratio test (GLRT) and the Akaike's Information Criterion (AIC) methods (Burnham and Anderson, 1998) also may be used to determine the number of the internal variables, but they have a similar drawback; as the sample size increases there is an increasing tendency to accept the more complex model (Raftery, 1986). The BIC takes the sample size into account, and thus avoids the over-fitting of a model to data. Although the BIC method was developed from a Bayesian standpoint, the result is insensitive to the prior distribution for the adequate sample size. Thus a prior distribution does not need to be specified (Schwarz,

1978; Raftery, 1986), which simplifies the method. For each model, the BIC is defined as:

$$BIC = -2 \cdot \left[\begin{array}{l} \log - \text{likelihood of the} \\ \text{estimation model} \end{array} \right] + \log(n) \cdot \left[\begin{array}{l} \text{number of the estimated} \\ \text{parameters in the model} \end{array} \right] \quad (4.14)$$

where n is the sample size (the number of genes in this case). Accordingly, the model is chosen with the smallest BIC based on the above definition of BIC.

After obtaining the expression data matrix of the internal variables \mathbf{Z} and the transformation matrix \mathbf{C} in Phase one, the dynamic equations describing the state transition in model (4.12) can be developed; i.e.,

$$\mathbf{z}(t + \Delta t) = \mathbf{A} \cdot \mathbf{z}(t) \quad (4.15)$$

from the data matrix \mathbf{Z} in Phase two. The matrix \mathbf{A} contains p^2 unknown elements while the matrix \mathbf{Z} contains $m \times p$ known expression data points. If $p > m$, Equation (4.15) will be underdetermined. Fortunately, using BIC the number of chosen internal variables p generally is less than the number of time points m . Therefore, all elements of matrix \mathbf{A} can be unambiguously identifiable.

To determine the elements of matrix \mathbf{A} , the time step Δt is chosen to be the highest common factor among all of the experimentally measured time intervals such that the

time of the j th measurement is $t_j = n_j \cdot \Delta t$, where n_j is an integer. For equally spaced measurements, $n_j = j$. Define a time-variant vector $\mathbf{v}(t)$ with the same dimensions as the internal state vector $\mathbf{z}(t)$ and with the initial value $\mathbf{v}(t_0) = \mathbf{z}(t_0)$. For all subsequent time points, $\mathbf{v}(t)$ is determined from $\mathbf{v}(t + \Delta t) = \mathbf{A} \cdot \mathbf{v}(t)$. For any integer k , there is

$$\mathbf{v}(t_0 + k \cdot \Delta t) = \mathbf{A}^k \cdot \mathbf{v}(t_0) \quad (4.16)$$

The p^2 unknown elements of the matrix \mathbf{A} are chosen to minimize the cost function (the sum of squared relative errors)

$$CF = \sum_{j=1}^m \|\mathbf{z}(t_j) - \mathbf{v}(t_j)\|^2 / \sum_{j=1}^m \|\mathbf{z}(t_j)\|^2 \quad (4.17)$$

where $\|\bullet\|$ stands for the Euclidean norm of a vector. For equally-spaced measurements, the problem is a linear regression one, and the solution to minimizing the cost function (4.17) can be a least square one. Compared to Equation (4.10), Equation (4.17) is a variation of the prediction power defined in Equation (4.10) for the internal variables. For unequally-spaced measurements, the problem becomes nonlinear, and it is necessary to employ an optimization technique such as those described in Chapter 10 of the book (Press et al., 1992) to determine matrix \mathbf{A} .

The analysis of computational complexity: In Phase one, MLFA and BIC were employed to establish the observation equations and to estimate the internal variables.

The computational complexity is linear in the numbers of genes, n , time points, m , and iterations of MLFA, R . In Phase two, a multiple regression method was employed to establish the state transition Equation (4.15). The computational complexity is linear in the number of time points, and at most cubic in the number of internal variables. Overall, the complexity of the state-space model identification is linear in the numbers of genes and iterations (i.e., $O(n * R)$) as the numbers of both time points and internal variables are much smaller than the number of genes and can be viewed as constants.

4.3.3 Computational Experiments and Results

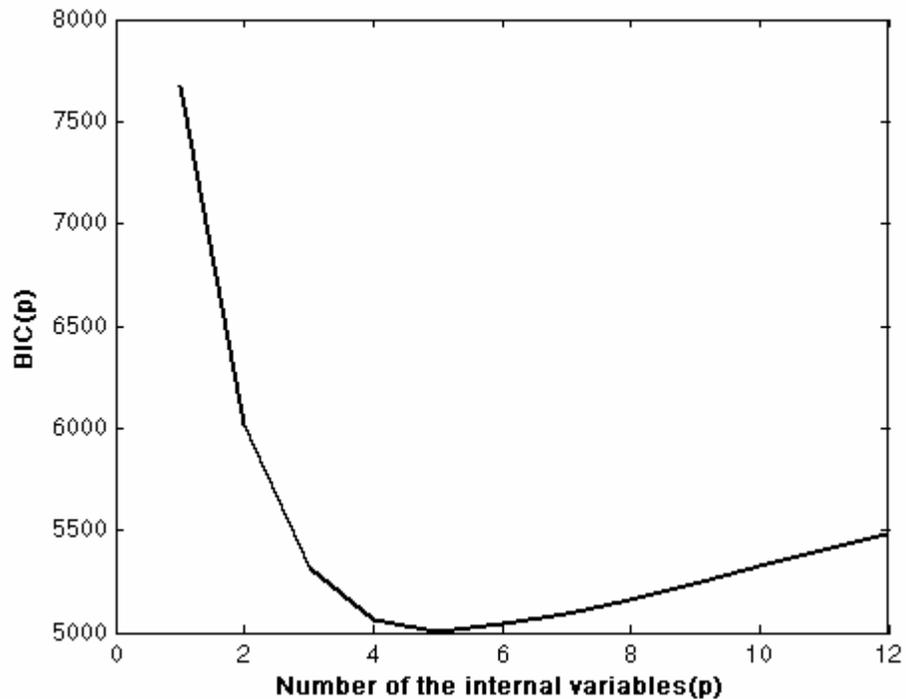


Figure 4.2 Profiles of BIC with respect of the number of the internal variables for dataset CDC15

Two datasets, CDC15 and BAC as described in Section 2.2, are used for investigating the proposed model in this section. The expression profile of each gene is normalized to have a length of one and then for expression values on each microarray as so to have a mean of zero and a length of one. Such normalizations make MLFA simple (Lawley and Maxwell, 1971).

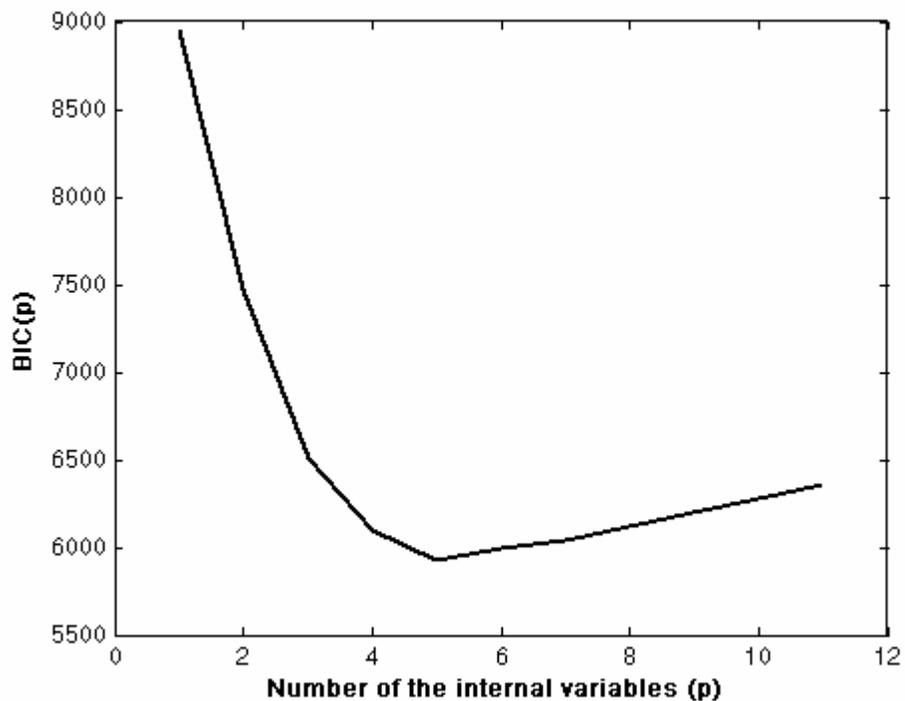


Figure 4.3 Profiles of BIC with respect of the number of the internal variables for dataset BAC

The EM algorithm for MLFA (Bubin and Thayer, 1982) was employed for the two datasets, respectively. Note that each gene expression profile corresponds to one observation. The total number of parameters to be identified is $p \cdot m$ (elements of the

matrix \mathbf{Z}) plus m (the variances of residue errors) (Bubin and Thayer, 1982). Figures 4.2 and 4.3 depict the profiles of BIC with respect to the number of internal variables for datasets CDC15 and BAC, respectively. Clearly, from Figures 4.2 and 4.3, the best choice is 5 as the number of internal variables for both datasets according to the BIC.

Table 4.1 The internal variable expression matrices for datasets CDC15 and BAC

CDC15	-0.2065 0.2914 -0.5766 0.2401 -0.0886 -0.7472 0.0812 -0.4848 0.1591 -0.0418 -0.5397 -0.6201 -0.2144 0.1406 -0.0389 0.2695 -0.7875 -0.0898 0.0950 0.1159 0.7960 -0.3190 -0.2828 -0.0038 0.1283 0.6692 0.4116 -0.3365 -0.0460 0.1430 -0.4139 0.4091 -0.3770 -0.4557 -0.0130 -0.7042 -0.2534 -0.0028 -0.4060 0.0820 -0.3371 -0.6247 0.0893 -0.1332 -0.0618 0.5592 -0.4646 -0.1469 -0.0957 -0.3433 0.7490 0.0429 -0.1504 -0.1983 -0.2431 0.0216 0.5261 0.2677 0.2599 -0.1465
BAC	-0.4478 0.0733 -0.5429 0.0938 -0.1839 -0.6954 0.2965 -0.4481 0.0018 -0.2020 -0.8355 0.4048 0.0408 -0.2612 0.0739 -0.7904 0.2241 0.1674 0.0162 0.0252 -0.7850 0.2158 0.2685 0.0289 0.0021 -0.8141 -0.0381 0.2671 0.2602 -0.1303 -0.7410 -0.4120 0.1512 0.0618 -0.0864 -0.6371 -0.5639 0.0442 -0.2583 -0.1583 -0.5635 -0.4091 -0.1484 -0.2821 0.0947 -0.7409 -0.2597 -0.2584 0.1761 0.3170 -0.7777 -0.0906 -0.1943 0.1666 0.1007

The expression matrices for the five internal variables are, respectively, listed in Table 4.1, where for each expression matrix each column describes one internal variable. In order to determine the state transition matrices in the models from the internal expression matrices, two optimization problems in Equation (4.17) for the two datasets need to be worked out, respectively. As both datasets are equally-spaced measurements, the least square method was used to obtain the two state transition matrices A in the models, as shown in Table 4.2.

Table 4.2 The state transition matrix of internal variables for datasets CDC15 and BAC

CDC15	$A = \begin{bmatrix} 0.4378 & -1.0077 & 0.5009 & 0.1851 & -0.1189 \\ 0.6649 & 0.5244 & 0.2475 & 0.1511 & -0.1356 \\ -0.0702 & 0.1734 & 0.6794 & -0.3092 & -0.5279 \\ -0.0699 & -0.0103 & 0.1786 & 0.6163 & -0.5190 \\ 0.0161 & 0.0316 & -0.0700 & 0.1358 & 0.6662 \end{bmatrix}$
BAC	$A = \begin{bmatrix} 0.0211 & -0.0455 & 0.3359 & 0.0384 & -0.0039 \\ 0.0742 & 0.8128 & -0.5702 & -0.2412 & 0.1636 \\ -0.0577 & 0.3790 & 0.6329 & 0.0228 & -0.0173 \\ -0.0202 & 0.1434 & 0.0700 & 0.1472 & 0.7819 \\ 0.0013 & -0.0953 & -0.1146 & -0.4443 & 0.3996 \end{bmatrix}$

To evaluate the predication power of the inferred gene regulatory networks, Figures 4.4 and 4.5 give comparisons of the internal variable expression profiles in Table 4.1 and their calculated profiles from Equation (4.15) for datasets CDC15 and BAC, respectively, where the solid lines stand for the profiles in Table 4.1 and the dash lines

for the calculated profiles from Equation (4.15). The values of the cost functions in Equation (4.17) are 0.2321 and 0.0761 for the CDC15 dataset and the BAC dataset, respectively. Therefore, two state transition matrices in Table 4.2 are plausible.

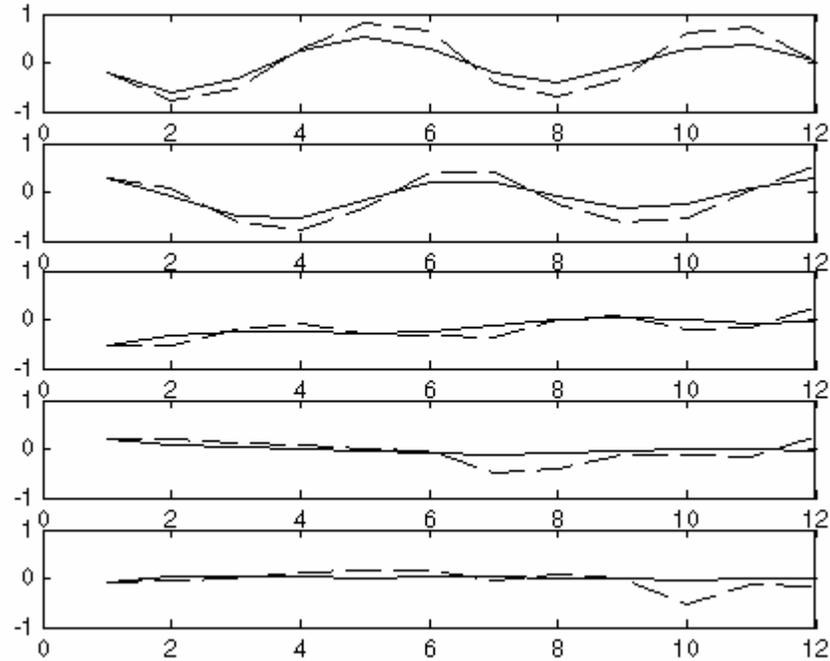


Figure 4.4 A comparison of the internal variable expression profiles in Table 4.1 and their calculated profiles from Equation (4.15) for dataset CDC15

To inspect the stability, the robustness, and the periodicity of inferred gene regulatory networks, the eigenvalues of matrix \mathbf{T} in Equation (4.9) need to be solved. For the state-space model without time delay (the current case), matrix \mathbf{T} in Equation (4.9) is equal to matrix \mathbf{A} in Equation (4.12) or (4.15).

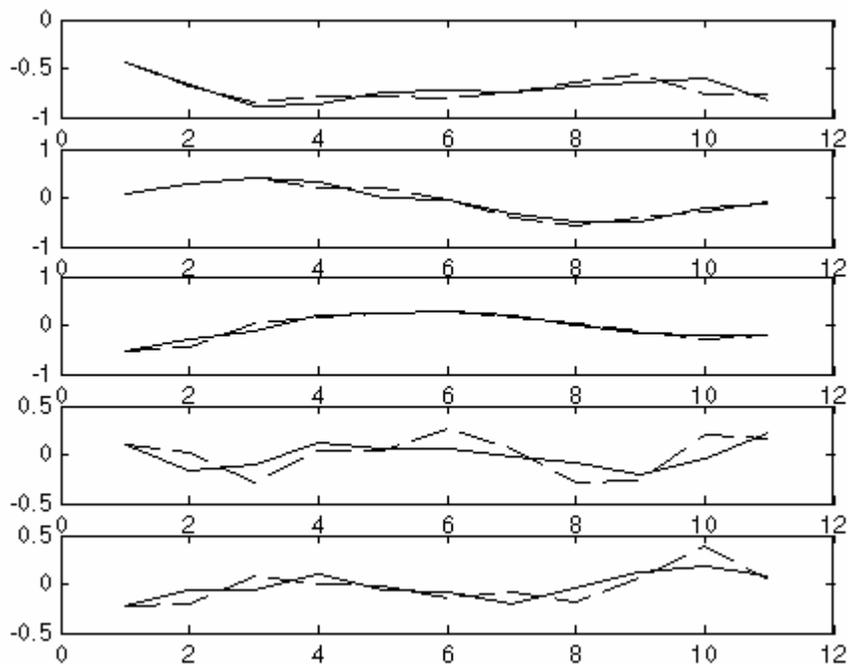


Figure 4.5 A comparison of the internal variable expression profiles in Table 4.1 and their calculated profiles from Equation (4.15) for dataset BAC

For dataset CDC15, five eigenvalues of the state transition matrix A described in Table 4.2 are $0.4262 \pm 0.8488i$, 0.5509 , and $0.7605 \pm 0.2950i$, all of which lie inside the unit circle (Figure 4.6). This means the inferred regulatory network for genes in dataset CDC15 is stable, and thus is robust to system noises, for example, the squared summable noises. Furthermore, the dominant eigenvalues of the network are a pair of conjugate complex numbers. Accordingly, this implies that at the stable states, the network behaves periodically. This result is not surprising because the genes in dataset CDC15 are cell-cycle regulated. In conclusion, the inferred gene regulatory network for dataset CDC15 has the properties of the real gene regulatory network.

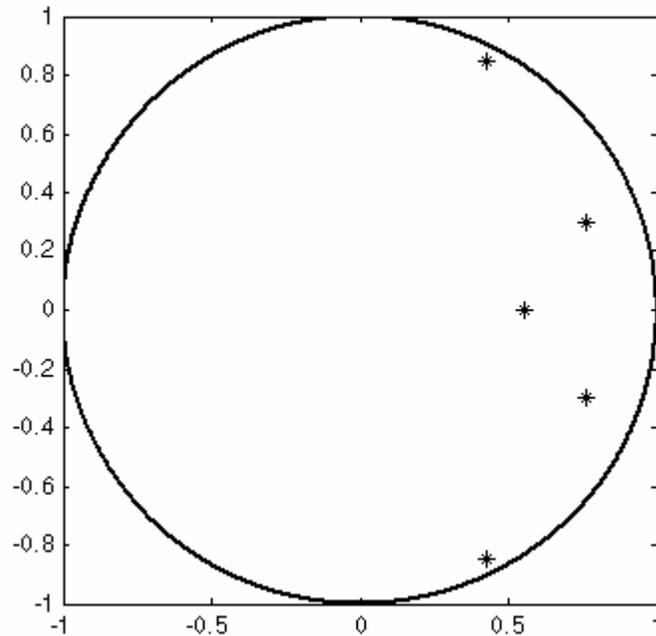


Figure 4.6 The distribution of eigenvalues of gene regulatory system for dataset CDC15

For dataset BAC, five eigenvalues of the state transition matrix A described in Table 4.2 are 1.0282 , $0.6835 \pm 0.4997i$, and $0.3092 \pm 0.5769i$. All of these except for the first one lie inside the unit circle (Figure 4.7). There are two possible reasons for the situation of the first eigenvalue. Firstly, there are noises in the gene expression dataset which cause the inaccuracy of parameters in the model. Secondly, there may be some structure feature of the real gene regulatory networks that is not captured by the current model. In this connection, the subsequent sections will present methods to improve the current model. Nonetheless, the first eigenvalue is very close to 1. This means that the inferred regulatory network is almost stable and robust. Furthermore, the behaviour of the inferred network appears approximately constant as the dominant eigenvalue (the first

one) is very close to the unit circle while other conjugate complex eigenvalues are far away from the unit circle.

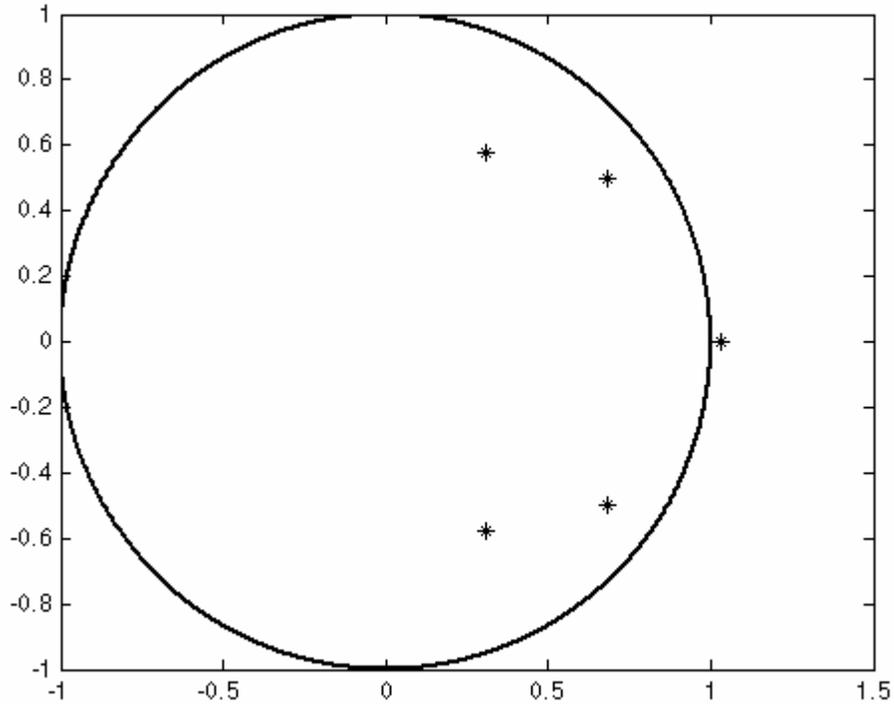


Figure 4.7 The distribution of eigenvalues of gene regulatory system for dataset BAC

In summary, this section has proposed a state-space model for inferring gene regulatory networks from time-course gene expression data, and the methods for model identification. The model is the state-space description of linear systems. The gene expression datasets, BAC and CDC15 were taken to illustrate how the methods work. The results demonstrate that for both datasets the inferred gene regulatory networks have some of features of the real gene regulatory networks, including the stability and the robustness. However, the inferred gene regulatory network for dataset BAC has no

periodicity at the stable states. Although this could be some uncontrollable noise in the dataset, a further attempt for improving the state-space model will be presented in the next section.

4.4 State-Space Model With Time Delays

The model proposed in the preceding section has not taken into account time delay in a cellular system. The real microarray data example reveals a considerable number of time delayed interactions, suggesting that time delay is ubiquitous in gene regulation (Dasika et al., 2004; Rosenfeld and Alon, 2003; Alter et al., 2000, 2001). From a biological viewpoint, time delay in gene regulation arises from the delays characterizing the various underlying processes, such as transcription, translation, and transportation. For example, time delays in regulation may stem from the time taken for the transportation of a regulatory protein to its site of action. Dasika et al. (2004) proposed a mixed integer linear programming framework for inferring time delays in gene regulatory networks. The high computational complexity of their algorithm hinders its application in the gene regulatory networks with a moderate number of genes as considered in this thesis. A straightforward attempt is therefore to extend the state-space model with consideration of time delays.

4.4.1 The Model

From Figure 4.1, the state-space model with time delays can mathematically be described by (Wu et al., 2004d, 2004e)

$$\begin{cases} \mathbf{z}(t+1) = \sum_{\tau=0}^{\tau_{\max}} \mathbf{A}_{\tau} \cdot \mathbf{z}(t-\tau) + \mathbf{n}_1(t) \\ \mathbf{x}(t) = \mathbf{C} \cdot \mathbf{z}(t) + \mathbf{n}_2(t) \end{cases} \quad (4.18)$$

where the vector $\mathbf{x}(t) = [x_1(t) \ \cdots \ x_n(t)]^T$ consists of the observation variables of the system, and $x_i(t)$ ($i = 1, \dots, n$) represents the expression level of gene i at time t , where n is the number of genes in the genetic regulatory network under consideration. The vector $\mathbf{z}(t) = [z_1(t) \ \cdots \ z_p(t)]^T$ consists of the internal state variables of the system and $z_i(t)$ ($i = 1, \dots, p$) represents the expression value of internal element i at time t which directly regulates gene expression, where p is the number of the internal state variables. The matrices $\mathbf{A}_{\tau} = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) are the time translation matrices of the internal state variables or the state transition matrices with time delay τ , while the integer parameter τ_{\max} denotes the maximum time delay accounted for. The matrices $\mathbf{A}_{\tau} = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) provide key information on the influences of the internal variables on each other. The matrix $\mathbf{C} = [c_{ik}]_{n \times p}$ is the transformation matrix between the observation variables and the internal state variables. The entries of this matrix encode information on the influences of the internal regulatory elements on the genes. Finally, the vectors $\mathbf{n}_1(t)$ and $\mathbf{n}_2(t)$ represent system errors and observation errors, respectively.

4.4.2. Model Identification

The task of parameter identification in model (4.18) is to estimate the elements in matrices $\mathbf{A}_\tau = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) and $\mathbf{C} = [c_{ik}]_{n \times p}$ such that both the system error and the observation error are minimized with some certain senses. Let \mathbf{X} be the gene expression data matrix with n rows and m columns, where n and m are the numbers of genes and time points in the dataset, respectively. The building of model (4.18) from microarray gene expression data \mathbf{X} can also be divided into two phases. Phase one extracts the internal state variables and their expression matrix \mathbf{Z} with p rows and m columns from the data matrix \mathbf{X} , and computes the transformation matrix \mathbf{C} such that Equation (4.13) holds. Phase two builds the dynamics equations of the internal states; i.e., determine the state transition matrices $\mathbf{A}_\tau = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$), from the expression matrix \mathbf{Z} . Phase one minimizes the observation error (i.e., maximize the data likelihood) with BIC, while Phase two minimizes the system error.

4.4.2.1 Extraction of the internal variables

In Section 4.3, the maximum likelihood factor analysis and EM algorithm (Lawley and Maxwell, 1971; Bubin and Thayer, 1982) were employed to extract the internal state variables and compute the transformation matrix from the gene expression data, i.e., to build Equation (4.13). The EM algorithm for the maximum likelihood estimate may fall into a local maximum (Dempster, 1977). Tipping and Bishop (1999) developed a probabilistic principal component analysis (PPCA) and, specifically, proposed two

methods for PPCA: maximum-likelihood algorithm and EM algorithm. Further, they proved that the maximum-likelihood algorithm for PPCA can find the global maximum.

In this section, the maximum-likelihood algorithm for PPCA (Tipping and Bishop,1999) is employed to extract the internal variables from time-course gene expression data, where \mathbf{X} is the $n \times m$ observation data matrix, each row of which is an observation sample; \mathbf{C} is the $n \times p$ transformation matrix, each row of which is a realization of latent variables; and \mathbf{Z} is the $p \times m$ loaded matrix, each row of which represents the expression profile of an internal state. Assume that the sample mean is shifted to zero. The log-likelihood for PPCA model is expressed by

$$L = -\frac{n}{2} \{m(\ln 2\pi) + \log|D| + \text{tr}(D^{-1}S)\}$$

where $D = Z^T Z + \sigma^2 I$ and $\mathbf{S} = \mathbf{X}^T \mathbf{X} / n$. For the given number of internal variables, p , the log-likelihood for the PPCA model finds its global maximum (Tipping and Bishop,1999)

$$L_k = -\frac{n}{2} \left\{ \sum_{j=1}^p \log(\lambda_j) + (m-p) \log \left(\frac{\sum_{j=p+1}^m \lambda_j}{(m-p)} \right) + m(\log(2\pi) + 1) \right\} \quad (4.19)$$

when

$$\mathbf{Z}_p = \mathbf{R}(\mathbf{Q}_p - \sigma^2 \mathbf{I}_p)^{1/2} \mathbf{U}_p^T \quad (4.20)$$

where λ_j ($j=1, \dots, p$) are the first p largest eigenvalues of the sample variance matrix \mathbf{S} , the matrix \mathbf{Q}_p is a $p \times p$ diagonal matrix, whose diagonal elements are these λ_j ($j=1, \dots, p$), \mathbf{U}_p is a $m \times p$ matrix, each column of which is a corresponding eigenvector of \mathbf{S} , \mathbf{I}_k is a $p \times p$ identity matrix, \mathbf{R} is an arbitrary $p \times p$ orthogonal matrix, and $\sigma^2 = \sum_{j=k+1}^m \lambda_j / (m-p)$.

Note that if $\{\mathbf{C}, \mathbf{Z}\}$ is an optimum solution of Equation (4.13), $\{\mathbf{CS}^{-1}, \mathbf{SZ}\}$ is its optimum solution, where \mathbf{S} is any $p \times p$ non-singular matrix. However, it can be proved that the state-space models from $\{\mathbf{C}, \mathbf{Z}\}$ and $\{\mathbf{CS}^{-1}, \mathbf{SZ}\}$ are algebraically equivalent (Chen, 1999). Therefore, one can always normalize the expression profiles of the internal state variables. For the optimum number of internal state variables, p , since $\mathbf{R}(\mathbf{Q}_p - \sigma^2 \mathbf{I}_p)^{1/2}$ is a $p \times p$ non-singular matrix, there is

$$\mathbf{Z} = \mathbf{U}_p^T \quad (4.21)$$

as the expression profiles of the internal state variables. Further, the corresponding transformation matrix \mathbf{C} can be calculated by formulae $\mathbf{C} = \mathbf{X} \cdot \mathbf{Z}^+$.

From Equation (4.19), the values of the maximum log-likelihood for the PPCA model increase with the increased numbers of internal state variables, p . The redundant

internal state variables may result in a complicated model. Since the PPCA has a solid probabilistic foundation, BIC is employed to determine the number of internal state variables, as in Section 4.3. For each model, the BIC is defined as:

$$BIC(p) = 2 \cdot L_p - \log(n) \cdot v_p \quad (4.22)$$

where n is the sample size (the number of genes), and $v_p (= mp + 1)$ is the number of parameters in the PPCA model. Note that the definition of BIC in Equation (4.22) is different from that in Equation (4.14) to avoid the negative BIC (Burnham and Anderson, 1998). Since the term $nm(\log(2\pi) + 1)/2$ in Equation (4.19) is a constant for a given dataset, the calculation of BIC can be simplified as

$$BIC(p) = -n \left\{ \sum_{j=1}^p \log(\lambda_j) + (m-p) \cdot \log \left(\frac{\sum_{j=p+1}^m \lambda_j}{m-p} \right) \right\} - \log(n) \cdot (mp + 1) \quad (4.23)$$

By this definition, the model with the largest BIC is chosen. Note that this definition of BIC is different from the Equation (4.14) on Page 98.

4.4.2.2 Identification of the internal state equation

After obtaining the expression matrix of the internal variables \mathbf{Z} and the transformation matrix \mathbf{C} in Phase one, the internal state transition equation

$$\mathbf{z}(t + \Delta t) = \sum_{\tau=0}^{\tau_{\max}} \mathbf{A}_{\tau} \cdot \mathbf{z}(t - \tau) \quad (4.24)$$

in model (4.18) can be established from the expression matrix \mathbf{Z} in Phase two. Each of state transition matrices \mathbf{A}_{τ} ($\tau = 0, \dots, \tau_{\max}$) contains p^2 unknown elements while the matrix \mathbf{Z} contains $m \cdot p$ known expression data points. If $(\tau_{\max} + 1)p > m$, Equation (4.24) will be underdetermined. To find the suitable state transition matrices, some additional conditions are necessary (Chen, 1999; Dasika, et al., 2004). Using BIC the number of chosen internal variables p is generally less than the number of time points m . Therefore these matrices can be identifiable if there are just a few time delays (e.g., $\tau_{\max} \leq 1$) accounted for.

For equally-spaced measurements of gene expression, the multivariable linear regression method (Aoki, 1990; Harvey, 1993) may be used to identify state transition matrices \mathbf{A}_{τ} ($\tau = 0, \dots, \tau_{\max}$). For unequally-spaced measurements, the problem becomes nonlinear, and it is necessary to determine these matrices by using an optimization technique such as those in Chapter 10 of Press's text (Press, et al., 1992).

The computational complexity: The computational cost in Phased one is bounded by the maximum likelihood algorithm for the PPCA and is $O(mn + m^3)$ (Tipping and Bishop, 1999). In Phase two, the computational cost is $O(mp + p^3)$. Since both m and p are much smaller than n , the overall computational cost of the state-space model identification is $O(n)$, i.e., linear in the number of genes in model. Such a

computational cost is much cheaper than that of the existing models such as the Boolean network models and differential/difference models (see Section 4.1).

4.4.3. Computational Experiments and Results

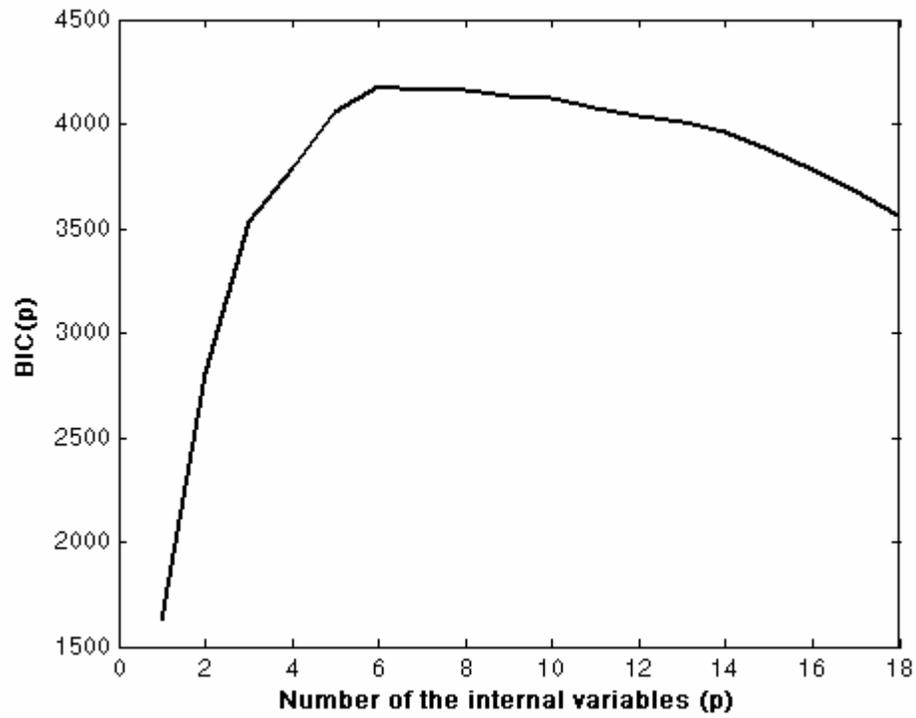


Figure 4.8 Profiles of BIC with respect to the number of internal variables for dataset ALP

To evaluate and illustrate the state-space model with time delays, gene expression datasets, ALP and ELU described in Section 2.2, are taken. The computational results are compared with the results from the state-space model without time delays proposed

in Section 4.3. Before applying PPCA, the expression profile for each gene was normalized to have a median of 0 and a standard deviation (from the median) of 1. Further the expression values of all genes on each microarray are normalized as so to have a mean of 0 and a standard deviation of 1. Thus in PPCA, the estimation of the mean is not needed (Tipping and Bishop, 1999).

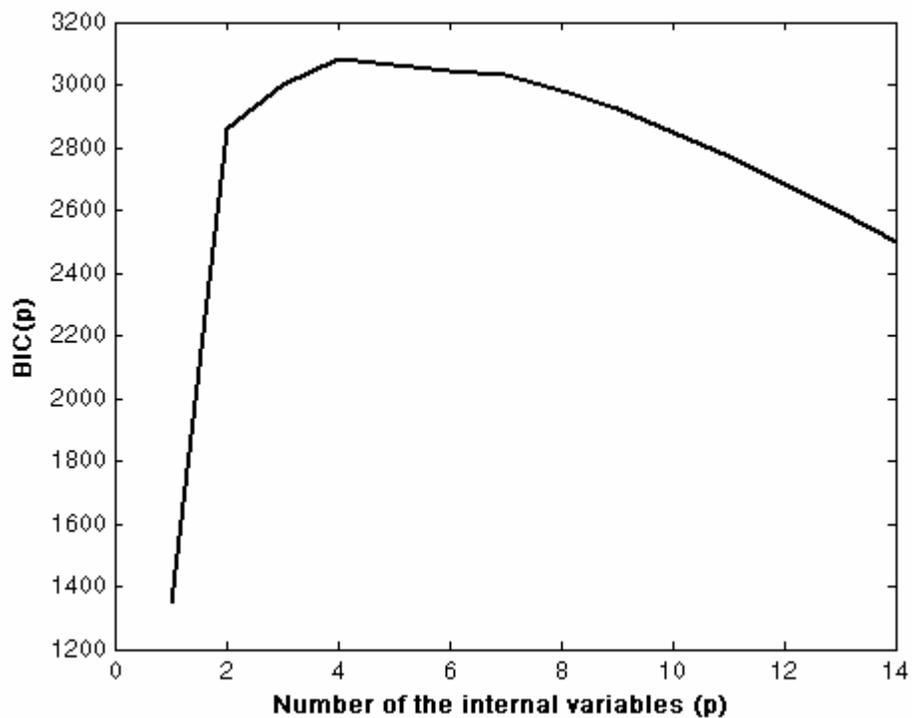


Figure 4.9 Plot for BIC with respect to the number of internal variables for dataset ELU

The maximum likelihood algorithm for PPCA (Tipping and Bishop, 1999) is employed to analyze the two datasets. For a variety of number p of internal state variables, $BIC(p)$ is calculated by Equation (4.23). Figures 4.8 and 4.9 depict the profiles of BIC

with respect to the number of internal variables for datasets ALP and ELU, respectively. With the BIC, one may conclude that the regulatory network for genes in dataset ALP has 6 internal variables from Figure 4.8 while the regulatory network for genes in dataset ELU has 4 internal variables from Figure 4.9.

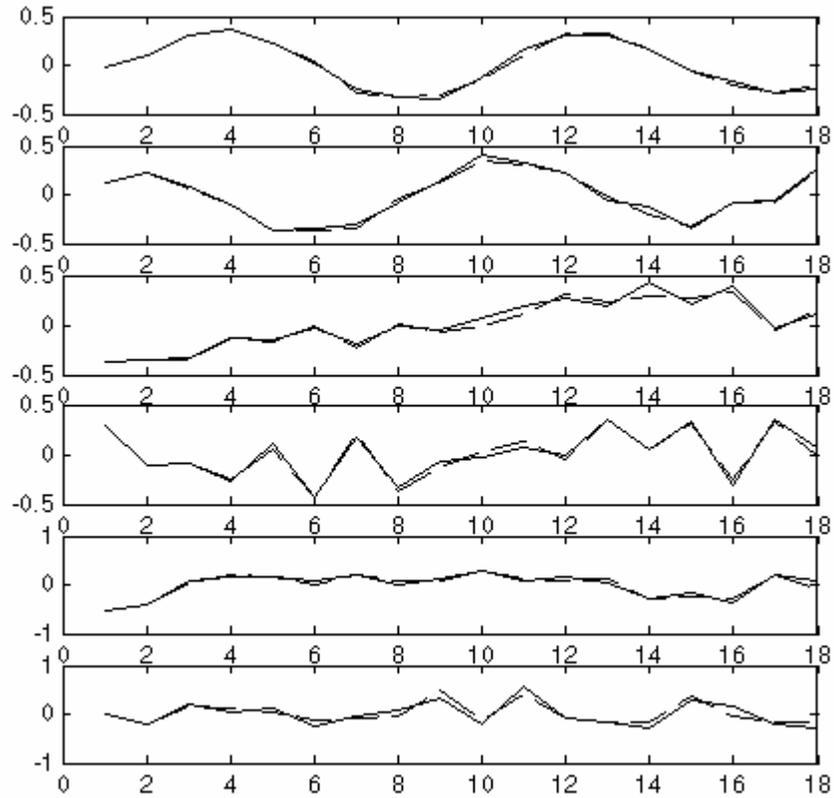


Figure 4.10 A comparison of 6 internal state expression profiles estimated by PPCA and predicted by dynamic equation model (4.24) for dataset ALP.

After the numbers of the internal variables are determined, both matrices \mathbf{Z} and \mathbf{C} can be represented in the form of Equation (4.13). Since the two datasets under

consideration are collected at equally-spaced time points, the multivariate regression method (Aoki, 1990; Harvey, 1993) is employed to determine the state transition matrices $\mathbf{A}_\tau (\tau=0, \dots, \tau_{\max})$ in the models from the expression matrices of internal variables, \mathbf{Z} 's. In this work, $\tau_{\max} = 1$ for both two datasets is taken.

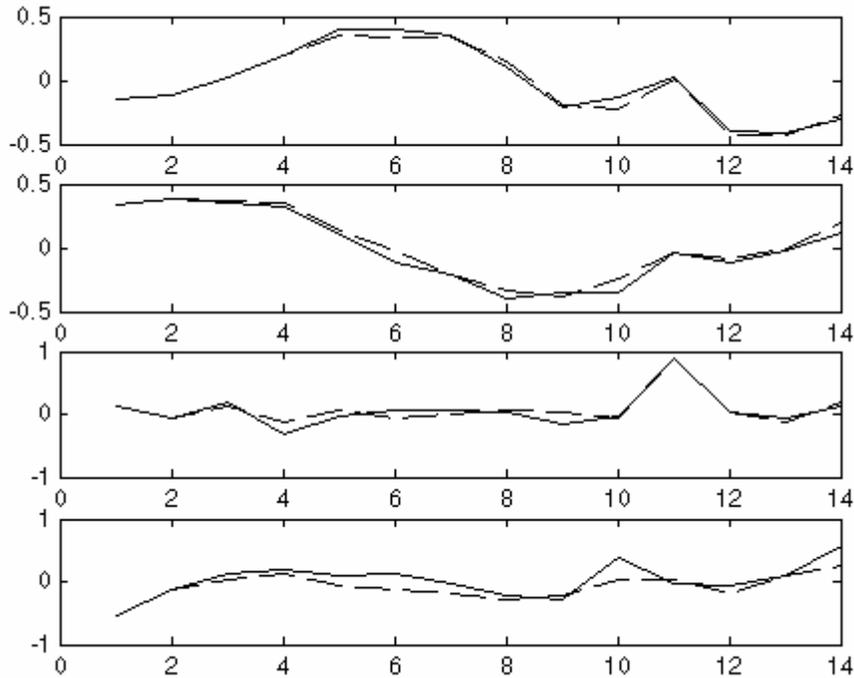


Figure 4.11 A comparison of 4 internal state expression profiles estimated by PPCA and predicted by dynamic equation model (4.24) for dataset ELU.

Figures 4.10 and 4.11 depict comparisons of the internal state profiles estimated by PPCA and predicted by the dynamic equation (4.24) for two datasets, respectively. In these figures, the solid lines stand for the estimated profiles; and the dash lines for the predicted profiles. These figures show that two kinds of profiles match very well for

both datasets. Furthermore, to quantitatively evaluate the state-space models with time delays, the prediction errors P_E (defined by Equation (4.10)) for both the model with time delays and the model without time delays are calculated and compared (Table 4.3).

Table 4.3 Comparisons of prediction error between the state-space models with time delays and without time delays for datasets ALP and ELU.

	Without time delays	With time delays	Improvement (%)
ALP	0.0844	0.0258	69.43
ELU	0.1286	0.0519	59.64

In Table 4.3, the *improvement* is defined as

$$P_E(\text{improvement}) = \frac{P_E(\text{without time delay}) - P_E(\text{with time delay})}{P_E(\text{without time delay})} \quad (4.25)$$

From Table 4.3, the prediction error of the space-state model without time delays presented in Section 4.3 is 0.0844 while the prediction error of the model with time delays proposed in this section is 0.0258 for dataset ALP. Comparing the state-space model without time delay, the state-space model with time delay improve in terms of the prediction error by about 70% for dataset ALP. Similarly, for dataset ELU the state-space model with time delay improve in terms of the prediction error by about 60%, as compared to the state-space model without time delay. These results demonstrate that

the state-space model with time delays outperforms the model without time delays for gene regulatory networks.

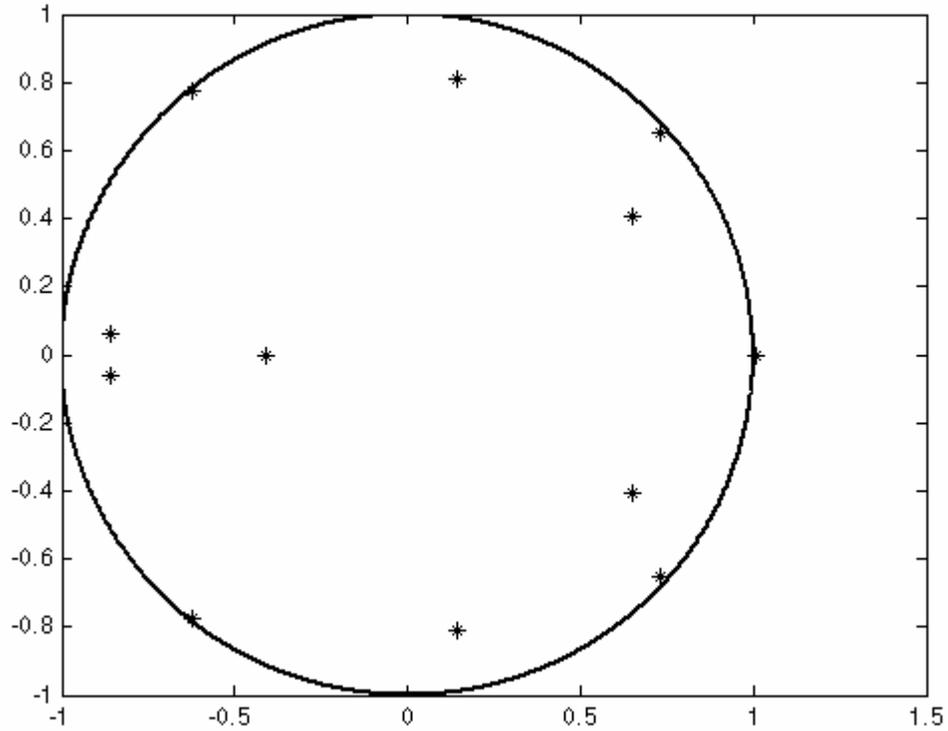


Figure 4.12 The distribution of eigenvalues of model with time delays for dataset ALP

To inspect the stability, the robustness, and the periodicity of the inferred gene regulatory networks based on the state-space model with time delays, the eigenvalues of the matrix \mathbf{T} in (4.9) are calculated for the models from both datasets, respectively. For dataset ALP with $\tau_{\max} = 1$, the matrix \mathbf{T} in (4.9) has twelve eigenvalues: two real numbers, 1.0073 and -0.4074 ; and five pairs of conjugate complex numbers, $0.6244 \pm 0.7757i$, $0.7282 \pm 0.6498i$, $-0.8580 \pm 0.0588i$, $0.1448 \pm 0.8083i$, 0.6498

$\pm 0.4086i$. All of these eigenvalues except for the first real eigenvalue lie inside the unit circle in the complex plane. However, the first real eigenvalue is very close to the boundary of the unit circle (Figure 4.12). This means that the inferred regulatory network for genes in dataset ALP is almost stable and robust. Furthermore, the dominant eigenvalues of the network are two pairs of conjugate complex numbers and a real number which are very close to the unit circle. Accordingly, this implies that at the stable states, the network behaves periodically. This result is not surprising because the genes in dataset ALP are cell-cycle regulated.

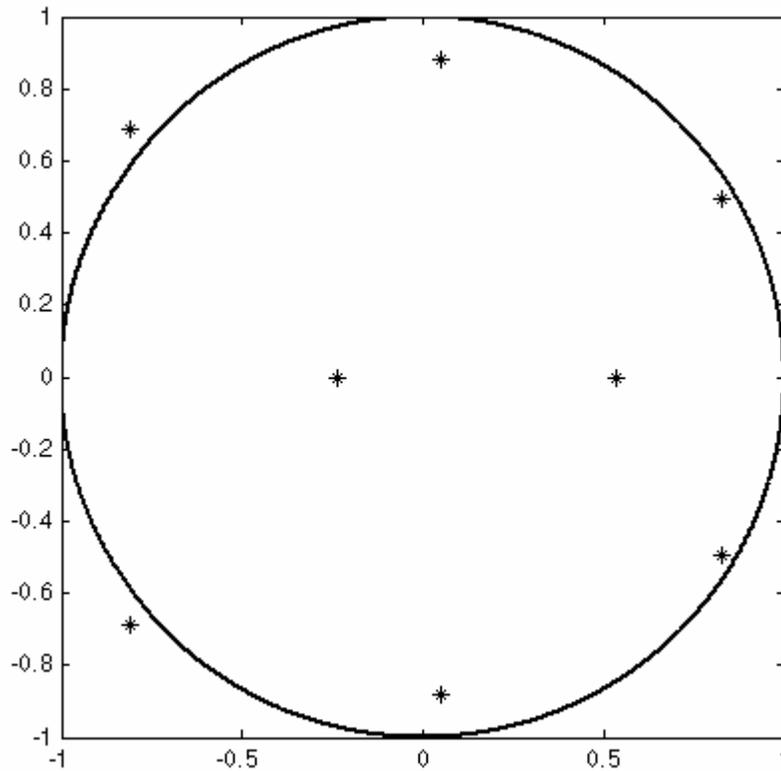


Figure 4.13 The distribution of eigenvalues of model with time delays for dataset ELU

For dataset ELU with $\tau_{\max}=1$, the matrix T in (9) has eight eigenvalues: two real numbers, 0.5344 and -0.2357 ; and three pairs of conjugate complex numbers: $-0.8079 \pm 0.6862i$, $0.8268 \pm 4940i$, and $0.0480 \pm 0.8810i$. All of these eigenvalues except for $-0.8079 \pm 0.6862i$ lie inside the unit, but their modulus is 1.0600 and is very close to the unit circle in the complex plane (Figure 4.13). This means the inferred regulatory network for genes in dataset ELU is almost stable and robust. Furthermore, the dominant eigenvalues of the network are two pairs of conjugate complex numbers which are very close to the unit circle. Accordingly, this implies that at the stable states, the network behaves periodically. Again this result is not surprising because the genes in dataset ELU are cell-cycle regulated as well.

In summary, this section proposed a state-space model with time delays for gene regulatory networks and the methods for model identification. Applications of this model to two gene expression datasets ELU and ALP have showed that the model with time delays has more prediction power than the model without time delays in Section 4.3, and has some features of the real gene regulatory network, for example, the stability, the robustness, and the periodicity.

4.5 Genetic Algorithm for Inferring Time Delays

In order to uniquely determine the state transition matrices, $p^2(\tau_{\max} + 1)$ equations are needed from Equation (4.24), where p is the number of internal variables and τ_{\max} is the maximum number of the discrete time delays accounted for. As there are mp

expression values of internal variables, only mp equations are available. This implies that the system parameters can be estimated only if $m > p(\tau_{\max} + 1)$, where m is the number of time points in gene expression dataset. This case is considered as $\tau_{\max} = 1$ in Section 4.4. In reality, for many gene expression data, the inequality $m > p(\tau_{\max} + 1)$ does not come true even for the case $\tau_{\max} = 1$, and implying the system is underestimated. Although Dasika et al. (2004) assumed that each regulatory interaction has only one single time delay to uniquely identify the parameters of the system, the solution space is still too large to search for the optimal time-delayed regulatory relationship using an exhaustive search method.

This section employs Boolean variables to capture the existence of the discrete time delays of the regulatory relationships among the internal variables, and proposes a genetic algorithm (GA) to determine the optimal Boolean variables (corresponding to the optimal time-delayed regulatory relationships), and to further infer gene regulatory networks with time delays (Wu et al., 2004f). Computational experiments will be performed on two datasets BAC and CDC28 described in Section 2.2 to evaluate the performance of the proposed method.

4.5.1 The Model

To emphasize the time-delayed relationships, the state-space model with time delays can be described as:

$$\left\{ \begin{array}{l} \mathbf{z}(t+1) = \sum_{\tau=0}^{\tau_{\max}} \mathbf{B}_{\tau} \circ \mathbf{A}_{\tau} \cdot \mathbf{z}(t-\tau) + \mathbf{n}_1(t) \\ \mathbf{x}(t) = \mathbf{C} \cdot \mathbf{z}(t) + \mathbf{n}_2(t) \end{array} \right. \quad (4.26)$$

where the symbol “ \circ ” denotes the Hadamard (element-wise) multiplication of two matrices (Schott, 1999). The matrices $\mathbf{B}_{\tau} = [b_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) are Boolean matrices, which capture the time-delayed regulatory relationships, $b_{ij\tau} = 1$ if internal variable j regulates internal variable i with time delay τ , and $b_{ij\tau} = 0$ otherwise, and

$$\sum_{\tau=0}^{\tau_{\max}} b_{ij\tau} = 1 \quad (i, j = 1, \dots, p). \quad (4.27)$$

The meanings of other symbols in (4.26) are the same as those in Equation (4.18). Note that Equation (4.27) mathematically describes the assumption that each regulatory interaction has only one single time delay.

The task of parameter identification in model (4.26) is to estimate the elements in matrices $\mathbf{A}_{\tau} = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) and $\mathbf{C} = [c_{ik}]_{n \times p}$ such that both the system error and the observation error are minimized with some senses. As done in Section 4.4, model (4.26) is constructed in two phases. Phase one employs PCCA and BIC to estimate the number of internal variables and their expressions from gene expression data, and to establish the observation equations (the lower one in (4.26)), by minimizing

the observation error with BIC. We can use PPCA (Tipping and Bishop, 1999) and BIC to estimate the number of internal variables, p , and their expression matrix Z as did in Section 4.4.2. Phase two employs the GA to find optimal Boolean matrices \mathbf{B}_τ (i.e. to determine the optimal time-delayed relationships) and the multiple regression method to determine $\mathbf{A}_\tau = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) by minimizing the prediction error.

4.5.2 Genetic Algorithm

The solution space for $\mathbf{B}_\tau (\tau = 0, \dots, \tau_{\max})$ consists of all Boolean matrix sets $\{\mathbf{B}_0, \dots, \mathbf{B}_{\tau_{\max}}\}$ satisfying (4.27), denote by SB , and is too large to use an exhaustive algorithm for searching for the optimum solution. Therefore, a genetic algorithm (GA) is proposed to find the optimum $\{\mathbf{B}_0, \dots, \mathbf{B}_{\tau_{\max}}\}$ as shown in Figure 4.14. In Figure 4.14, GEN is the number of generations of the GA, and is set by the users. In the following the computational fitness (CF) operator, the encoding, the selection operator, the crossover operator and the mutation operator will be discussed.

Encoding: Define a matrix set BB consisting of all $p \times p$ matrices on set $\{0, \dots, \tau_{\max}\}$.

Further define a mapping from SB to BB :

$$\{\mathbf{B}_0, \dots, \mathbf{B}_{\tau_{\max}}\} \in SB \mapsto \mathbf{B} = [b_{ij}] \in BB \quad (4.28)$$

and $b_{ij} = \tau$ if $b_{ij\tau} = 1$ ($\tau = 0, \dots, \tau_{\max}$). It is obvious that this mapping is one-to-one.

Thus it is sufficient to encode the set BB . The VEC operator (Schott, 1997) is employed to transfer $\mathbf{B} = [b_{ij}]_{p \times p} \in BB$ into an integer string with length p^2 over the set

$\{0, \dots, \tau_{\max}\}$. It is natural to use such an integer string to encode matrix $B = [b_{ij}]_{p \times p} \in BB$. Note that such a string completely describes a set of time-delayed relationships in gene regulatory network (4.26) and is called an individual in terms of the GA, denoted by s . A population is composed of N individuals, denoted by Σ , where N is an odd positive integer. One may set some additional conditions to refine the search space. For example, if some time-delayed relationships are known, one may set some fixed values for the elements expressing these relationships on the string.

1. Randomly select N integer string with length p^2 over the set $\{0, \dots, \tau_{\max}\}$ to make up a population Σ , $g = 1$
2. $[F, \mu, \sigma^2, \Sigma] = CF(\Sigma, N)$, denote $s^* = s_1$, $f(g) = F(1)$
3. While ($g \leq GEN$)
4. $g = g + 1$
5. $\tilde{\Sigma} = Selection(\Sigma, N)$;
6. $\Sigma = Crossover(\tilde{\Sigma}, N)$;
7. $\Sigma = Mutation(\Sigma, Pm, N)$;
8. $[F, \mu, \sigma^2, \Sigma] = CF(\Sigma, N)$;
9. If $F(1) < f(g)$, then $s^* = s_1$, and set $f(g) = F(1)$ **else** $f(g) = f(g - 1)$;
10. End while
11. Return $f(GEN)$ and s^* .

Figure 4.14 Genetic Algorithm for inferring time-delayed relationships

CF operator--- $[F, \mu, \sigma^2, \Sigma] = CF(\Sigma, N)$: The CF operator does the following things: (1) to calculate the fitness values $F(s_i)$ ($i = 1, \dots, N$) for all individuals in population Σ by using Equation (4.10), (2) to order individuals such that the first individual is the optimal one, and (3) to estimate μ and σ^2 of normal distribution $N(\mu, \sigma^2)$ using the maximum likelihood estimate methods (Vardeman, 1994) from fitness values $F(s_i)$ ($i = 1, \dots, N$) of all individuals in population Σ . Note that $F(s_1)$ is not smaller than the fitness value of any other individual in population Σ .

Selection operator--- $\tilde{\Sigma} = Selection(\Sigma, N)$: The selection operation creates a mediate population $\tilde{\Sigma}$. For convenience of the manipulation, the GA always assigns the best individual found over time in the population to individual 1 and copies it to the next population. Operator $\tilde{\Sigma} = Selection(\Sigma, N)$ selects $(N-1)/2$ individuals from the previous population according to the normal distribution $N(\mu, \sigma^2)$. The GA employs the prediction error as the fitness value $F(s_i)$ of an individual s_i calculated by Equation (4.10) in Section 4.2. Note that there are only $(N-1)/2 + 1$ individuals in the mediate population $\tilde{\Sigma}$.

Crossover operator--- $\Sigma = Crossover(\tilde{\Sigma}, N)$: The intention of the crossover operation is to create new (and hopefully better) individuals from two selected parent individuals. In the GA, of two parent individuals, one is always the first individual that is the optimal individual found over time, and the other is the one selected from the $(N-1)/2$ individuals out of the parent population other than the first individual in the mediate

population $\tilde{\Sigma}$. Here, the crossover operator adopts the single-point crossover method for simplicity. Note that after the crossover operation, population Σ has N individuals.

Mutation operator--- $\Sigma = \text{Mutation}(\Sigma, P_m, N)$: Each position in a coding string is randomly selected with a mutation probability P_m , and the number in the selected position is uniformly randomly replaced by another integer from the set $\{0, \dots, \tau_{\max}\}$.

4.5.3. Computational Experiments and Results

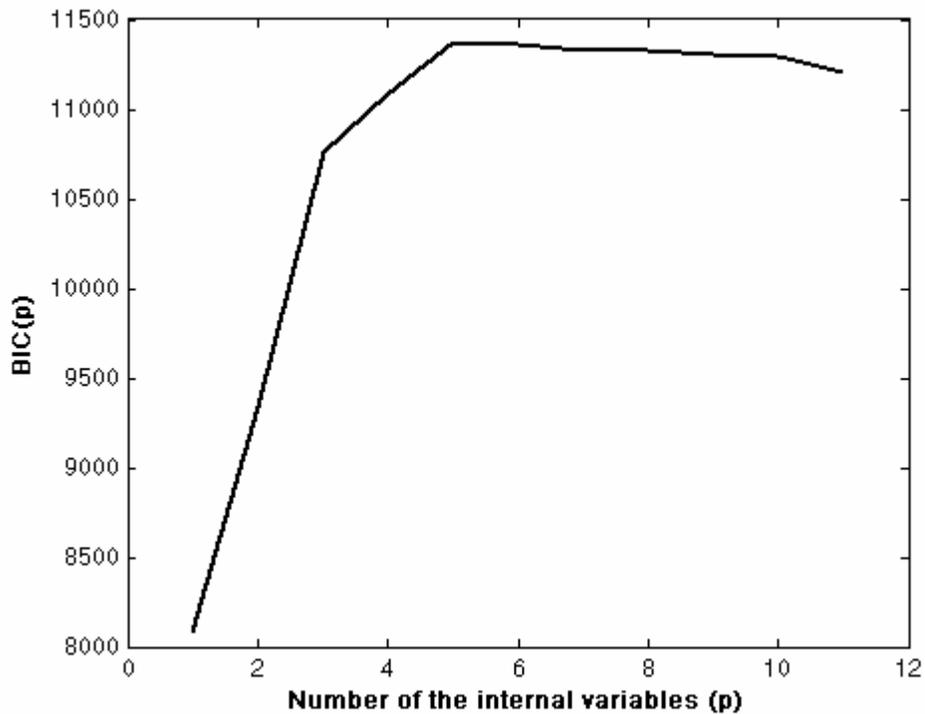


Figure 4.15 Profiles of BIC with respect to the number of internal variables for dataset BAC

To evaluate the proposed method, it is applied to two datasets BAC and CDC28 described in Section 2.2. The expression profile for each gene is normalized to have a median of 0 and a standard deviation (from the median) of 1. Further, the expression values of all genes on each microarray are normalized as so to have a mean of 0 and a standard deviation of 1. Thus in PPCA, there is no need to estimate the mean in the PPCA model (Tipping and Bishop, 1999).

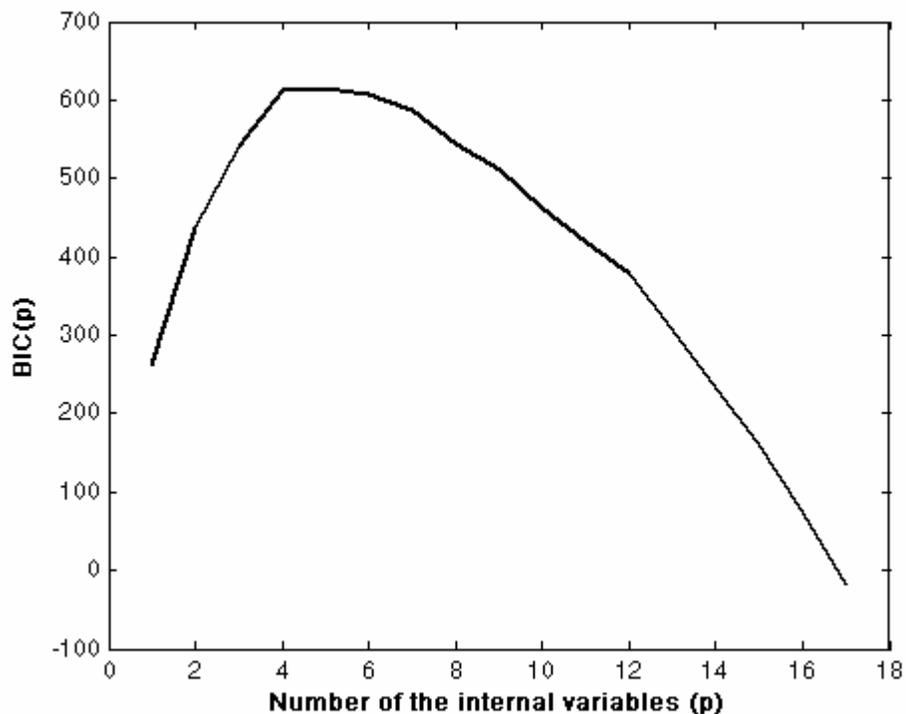


Figure 4.16 Profiles of BIC with respect of the number of the internal variables

For dataset CDC28

Using PPCA and BIC (defined by Equation (4.23)), the profiles of BIC with respect to the number of internal variables are shown in Figures 4.15 and 4.16 for datasets BAC

and CDC28, respectively. The BICs reach their maximum values at five for both datasets. This means that gene regulatory networks for both dataset BAC and dataset CDC28 should have five internal variables.

The elements of matrices $\mathbf{A}_\tau = [a_{ij\tau}]_{p \times p}$ ($\tau = 0, \dots, \tau_{\max}$) are estimated using the proposed GA described in Figure 4.14. The case $\tau_{\max} = 1$ is considered for the sake of simplicity. Let the size of the population $N = 30$, the number of maximum generations and the mutation probability $P_m = 0.02$.

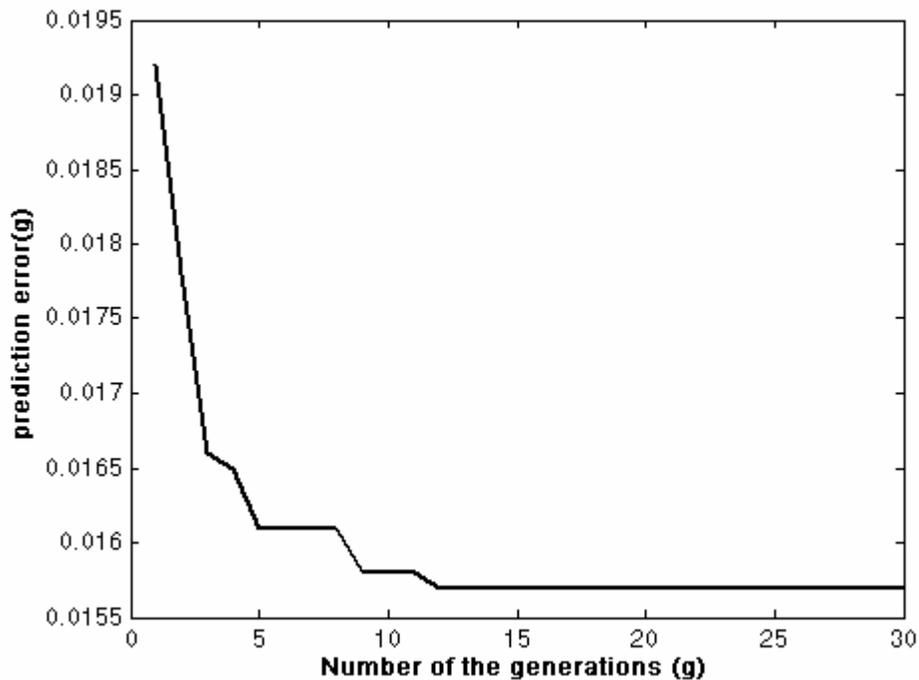


Figure 4.17 Plot of prediction error with respect to the number of generations for dataset BAC

Figure 4.17 depicts the profile of the prediction error with respect to the number of generations of GA for dataset BAC. The GA converges in 15 generations from Figure 4.17. At the convergence of GA, it follows that

$$B_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \text{ and } B_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (4.29)$$

From the optimal B_τ ($\tau = 0,1$) above, it follows that 14 (of 25) regulatory relationships are time-delayed. Further, the elements of matrices $B_\tau \circ A_\tau$ ($\tau = 0,1$) are estimated as follows:

$$B_0 \circ A_0 = \begin{bmatrix} 0 & 0 & 0 & 0.1134 & 0.1295 \\ -0.0708 & 0 & 0 & 0 & 0.5425 \\ -0.1552 & 0 & 0 & -0.0671 & -0.1574 \\ 0 & 0 & 1.0023 & 0.7894 & 0 \\ 0 & -0.3501 & 0 & 0 & 0.5815 \end{bmatrix} \quad (4.30a)$$

and

$$B_1 \circ A_1 = \begin{bmatrix} 0.9351 & -0.0121 & 0.0176 & 0 & 0 \\ 0 & 0.7299 & -0.5670 & -0.3980 & 0 \\ 0 & 0.54331 & 0.4165 & 0 & 0 \\ -0.3115 & 0.4018 & 0 & 0 & 1.8381 \\ -0.3701 & 0 & -0.7534 & -0.5879 & 0 \end{bmatrix} \quad (4.30b)$$

Using model (4.26) with the estimated matrices B_τ and $A_\tau(\tau=0,1)$, the predicted behaviour of the internal variables can be calculated (as show in Figure 4.18). It shows that the prediction performance of model (4.26) for the interval variables is pretty good. To evaluate the prediction performance of model (4.26), Equation (4.10) is employed to calculate the predication errors of the inferred networks model (4.26) and model (4.12). The degree of improvement is calculated by Equation (4.25). These results are listed Table 4.4 and show that the gene regulatory network with time delays may improve in terms of the prediction error by 64% for dataset BAC.

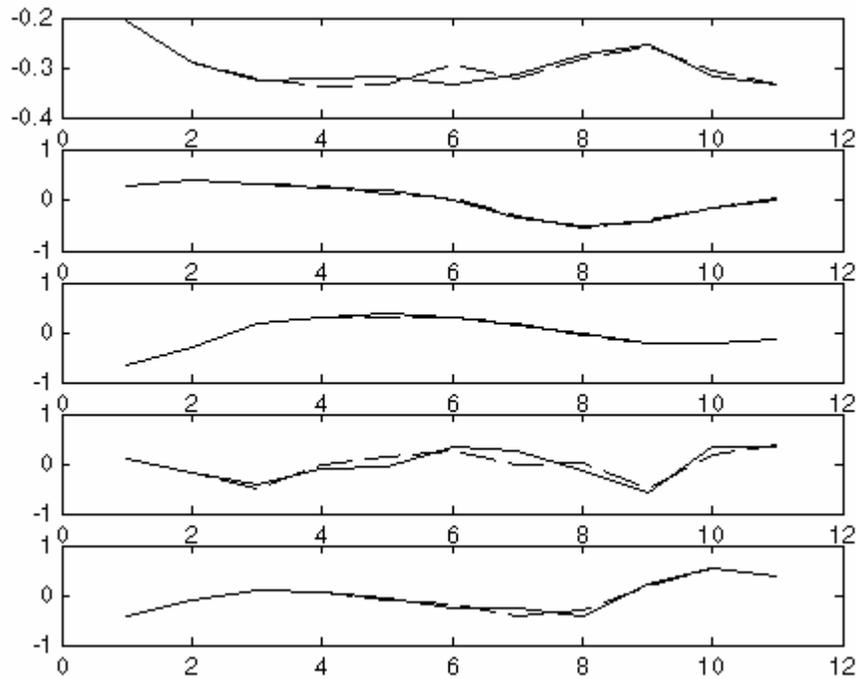


Figure 4.18 A comparison of 5 internal state expression profiles estimated by PPCA and predicted by dynamic equation model with time delays for dataset BAC.

The solid line: estimated profiles; and the dash lines: predicted profiles.

To inspect the stability, the robustness, and the periodicity of the inferred gene network from dataset BAC, the eigenvalues of matrix T are calculated by substituting matrices in (4.30a) as A_0 and in (4.30b) as A_1 into the generalized matrix T in Equation (4.9).

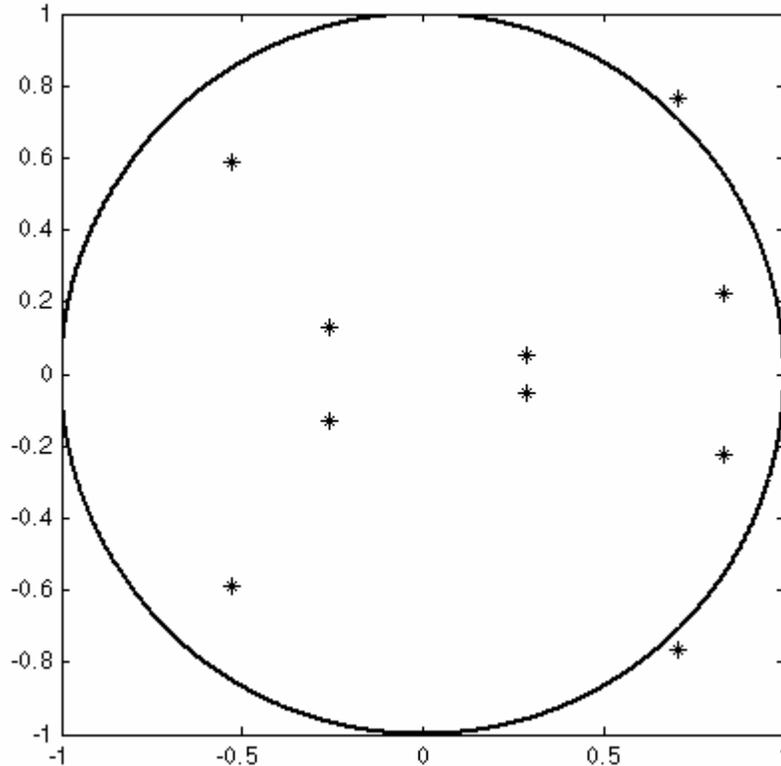


Figure 4.19 The distribution of eigenvalues of the inferred gene regulatory network with time delays for dataset BAC

The 10 eigenvalues of the inferred gene regulatory network are $0.7061 \pm 0.7673i$, $0.8332 \pm 0.2225i$, $-0.5284 \pm 0.5904i$, $-0.2564 \pm 0.1324i$, $0.2862 \pm 0.0514i$, all of which except for the first pair are inside the unit circle. The modulus of the first pair of eigenvalues is 1.0427 which is very close to the unit circle as shown Figure 4.19. From

the difference equation theory, the inferred regulatory network from dataset BAC is almost stable and robust. Furthermore, as the dominant eigenvalues are a pair of conjugate complex numbers, the behaviour of the systems is periodic at the stable states. These are the expected properties of a real gene regulatory network. Recall that the inferred network without time delays from the same dataset discussed in Section 4.3 is not periodic.

Table 4.4 Comparison of prediction power between the state-space models with time delays and without time delays for dataset BAC and CDC28

	Without time delays	With time delays	Improvement (%)
BAC	0.0430	0.0157	64.39
CDC28	0.2284	0.1530	33.01

Figure 4.20 depicts the profile of the prediction error with respect to the number of generations of GA for dataset CDC28. The GA converges in 15 generations from Figure 4.20. At the convergence of GA, it follows that

$$B_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (4.31)$$

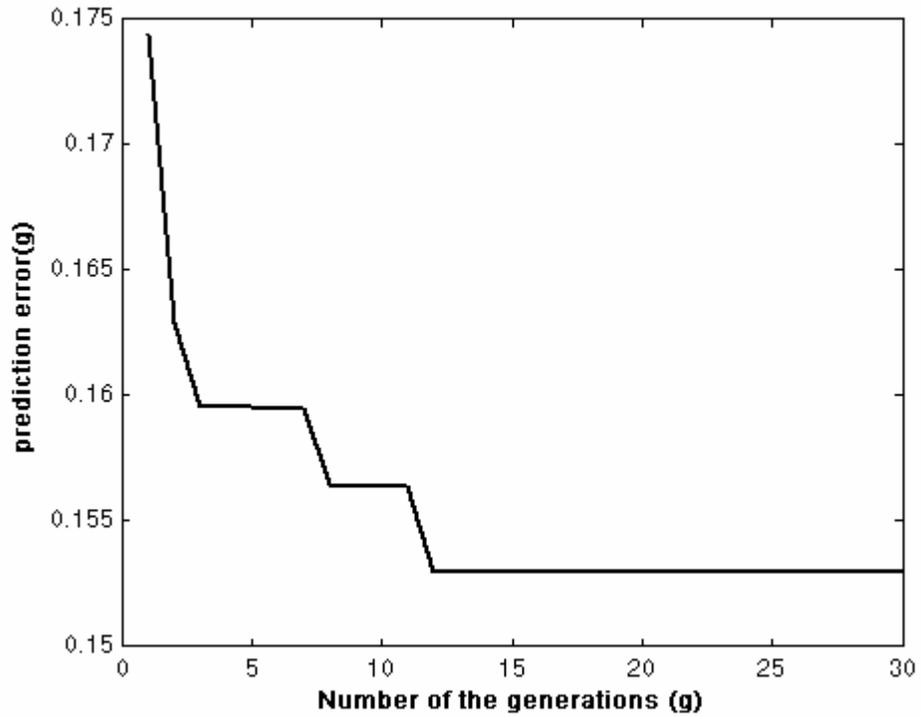


Figure 4.20 Plot of prediction error with respect to the number of generations for dataset CDC28

From the optimal $\mathbf{B}_\tau(\tau = 0,1)$ above, it follows that 12 (of 25) regulatory relationships are time-delayed. Further, the elements of matrices $\mathbf{B}_\tau \circ \mathbf{A}_\tau(\tau = 0,1)$ are estimated as follows:

$$\mathbf{B}_0 \circ \mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.0319 \\ 0 & 0.6024 & -0.7635 & 0.3904 & 0 \\ -0.6027 & 0 & 0.4786 & -0.0995 & -0.2749 \\ -0.0609 & 0 & 0.2701 & -0.2514 & 0 \\ 0 & 0.4201 & 0.1453 & 0 & 0 \end{bmatrix} \quad (4.32a)$$

and

$$\mathbf{B}_1 \circ \mathbf{A}_1 = \begin{bmatrix} 0.8714 & -0.3649 & 0.3679 & -0.1251 & 0 \\ 1.1498 & 0 & 0 & 0 & 0.3450 \\ 0 & 0.8347 & 0 & 0 & 0 \\ 0 & 0.2686 & 0 & 0 & -0.8007 \\ 0.1170 & 0 & 0 & 0.0432 & 0.4207 \end{bmatrix} \quad (4.32b)$$

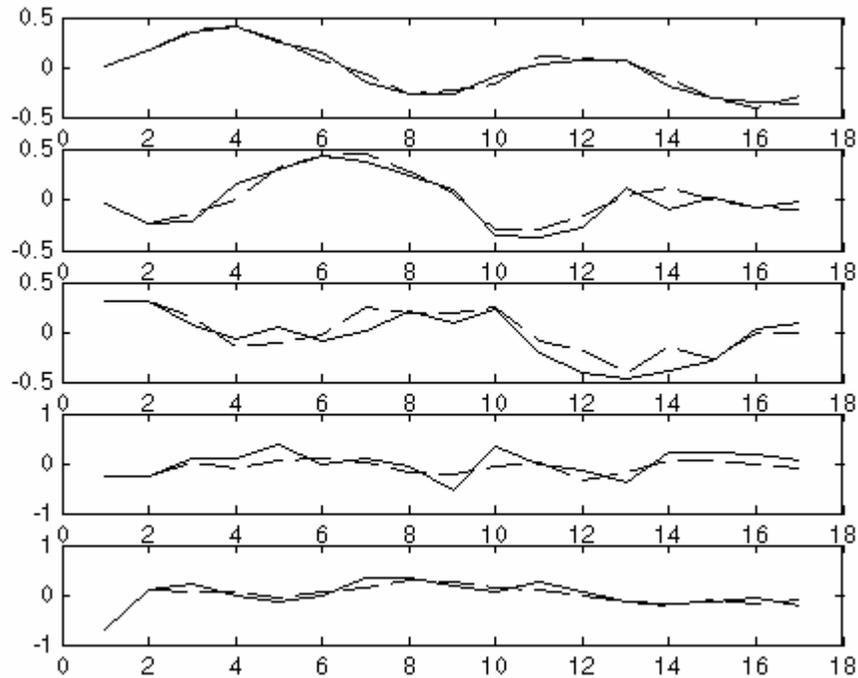


Figure 4.21 A comparison of 5 internal state expression profiles estimated by PPCA and predicted by dynamic equation model with time delays for dataset CDC28.

The solid line: estimated profiles; and the dash lines: prediction profiles.

Using model (4.26) with the estimated matrices $\mathbf{B}_\tau(\tau=0,1)$ and $\mathbf{A}_\tau(\tau=0,1)$, the predicted behaviour of the internal variables can be calculated, as shown in Figure 4.21.

It shows that the prediction performance of model (4.26) for the interval variables is

pretty good. To evaluate the prediction performance of model (4.26) for dataset CDC28, Equation (4.10) is employed to calculate the predication errors of the inferred networks model (4.26) and model (4.12). The degree of improvement is calculated by Formulae (5.25). These results are listed Table 4.4 and show that the gene regulatory network with time delays may improve the prediction accuracy by 33% for dataset CDC28.

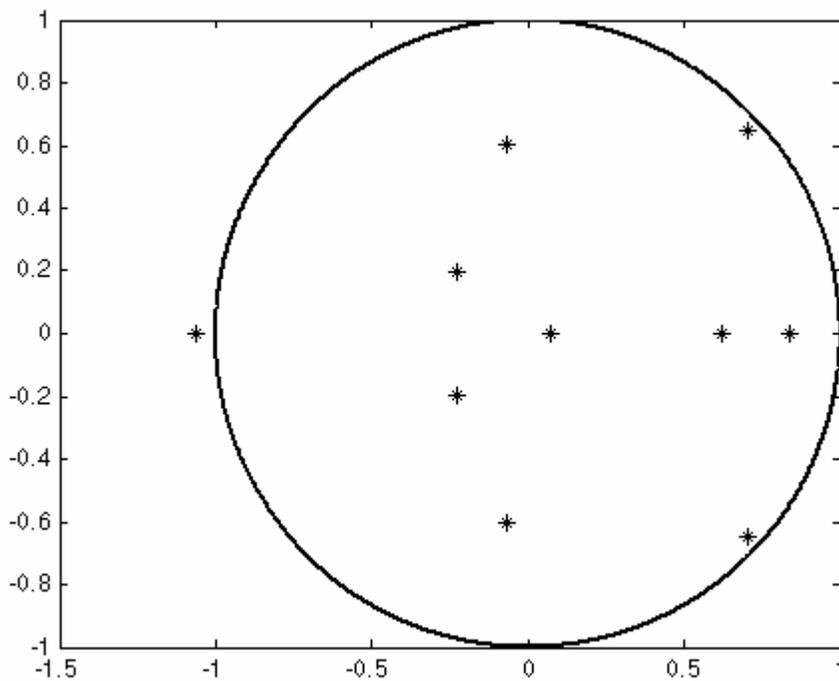


Figure 4.22 The distribution of eigenvalues of gene regulatory network with time delays for dataset CDC28

To investigate the stability, the robustness, and the periodicity of the inferred gene network from dataset CDC28, The eigenvalues of matrix T are calculated by substituting matrices in (4.32a) as A_0 and in (4.32b) as A_1 into the generalized matrix T in Formulae (4.9). The 10 eigenvalues of the inferred gene regulatory network are -

1.0605, $0.7036 \pm 0.6467i$, 0.8347, 0.6208, $-0.0703 \pm 0.6051i$, $-0.2292 \pm 0.1946i$, 0.0709, all of which except for the first real one are inside the unit circle. However, the first eigenvalue (-1.0605) is very close to the unit circle, as shown in Figure 4.22. From the difference equation theory, the inferred regulatory network from dataset CDC28 is almost stable and robust. Furthermore, as the dominant eigenvalues are the real number eigenvalues (-1.0605) and a pair of conjugate complex numbers ($0.7036 \pm 0.6467i$, whose modulus is 0.9557), the behaviour of the systems is periodic at the stable states. These are the expected properties of the inferred gene regulatory network as genes in this dataset are associated with cell division process of the yeast (Cho et al., 1998).

In summary, this section has proposed a GA approach to infer the time-delayed relationships in gene regulatory networks. Applications of this approach to two gene expression datasets BAC and CDC28 has shown that the GA approach can effectively infer the time-delayed relationships in gene regulatory networks, and with optimal time-delayed relationships, the model with time delays has not only more prediction power than the model proposed in Section 4.3, but also some features of the real gene regulatory network, for example, the stability, the robustness, and the periodicity.

4.6 Conclusions

This chapter firstly proposed a state-space model for gene regulatory networks and the use of MLFA, BIC, and multiple regression method for model identification. The analysis shows that the computational complexity to identify the proposed model is

much lower than that to identify the Boolean network models and differential/difference equation models. The computational experiments on two gene expression datasets have shown that the inferred gene regulatory networks for the two dataset have some features of the real gene regulatory networks, such as the stability and robustness.

Furthermore, a state-space model with time delays for inferring gene regulatory networks is proposed, which is an extension of the state-space model to account for the time-delayed relationships in the real gene regulatory networks. The PPCA, the BIC, and the multiple regression method are employed to identify the proposed model. The results of computational experiments have show that not only does the model with time delays improve the predication power as compared to the model without time delays, but also have more features of the real gene regulatory networks than the model without time delays, for example, the periodicity besides the stability and the robustness.

Finally, a genetic algorithm is proposed for inferring time delays in gene regulatory networks. Computational experiments on two gene expression datasets are performed to evaluate the proposed algorithm. The results show that the proposed algorithm can effectively infer time-delayed relationships in gene regulatory networks, and that with optimal time-delayed relationships the model with time delays has not only more prediction power than the model without time delay, but also some features of the real gene regulatory network, for example, the stability, the robustness, and the periodicity.

SUMMARY AND FUTURE WORK

5.1 Summary

Time-course gene expression data contain much information that may lead to new insights into the understanding of biological processes. Yet, the analysis of such data is more difficult because of their complexities (e.g., time dependencies) and their limitations (e.g., the number of time points is much smaller than the number of genes in a typical dataset). Existing methods for analysis of gene expression data are generally not well suited to time-course gene expression data. This dissertation has proposed several analysis methods for both inferring gene regulatory relationships and networks from time-course gene expression data.

For inferring gene regulatory relationships from time-course gene expression data, two dynamic model-based clustering methods are presented: MCM-based clustering and ARM-model based clustering. These two methods explore the time dependency feature within time-course gene expression profiles. Specifically, a Markov chain model is employed in MCM-based clustering and an autoregressive equation is employed in

ARM-based clustering in order to account for the dynamics of time-course gene expression data.

Computational experiments are performed on two datasets to validate the MCM-based clustering method. AARI is used as a measure of the quality of a clustering. Results show that the MCM-based clustering method outperforms the static model-based clustering methods. Likewise, the computational experiments are performed on four datasets to validate the ARM-based clustering methods. Two of these four datasets are those used for the MCM-based clustering method for the purpose of comparison between the MCM-based and ARM-based clustering methods. The results show that the ARM-based clustering methods outperform not only the static model-based clustering methods, but also the MCM-based clustering methods. The superior performance of the ARM-based clustering methods over the MCM-based clustering methods is perhaps due to the fact that useful information may be lost with the MCM-based clustering methods when the gene expression data are mapped onto three states (I, R, C). Our results support the conclusion that consideration of the dynamics of gene expression can improve the quality of clustering.

For inferring gene regulatory networks from time-course gene expression data, a state-space model without time delays is proposed. In this model, genes are viewed as observation variables, whose expression values depend on the current internal state variables and other external inputs, if they exist. The idea behind this view is that genes may be regulated by other elements in a cell. The BIC, the MLFA, and the multiple

regression method are employed to infer gene regulatory networks in terms of the proposed model from time-course gene expression data. The computational complexity of the algorithm to identify the model is much lower than that to identify competing models, such as Boolean network models and differential/difference equation models. Computational experiments are performed on two datasets. The results show that not only may gene regulatory networks be unambiguously inferred from current time-course gene expression datasets in terms of the state-space model without time delays, but the inferred networks have some features of real gene regulatory networks, such as stability and robustness.

As an improvement, a state-space model with time delays for inferring gene regulatory networks is proposed to account for the time-delayed relationships in real gene regulatory networks. The PPCA, the BIC, and the multiple regression method are proposed to identify the parameters of the proposed model. Computational experiments are performed on two datasets, where it is assumed that each regulatory interaction has two time delays (0 and 1) and time points in the datasets are enough to infer all time-delayed regulatory relationships. The results show that the inferred gene regulatory networks with time delays have improved prediction error and capture more features of real gene regulatory networks than the inferred gene regulatory networks without time delays.

In the state-space model with time delay, it is often assumed that each regulatory interaction has only one single time delay as there are not enough time points available in the datasets under consideration. The identification of time-delayed regulatory

relationships thus becomes very important. A genetic algorithm is proposed to infer the time-delayed relationships in the gene regulatory network from time-course gene expression datasets. Applications of the proposed GA to two gene expression datasets show that the GA can effectively infer the time-delayed relationships in gene regulatory networks. Furthermore, with optimal time-delayed relationships from the GA, the inferred gene regulatory networks with time delays have improved the prediction power and capture more features of real gene regulatory networks than the inferred networks without time delays, for example, stability, robustness, and periodicity.

5.2 Future Work

5.2.1 Improvement of Dynamic Model-Based Clustering

Some further improvements could still be made to both proposed dynamic model-based clustering methods. For example, the current MCM-based clustering method does not address the problems arising from missing data (which often occurs in gene expression experiments) and time delay in gene regulatory processes. It would be desirable to improve the proposed method using more sophisticated dynamic models for describing gene expression dynamics. One way to do this, for example, is using hidden Markov models (HMMs) (Rabiner, 1989).

For the ARM-based clustering method, gene expression dynamics are accounted for by the p order autoregressive equations. The time span between two consecutive

measurements of a time series may influence the complexity of parameter estimation of the equations. For gene expression datasets with equally-spaced measurements, the computational complexity of parameter estimation for autoregressive equations is linear as shown in this dissertation. However, for those gene expression datasets with unequally-spaced measurements, the problem becomes nonlinear. It is noted that a study by Ramoni et al. (2002) ignored the effects of the unequally-spaced measurements without giving any explanation. It is hypothesized that unequally-spaced time points should have some significant effect on modelling in general; the problem is similar, in nature, to the problem of time gaps in time-course gene expression datasets. Furthermore, at this point the selection criterion for the optimal order of autoregressive equations in ARM-based clustering has not been given, which calls for a further study. Some biological knowledge about datasets under consideration may help to choose an appropriate order for autoregressive models.

Since both proposed dynamic model-based methods have a probabilistic foundation, any prior information about genes may be incorporated into models (3.13) and (3.23) using Bayesian inference. Such *a priori* information may include gene sequence information, the cluster labels of a subset of genes, and so on. Incorporation of such information may be desirable to improve the quality of clustering further.

5.2.2 Improvement of the Inference of Gene Regulatory Network

Gene regulatory networks play a central role in systems biology. Many computational methods and models have been proposed for inferring gene regulatory networks from gene expression data. However, at present the inferred gene regulatory networks with these models can not completely explain complex organismal or suborganismal behaviours. On the other hand, any subjective assumptions-enforced models may result in misinterpreting organismal or suborganismal behaviours. Therefore, further studies are needed to improve the models and methods for inferring gene regulatory networks, specifically on the following points.

First, many models proposed previously for inferring gene regulatory networks, including those developed in this dissertation, are linear, but real gene regulatory networks may be nonlinear (Baldi and Hatfield, 2002; Gardner et al., 2003; Di Bernardo et al., 2004; Goutsias and Kim, 2004). With the current volume of gene expression datasets, many linear models are underestimated. The identification of the more complicated nonlinear models is even worse. However, in the framework of the state-space model, it is more possible to construct nonlinear dynamic equations for describing nonlinear regulatory relationships among the internal variables because the number of internal variables is smaller.

Second, one of the important tasks which challenge any effort on building models for cellular systems is the identification of biological implications of variables and / or clusters of variables. This is because one ultimate goal of systems biology is to develop methods to manipulate living cells to change and control their behaviours and states. As

shown by De Jong (2003), various computational models for gene regulatory networks have been proposed. These works encounter a general problem, the biological implications of the proposed models, to varying degrees. In order to give biological implications for the models proposed in this thesis, the following questions should be answered: Question 1 — what is the biological meaning of the internal variables? Question 2 — what are the internal variables from the perspectives of biology? Although seeking answers to these questions is beyond the scope of this thesis, the following are some general thoughts and approaches that may be followed to pursue the answers.

It has already been known from biology that regulatory interactions among genes are mediated by regulatory proteins encoded by genes (see Figure 2.1). Further, not all gene products (proteins) directly regulate gene expression in a gene regulatory network; only some of genes are translated into regulatory proteins, while others are translated into structural proteins (Albters et al., 1998; Liebler, 2002; Baldi and Hatfield, 2002). Therefore, the internal variables are likely to correspond to some regulatory proteins. Following this thought, there are two possible situations: (1) one protein corresponding to one internal variable, and (2) a cluster of proteins corresponding to one internal variable.

Hartemink et al.'s (2001) study may provide a support for the conjecture above. They proposed a Bayesian network model for gene regulatory networks. Their model introduced so-called latent variables to capture unobserved factors, and they used the

BIC method to select the latent variables. They considered the latent variables in their model to be known regulatory proteins with known regulatory relationships with genes.

As opposed to the situation considered in Hartemink et al.'s paper (2001), our models do not consider that the proteins corresponding to internal variables or their relationships with genes are known a priori. To investigate the relationships between the internal variables and the regulatory proteins, both gene expression data and corresponding protein expression data should simultaneously be collected in a biological process under consideration. This is possible as a number of techniques for measuring gene expression are available; see discussions in Section 2.1, and proteomics technologies such as mass spectrometry and two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) (Pennington and Dunn, 2001; Liebler, 2002) have also been employed to measure protein expression. As described in Chapter 4, from gene expression data, the expression profiles of internal variables can be estimated using BIC and either MLFA or PPCA. The correlations between each expression profile of the internal variables and each protein expression profile are calculated. If the correlation coefficient between one internal variable (say, IV) expression profile and one protein (say, P) expression profile is large enough, it can be conjectured that the internal variable IV corresponds to the protein P .

To identify a possible correspondence between the internal variable and the cluster of proteins, some supervised clustering techniques (Hartigan, 1975) can be employed. In this case, the internal variable expression profiles are viewed as the reference patterns

(supervisors). The correlation coefficient (see Section 3.1) can be employed to measure the similarity among the internal variable expression profiles and the protein expression profiles. The proteins whose expression profiles are similar to one specific internal variable expression profile are classified in the same group. As such, a group of proteins is established to correspond to an internal variable. It should be noted that time delay must be taken into account when calculating the similarity between an internal variable expression profile and a protein expression profile.

Finally, once an inferred gene regulatory network is built up with the proposed model and method, its accuracy needs to be verified. The quality of the inferred networks may be evaluated in two ways. One way is to develop more criteria from the perspective of bioinformatics to evaluate models for inferring large gene regulatory networks; e.g., validations involving mathematical, statistical, and simulation approaches. These approaches can give support to a prospective network model. However, only with wet-lab experiments can we actually say definitively if the model is wrong or is consistent with real life (i.e. correct). For example, some stimulus is applied to a real network, and gene expression values are then measured at a given series of time points. One feasible stimulus is to suddenly increase the amount of some transcripts, i.e., increase corresponding gene expression values (Gardner et al., 2003). On the other hand, gene expression values at the given series of time points are computed according to the inferred network with the values at the first time point of the experiment as the initial values. Then the computed gene expression values are compared to the corresponding experimental values. The smaller their differences are, the better the inferred network is.

REFERENCES

- Aitman T.J., et al. (1999) Identification of CD36 (Fat) as an insulin-resistant gene causing defective fatty acid and glucose metabolism in hypertensive rats. *Nature Genetics* **21**: 76-83.
- Akutsu T., et al. (1999) Identification of gene networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* **4**: 17-28.
- Akutsu T., Miyano S., and Kuhara S. (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Proceedings of the fourth Annual International Conference on Research in Computational Molecular Biology*, Tokyo Japan, pp: 8-14.
- Alberts B., et al. (1998) *Essential Cell Biology*. New York: Garland.
- Alter O., Brown P.O., and Botstein D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA* **97**: 10101-10106.

- Alter O., Brown P.O., and Botstein D. (2001) Processing and modeling genome-wide expression data using singular value decomposition. *Microarrays: Optical Technologies and Informatics* **4266**: 171–186.
- Alon U., et al. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed with oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* **96**: 6745–6750.
- Anderson T.W. (1984) An introduction to multivariate statistical analysis. New York: Wiley Press.
- Audic S. and Claverie J.M. (1997) The significance of digital gene expression profiles. *Genome Research* **7**: 986-995.
- Baldi P. and Hatfield G.W. (2002) *DNA Microarrays and Gene Expression*, Cambridge University Press.
- Barash Y. and Friedman N. (2001) Context-specific Bayesian clustering for gene expression data, in *Proceedings of the Fifth Annual International Conference on Research in Computational Molecular Biology*, Montreal, Canada, pp: 12-21.
- Batatgelj V. (1981) Note on ultrametric hierarchical clustering algorithms, *Psychometrika* **46**: 351-352.
- Baker F.J. (1974) Stability of two hierarchical grouping techniques Case I: Sensitivity to data errors. *Journal of the American Statistical Association* **69**: 440-445.

- Bertelsen A.H. and Velculescu V.E. (1998) High-throughput gene expression analysis using SAGE. *Drug Discovery Today* **3**: 152-158.
- Bertalanffy L.V. (1968) *General System Theory; Foundation, Development, Applications*. New York: Braziller.
- Breckenridge J.N. (1989) Replicating clustering analysis: method, consistency, and validity. *Multivariate Behaviour Research* **24**: 147-161, 1989.
- Bubin D.B. and Thayer D.T. (1982) EM algorithms fro ML factor analysis *Psychometrika* **47**: 69-76.
- Buchman T.G. (2002) The community of the self. *Nature* **420**: 246 – 251.
- Burnham K.P. and Anderson D.R. (1998) *Model Selection and Inference: a Practical Information-Theoretic Approach*. New York: Springer.
- Butte A.J. and Kohane I.S. (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing* **5**: 415-42.
- Calinski T. and Harabasz J. (1974): A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods A* **3**: 1-27.
- Chen C.T. (1999) *Linear System Theory and Design*. 3rd edition, New York: Oxford University Press.

- Chen G., et al. (2001) Cluster analysis of microarray gene expression data: application to and evaluation with NIA mouse 15K array on ES cell differentiation. *Statistica Sinica* **12**: 241-262.
- Chen T., He H.L., and Church G.M. (1999) Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing* **4**: 29-40.
- Cho R.J., et al. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell* **2**: 65-73.
- Chu S., et al. (1998) The transcriptional program sporulation in budding yeast. *Science* **282**: 699–705.
- Clark D.P. and Russell L.D. (2001) *Molecular Biology: Made Simple and Fun*, 2nd Edition. Vienna, IL: Cache River Press.
- Claverie J.M. (1999) Computational methods for the identification of differential and coordinated gene expression. *Human Molecular Genetics* **8**:1821–1832.
- Cunningham K.M. and Ogilvie J.C. (1972) Evaluation of hierarchical grouping techniques: A preliminary study. *The computer Journal* **15**: 209-213.
- De Jong H. (2002) Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology* **9**: 67-103
- Dembele D. and Kastner P. (2003) Fuzzy C-means method for clustering microarray data. *Bioinformatics* **19**: 937-980.

- Der S.D., et al., (1998) Identification of genes differentially regulated by interferon a, b, or g using oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* **95**: 15623–15628.
- DeRisi J., et al. (1996) Use of a cDNA microarray to analyze gene expression pattern in human cancer. *Nature Genetics* **14**: 457-460.
- D'haeseleer P., et al. (1997) Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. *Information Processing in Cells and Tissues*, Paton, R.C., and Holcombe, M. Eds., pp: 203-212.
- D'haeseleer P., et al. (1999) Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing* **4**: 41-52.
- D'haeseleer P., Liang S. and Somogyi R. (2000) Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **16**: 707-726.
- Di Bernardo D., Gardner T.S., and Collins J.J. (2004) Robust identification of large genetic networks. *Pacific Symposium on Biocomputing* **9**: 486-497.
- Dopazo J. and Carazo J.M. (1997) Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of Molecular Evolution* **44**: 226–233.
- Dopazo J. et al. (2001) Methods and approaches in the analysis of gene expression data. *Journal of Immunological Methods* **250**: 93-112.
- Dougherty E.R., et al. (2002) Inference from clustering with application to gene-expression micrarrays. *Journal of Computational Biology* **9**:105-126.

- Duda R.O., Hart P.E., and Stork D.G. (2001) *Pattern Classification*. New York: Wiley Press.
- Dudoit S. and Fridlyland J. (2002) A prediction-based resampling method for estimating the number of clustering in a dataset. *Genome Biology* **3**: research 0036.1-0036.21.
- Duggan D.J., et al. (1999) Expression profiling using cDNA microarrays. *Natural Genetics* **21**(Sup1): 10-14.
- Eisen M.B., et al. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc Natl. Acad. Sci. USA* **95**:14863-14868.
- Eisen M.B. and Brown, P.O. (1999) DNA arrays for analysis of gene expression. *Methods Enzymol* **303**: 179-205.
- Everitt B.S. and Dunn G. (1992) *Applied Multivariate Data Analysis*, Oxford University Press.
- Estivill-Castro V. (2002) Why so many clustering algorithms --- A position paper. *SIGKDD Explorations* **4**: 65-73.
- Fambrough D., et al. (1999) Diverse signaling pathways activated by growth factor receptors induce broadly overlapping, rather than independent sets of genes. *Cell* **97**: 727-741.
- Fields C., et al. (1994) How many genes in human genome? *Nature Genetics* **7**: 345-346.

- Filkov V., Skiena S., and Zhi J.Z. (2001) Analysis techniques for microarray time-series data, in *Proceedings of the Sixth Annual International Conference on Research in Computational Molecular Biology*, Montreal, Canada, pp: 124-131.
- Fislage R. (1998) Differential display approach to quantitation of environmental stimuli on bacterial gene expression. *Electrophoresis* **19**: 613-616.
- Fowlkes E.B. and Mallows C.L. (1983) A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* **78**: 553-569.
- Gardner T.S., Di Bernardo D., Lorenz D., and Collins J.J. (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* **301**: 102-105.
- Gasch A.P., et al. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell* **11**: 4241-4257.
- Gasch A. P. and Eisen M. B. (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology* **3**: research: 0059.1-0059.22.
- Ghosh D. and Chinnaiyan A.M. (2002) Mixture modeling of gene expression data from microarray experiments. *Bioinformatics* **18**: 275-286.
- Golub T.R., et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**: 531-537.

- Goodman L.A. and Kruskal W.H. (1954) Measure of association for cross classification. *Journal of the American Statistical Association* **49**: 732-764.
- Goutsias J. and Kim S. (2004) A Nonlinear Discrete Dynamical Model for transcriptional Regulation: Construction and Properties. *Biophysical Journal* **86**: 1922-1945.
- Hall I., et al. (1999) Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation* **3**: 103-112.
- Harrington C.A., Rosenow C., and Retief J. (2000) Monitoring gene expression using DNA microarrays. *Current Opinion in Microbiology* **3**: 285–291.
- Hartigan J.A. (1967) Representation of similarity matrices by tree. *Journal of the American Statistical Association* **62**: 1140-1158.
- Hartigan J.A. (1975) *Clustering Algorithms*. Wiley, New York, NY.
- Hartigan J.A. (1985) Statistical theory in clustering. *Journal of Classification* **2**: 63-76.
- Harvey A.C. (1993) *Time Series Models*, 2nd edition. Cambridge: The MIT Press.
- Hays W.L. (1973) *Statistics for the Social Sciences*, 2nd edition. New York: Holt, Rinehart and Winston, Inc.
- Heller R.A., et al. (1997) Discovery and analysis of inflammatory disease-related genes using cDNA microarrays. *Proc. Natl. Acad. Sci. USA* **94**: 2150–2155.

- Hartemink A.J. et al. (2001) Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing* **6**: 422-433.
- Herwig R., et al. (1999) Large-scale clustering of cDNA-fingerprinting data. *Genome Research* **9**: 1093–1105.
- Herrero J., Valencia A., and Dopazo J. (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics* **17**: 126-136.
- Heyer L.J., et al. (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Research* **9**: 1106-1115.
- Holter N.S., et al. (2001) Dynamic modeling of gene expression data. *Proc. Natl. Acad. Sci. USA* **98**:1693-1698.
- Holstege F.C.P., et al. (1998) Dissecting the regulatory circuitry of a eukaryotic genome. *Cell* **95**: 7171-728.
- Hubert L. and Arabie P. (1985) Comparing partitions. *Journal of Classification*. **2**: 193-218.
- Hwang D., et al. (2002) Determination of minimum sample size and discriminatory expression patterns in microarray data. *Bioinformatics* **18**: 1184-1193.
- Iyer V.R., et al. (1999) The transcript-ional program in the response of human fibroblasts to serum. *Science* **283**: 83-87.

- Jain A.K. and Dubes R.C. (1988) *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall.
- Kaufman L. and Rousseeuw P.J. (1990) *Finding Groupings in Data: An Introduction to Cluster Analysis*. New York: Wiley Press.
- Kauffman S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford: Oxford University Press.
- Kedem B. and Fokianos K. (2002) *Regression Models for Time Series Analysis*. Hoboken, N.J. : Wiley-Interscience.
- Kohonen T. (1997) *Self-Organizing Maps, Second Edition*, Berlin: Springer-Verlag.
- Kreiger A.K. and Green P.E. (1999) A generalized Rand-index method for consensus clustering of separate partitions of the same data base. *Journal of Classification* **16**: 63-89.
- Krzanowski W. and Lai Y. (1985) A criterion for determining the number of group in a dataset using sum of squares clustering. *Biometrics* **44**: 23-34.
- Kwon A.T., Hoos H.H., and Ng R. (2003) Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics* **19**: 905-912.
- Lance G.N. and Williams W.T. (1967) A general theory of classificatory sorting strategies: 1. Hierarchical systems. *The Computer Journal* **9**: 373-380.

- Lander E.S., et al. Initial sequencing and analysis of the human genome. *Nature* **409**: 860–921.
- Langmead C.J., et al. (2002) Phase-independent rhythmic analysis of genome-wide expression patterns, in *Proceedings of the Sixth Annual International Conference on Research in Computational Molecular Biology*, Washington DC, USA, pp: 205-215.
- Lawley D.N. and Maxwell A.E. (1971) *Factor Analysis as a Statistical Method*. 2nd edition. New York: American Elsevier Pub. Co.
- Laub M.T., et al. (2000) Global analysis of the genetic network controlling a bacteria cell cycle. *Science* **290**: 2144-2148.
- Lee M.L.T., et al. (2000) Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations, *Proc Natl. Acad. Sci. USA* **97**: 9834-0839.
- Lee C.K., et al. (1999) Gene expression profile of aging and its retardation by caloric restriction. *Science* **285**: 1390-1393.
- Legendre P. and Legendre L. (1998) *Numerical Ecology, Second English Edition*. Elsevier Science B. V.: Amsterdam.
- Liang P. and Pardee A.B. (1992) Differential display of eukaryotic messenger RNA by means of the polymerase chain reaction. *Science* **257**: 967-971.
- Liang P., et al. (1994) Differential display using one-based anchored oligodT primers. *Nucleic Acids Research* **22**: 5763-5764.

- Liang S., et al. (1998) REVEAL, A general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing* **3**: 18-29.
- Liebler D. C. (2002) *Introduction to Proteomics*. Totowa, NJ: Humana Press.
- Lockhart D.J., et al. (1996) Expression monitoring by hybridization to high density oligonucleotide arrays. *Nature Biotechnology* **14**: 1675-1680.
- Madden S.L., et al. (1997) SAGE transcript profiles of p53-dependent growth regulation. *Oncogene* **15**: 1079-1085.
- Mcgall G., et al. (1996) Light-directed synthesis of high-density oligonucleotide arrays using semiconductor photoresists. *Proc Natl. Acad. Sci. USA* **93**: 13555-13560.
- McLachlan G.J., et al. (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics* **18**: 413-422.
- Michaels G.S., et al. (1998) Cluster Analysis and Data Visualization of Large-Scale Gene Expression Data. *Pacific Symposium on Biocomputing* **3**: 42-53.
- Moch H., et al. (1999) High-throughput tissue microarray analysis to evaluate genes uncovered by cDNA microarray screening in renal carcinoma. *American Journal of Pathology* **154**: 981-986.
- Morgan B.J.T. and Ray A.P.G (1995) Non-uniqueness and inversion in clustering analysis. *Applied Statistics* **44**: 117-134.

- Pan W., Lin J., and Le C.T. (2002) How many replicates of arrays are required to detect gene expression changes in microarray experiment? A mixture model approach. Department of Biostatistics, University of MN, Technical Report.
- Pan W., Lin J., and Le C.T. (2002) Model-based cluster analysis of microarray gene-expression data, *Genome Biology* **3**: research0009.1-0009.8.
- Patten C., et al. (1998) Identification of two novel diurnal genes by screening a rat brain cDNA library. *Neuroreport* **9**: 3935-3941.
- Pease A.C., et al. (1994) Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proc. Natl. Acad. Sci. USA* **91**: 5022-5026.
- Pennington S.R. and Dunn M.J. (2001) *Proteomics from Protein Sequence to Function*. Oxford: BIOS Scientific Published Limited.
- Perou C.M, et al. (1999) Distinctive gene expression patterns in human epithelial cells and breast cancers. *Proc Natl. Acad. Sci. USA* **96**: 9212-9217.
- Polyak K., et al., (1997) A model for p53 induced apoptosis. *Nature* **389**:300-305.
- Press W. H., et al. (1992) *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition. Cambridge, UK: Cambridge University Press.
- Rabiner L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE* **77**: 257-285.

- Raftery A.E. (1986) Choosing models for cross-classification. *American Sociological Review* **51**: 145-146.
- Rand W.M. (1971) Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**: 846-850.
- Renner C., et al. (1998) differential mRNA display at the single cell level. *BioTechniques* **24**:720-724.
- Ramoni M.F., et al. (2002a) Bayesian clustering by dynamics. *Machine Learning* **47**: 91-121.
- Ramoni M.F., et al. (2002b) Cluster analysis of gene expression dynamics. *Proc. Natl. Acad. Sci. USA*, **99**: 9121-9126.
- Rogge L., et al. (2000) Transcript imaging of the development of human T helper cells using oligonucleotide arrays. *Nature Genetic* **25**: 96–101.
- Rousseeuw P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**: 53-65.
- Rosenfeld N and Alon U. (2003) Response delays and the structure of transcription networks. *Journal of Molecular Biology* **329**: 645-654.
- Saha S., et al. (2002) Using the transcriptome to annotate the genome. *Nature Biotechnology* **20**: 508-512.

- Schwarz G. (1978) Estimating the dimension of a model. *Annals of Statistics* **6**: 461-464.
- Schena M., et al. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270**: 467-470.
- Schena M., et al. (1996) Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA* **93**: 10614–10619.
- Sherlock G., et al. (2001) The Stanford Microarray Database. *Nucleic Acids Research* **29**: 152-155.
- Shiue L. (1997) Identification of candidate genes for drug discovery by differential display. *Drug Development Research* **41**: 142-159.
- Singh-Gasson S., et al. (1999) Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array. *Nature Biotechnology* **17**: 974-978.
- Sokal R.R. and Rohlf F.J. (1962) The comparison of dendrograms by objective methods. *Taxon* **11**: 33-39.
- Somogyi R. and Sniegowski C.A. (1996) Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation. *Complexity* **1**: 45-63.
- Spellman P.T., et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, *Molecular Biology of the Cell* **9**: 3273-3297.

- Syed F., et al. (1999) CCR7 (EBI 1) receptor down-regulation in asthma: differential gene expression in human Cd4+ T lymphocyte. *QJM: An International Journal of Medicine* **92**: 463-471.
- Tamayo P., et al. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci.* **96**: 2907-12.
- Tavazoie S. et al. (1999) Systematic determination of genetic network architecture. *Nature Genetics* **22**: 281-285.
- Teague T.K., et al. (1999) Activation changes the spectrum but not the diversity of genes expressed by T cells. *Proc. Natl. Acad. Sci. USA* **96**: 12691–12696.
- Theodoridis S. and Koutroumbas K. (1999) *Pattern recognition*. San Diego: Academic Press.
- Tibshirani R., Walther G. and Hastie T. (2000) Estimating the number of clusters in a dataset via the gap statistic. Technique report, Department of Biostatistics, Stanford University.
- Tipping M.E. and Bishop C.M. (1999) Probabilistic principal component analysis. *Journal of the Royal Statistic Society, Series B* **61**: 611-622.
- Torkkola K. (2001) Self-organizing maps in mining gene expression data. *Information Sciences* **139**: 79-96.

- Toronen P., et al. (1999) Analysis of gene expression data using self-organizing maps. *FEBS Letter* **451**: 142-146.
- Vardeman S.B. (1994) *Statistics for Engineering Problem Solving*. Boston: PWS Pub. Co.
- Velculescu V.E., et al. (1995) Serial analysis of gene expression. *Science* **27**: 484-487.
- Velculescu V.E., et al. (1997) characterization of the yeast transcriptome. *Cell* **88**: 243-251.
- Wang X. and Feuerstein G.Z. (1997) The use of mRNA differential display for discovery of novel therapeutic targets in cardiovascular disease. *Cardiovascular Research* **35**: 414-421.
- Weaver D.C., Workman C.T. and Stormo G.D. (1999) Modeling regulatory networks with weight Matrices. *Pacific Symposium on Biocomputing* **4**: 112-123.
- Webb G.C., et al. (2000) Expression profiling of pancreatic b cells: glucose regulation of secretory and metabolic pathway genes. *Proc. Natl. Acad. Sci. USA* **97**: 5773–5778.
- Wen X., et al. (1998) Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA* **95**: 334-339.
- Wessels L.F.A, Van Someren E.P., and Reinders M.J.T. (2001) A comparison of genetic network models. *Pacific Symposium on Biocomputing* **6**: 508-519.

- White J.A and Petkovich M. (1998) Identification and cloning of RA-regulated genes by mRNA differential display. *Methods in Molecular biology* **89**: 389-404.
- Whitfield M.L., et al. (2002) Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular Biology of the Cell* **13**: 1977-2000.
- Wiener N. (1948) *Cybernetics: Or Control and Communication in the Animal and the Machine*. Cambridge: Technology Press.
- Wodicka L., et al. (1997) Genome-wide expression monitoring in *Saccharomyces cerevisiae*. *Nature Biotechnology* **15**: 1359-1367.
- Wu F.X., Zhang W.J., and Kusalik A.J. (2004a) Modelling gene expression from microarray expression data with state-space equation (proceedings paper with oral presentation). *PSB2004*: 581-592.
- Wu F.X., Zhang W.J., and Kusalik A.J. (2004b) Model-based clustering with genes expression dynamics for time-course gene expression data. *BIBE2004*: 267-274.
- Wu F.X., Zhang W.J., and Kusalik A.J. (2004c) Dynamic model-based clustering for time-course gene expression data. Submitted to *Journal of computational biology and Bioinformatics*, May 2004.
- Wu F.X., Zhang W.J., and Kusalik A.J. (2004d) State-space model for gene regulatory networks with time delays. Accepted by *CSB2004*, June 2004.

- Wu F.X., Zhang W.J., and Kusalik A.J. (2004e) State-space model with time delays for gene regulatory networks. Accepted by *Journal of Biological Systems*.
- Wu F.X., Kusalik A.J., and Zhang W.J. (2004f) Genetic algorithm for inferring time delays in gene regulatory networks. Accepted by *CSB2004*, June 2004.
- Wu F.X. (2003) Analysis and Modeling of Gene Expression Data from DNA Microarray Experiments. Technical Report, Division of Biomedical Engineering, University of Saskatchewan, Oct, 2003.
- Wuensche A. (1998) Genomic regulation modeled as a network with basins of attraction. *Pacific Symposium on Biocomputing* **3**: 89-102.
- Xiong M. et al. (2001) Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism* **73**: 239-247.
- Yershov K., et al. (1996) DNA analysis and diagnostics on oligonucleotide microchips. *Proc Natl. Acad. Sci. USA* **96**: 4913-4918.
- Yeung K.Y., et al. (2001) Model-based clustering and data transformations for gene expression data. *Bioinformatics* **17**: 977-987.
- Yildirim N. and Mackey M.C. (2003) Feedback regulation in the lactose operon: A mathematical modeling study and comparison with experimental data *Biophysical Journal* **84**: 2841-2851.
- Zhang L., et al. (1997) Gene expression profiles in normal and cancer cells. *Science* **276**: 1268-1272.

- Zhang M.Q. (1999) Large-scale gene expression data analysis: A new challenge to computational biologists. *Genome Research* **9**: 681-688.