2005

Multi-scale genetic network inference based on time series gene expression profiles

Pan Du
Iowa State University

Multi-scale genetic network inference based on time series gene expression profiles

by

Pan Du

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Bioinformatics and Computational Biology; Electrical Engineering

Program of Study Committee:
Julie A. Dickerson, Co-major Professor
Eve Syrkin Wurtele, Co-major Professor
Nicola Elia
Xun Gu
Yao Ma

Iowa State University

Ames, Iowa

2005

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Pan Du

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

For the Co-major Program

Signature was redacted for privacy.

For the Co-major Program

DEDICATION

To my beloved wife, Zhaomin Huang, my parents, Xingen Du and Xiuzhen Chen and my brothers for their support, encouragement and love.

谨以此论文

献给我亲爱的妻子、父母和兄长。

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This work integrates multi-scale clustering and short-time correlation to estimate genetic networks with different time resolutions and detail levels. Gene expression data are noisy and large scale. Clustering is widely used to group genes with similar pattern. The cluster centers can be used to infer the genetic networks among these clusters. This work introduces the Multi-scale Fuzzy K-means clustering algorithm to uncover groups of coregulated genes and capture the networks in different levels of detail.

Time series expression profiles provide dynamic information for inferring gene regulatory relationships. Large scale network inference, identifying the transient interactions and feedback loops as well as differentiating direct and indirect interactions are among the major challenges of genetic network inference. Time correlation can estimate the time delay and edge direction. Partial correlation and directed-separation theory help differentiate direct and indirect interactions and identify feedback loops. This work introduces the constraint-based time-correlation (CBTC) network inference algorithm that combines these methods with time correlation estimation to more fully characterize genetic networks. Gene expression regulation can happen in specific time periods and conditions instead of across the whole expression profile. Short-time correlation can capture transient interactions.

The network discovery algorithm was mainly validated using yeast cell cycle data. The algorithm successfully identified the yeast cell cycle development stages, cell cycle and negative feedback loops, and indicated how the networks dynamically changes over time. The inferred networks reflect most interactions previously identified by genome-wide location analysis and match the extant literature. At detailed network level, the inferred networks provide more detailed information about genes (or clusters) and the interactions among them. Interesting genes, clusters and interactions were identified, which match the literature and the gene ontology information and provide hypotheses for further studies.

CHAPTER 1. INTRODUCTION

1.1 Background

Exploring how DNA enables life is the main topic in biology. It is widely believed that thousands of genes and their products (i.e., RNA and proteins) in a given living organism function in a complicated and orchestrated way that enables life. Figure 1-1 shows the framework of this process. Genes are pieces of DNA sequences which encode how and when to make proteins. Genes are first transcribed as mRNA (messenger RNA), then translated as polypeptides (proteins) (the splicing process after transcription in Eukaryotes is omitted for simplicity). Proteins perform most essential life functions. They may function alone, but most often in the form of protein complexes. Proteins are essential to the structure and function of all living cells and virus. They interact with each other or through complex, interconnect pathways (such as signal transduction, regulatory and metabolic pathways) to make cells come alive. These living cells interact with each other to form communities of cells or living life.

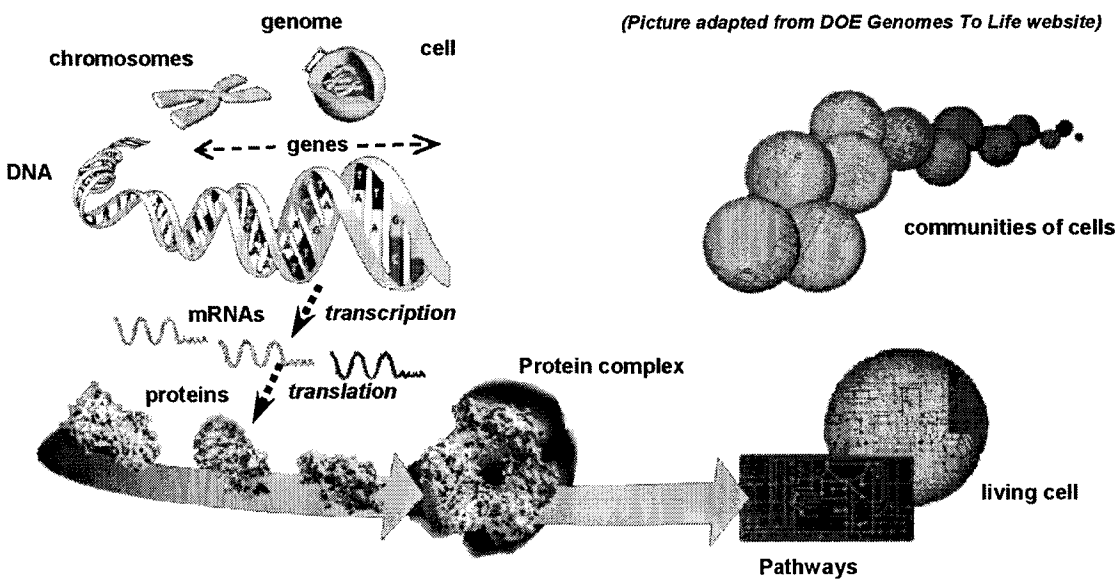


Figure 1-1 From DNA to working cells

Traditional molecular biology typically focuses on a single gene, protein, reaction or pathway, and follows a reductionist approach to studying the biological system. Over the years, this practice has led to remarkable achievements. However, biological processes are inherently integrated and interactive, so traditional studies cannot resolve the complex relationships among biological entities. Numerous examples show that the manipulation of a key enzyme in a biological pathway does not lead to the expected effects (Bailey 1999). This may happen because the biological processes are inherently integrated, and the intended effects are compensated by gene regulation. Such examples indicate the importance of studying the entities together as integrated networks. With the help of new high throughput technologies, this becomes possible. Systems biology, an emerging field, is the exploration of life at the ultimate level of the whole organism instead of single genes or proteins. It endeavors to quantify all of the molecular elements of a biological system to assess their interactions and to integrate that information into graphical network models that serve as predictive hypotheses to explain emergent behaviors (Kitano 2002; Hood, Heath et al. 2004).

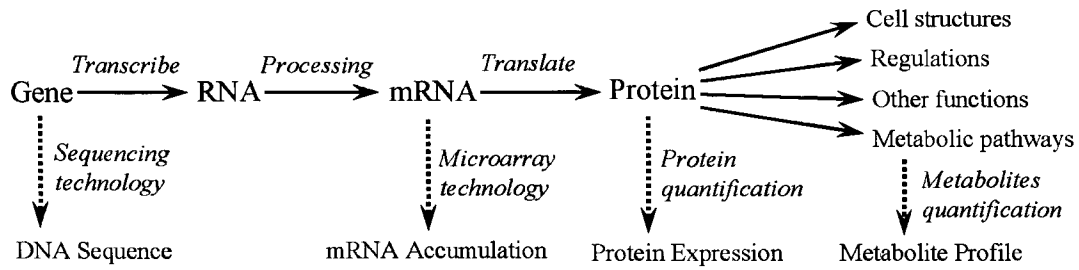


Figure 1-2 Gene expression process and new high throughput technology

Figure 1-2 shows gene expression process and associated major high throughput technologies. Genes are first transcribed as RNAs, RNAs become mRNAs(messenger RNA) after processing, then mRNAs are translated and processed as proteins. Proteins take roles in constituting cell structures, regulating cellular processes, catalyzing biochemical reactions in metabolic pathways, and performing other functions. The gene expression processes depend on various factors not depicted above, which include chromosomal activation or deactivation, control of transcription initiation, processing of RNA (like splicing), RNA transport, mRNA degradation, initiation of translation and post-translational modifications. All these processes

are regulated by proteins and other entities. Therefore, gene expression is a very complex and coordinated process.

High-throughput technologies make studying gene expression processes at the system level possible. Sequencing technology resolves the exact sequence of nucleotides (A, C, G, T) of the genome. The complete genomic sequences of many organisms including yeast, *Arabidopsis* and human, have been determined. DNA sequences of many genes have also been identified. Microarray technology measures mRNA accumulation levels of tens of thousands of genes in parallel. It gives a snapshot of the mRNA accumulation levels at a specific time and condition. Proteomics quantifies the protein of the corresponding genes and gives information about protein complexes and protein modifications. Metabolites are the end products of the gene expression. While mRNA accumulation and protein accumulation do not tell the whole story of what might be happening in a cell, metabolite profiling can give an instantaneous 'snapshot' of the physiology of that cell. Among these high-throughput data, gene sequence and microarray data are most widely available and used. There are a quickly growing number of public databases of genome sequence and microarray data for many different species. Details of microarray technology will be illustrated in Chapter 2. Protein and metabolite profiling technologies are still emerging, data sets in these fields are also increasing in number. The integration of these data to resolve complex relationships among biological entities and to infer a more complete picture of living organisms is a big challenge to scientists and the major task in systems biology.

In this work, our focus is to infer the genetic networks mainly based on microarray data. The genetic networks reflect how genes interact with each other. The microarray data (mRNA accumulation) are collected under different times and conditions, like temperature and mutation. The patterns of expression profiles reflect the internal mechanisms of gene interactions. Therefore, specific mathematic models and algorithms are developed to infer gene interactions based on the expression profiles. Regulatory sequence information is also combined to resolve the detailed transcription regulatory relationships between the TF (Transcription Factor) and corresponding genes.

1.2 Overview of genetic network inference

1.2.1 Genetic network inference based on expression profiles.

Genes having similar functions or participating in related cellular processes are usually coregulated. Their expression profiles share similar patterns. In large scale network inference, clustering is usually used to find the coregulated genes, then genetic networks are constructed based on the cluster centers. An introduction to clustering algorithms will be given in Chapter 2. (D'Haeseleer, Liang et al. 1999; D'Haeseleer, Liang et al. 2000) review regulation inference from clustering of gene expression data.

In order to infer the genetic networks from expression profiles, first we need to define mathematic models to reflect the internal mechanism of gene expression. Different genetic network models, such as Boolean networks, linear models, differential equations, stochastic models and Bayesian models, have been proposed. Each model has its best suited applications (see Chapter 3). (Bolouri and Davidson 2002; van Someren, Wessels et al. 2002) give a thorough comparative review of genetic network modeling algorithms. (de Jong 2002) presents a similar study with emphasis on simulation. (Kaern, Blake et al. 2003) review the gene network engineering from a combined experimental and modeling perspective. Although much research has been conducted; many problems remain unsolved, and many solutions are quite primitive. Biotechnology is developing very quickly, as more data and results become available, new challenges will continue to appear.

Genetic network inference can be performed in different ways with different kinds of data sets and information. Apart from using gene expression data, genetic network inference can be based on regulatory sequence analysis, gene perturbation analysis, constraint based analysis and so on. In order to obtain an overview of this field, we present a brief introduction to these methods.

1.2.2 Regulatory sequence analysis

The genome encodes two major types of information: genes and cis-regulatory elements. The cis-regulatory elements, together with transcription factors, regulate the levels of expression of individual genes. The promoter containing most cis-regulatory elements is

located upstream of gene coding region. Figure 1-3 illustrates the regulation of gene transcription (Kamvysselis 2003). In order to induce or repress transcription of gene GAL1, Transcription Factors (TF) like Mig1 and Gal4 bind to specific cis-regulatory sequences. These binding sequences have specific patterns (over-represented motifs), and the TFs can recognize these patterns and bind preferentially to them. For example, CCCCW is a motif for TF Mig1, where W can be . A variety of molecular technologies, such as CHiP chip analysis, are used to identify cis-regulatory motifs.

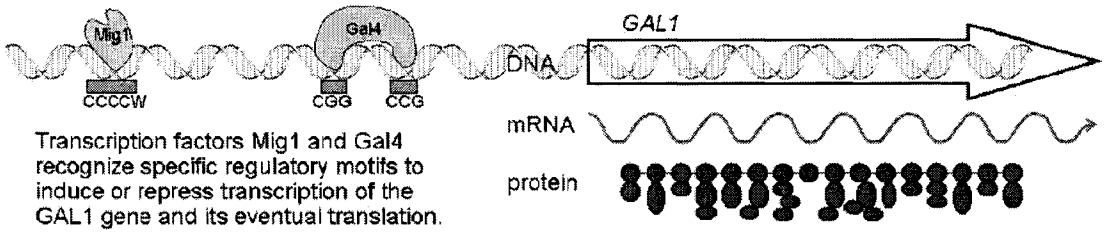


Figure 1-3 Regulation of gene transcription. (Picture adapted from (Kamvysselis 2003))

Because coregulated genes may be regulated by the same TFs, motifs corresponding to the TFs should exist at the upstream regions of all coregulated genes. So, it is possible to identify regulatory motifs by searching the over-represented sequence patterns upstream of coregulated genes. We can further infer possible transcription factors based on the over-represented patterns. On the other hand, from the motifs of some given transcription factors, we can search the possible locations of binding sites by pattern matching methods. Therefore, regulatory sequence analysis provides another approach to determine the gene regulatory relationships at the transcription level, based on the genome sequence. A lot of work has been or is being done in this area (Segal, Shapira et al. 2003; Segal, Yelensky et al. 2003). There are also online regulatory sequence analysis tools available (van Helden 2003). More detailed description of regulatory sequence analysis methods and integration with genetic network inference will be provided in Chapter 7.

1.2.3 Gene perturbation or over-expression analysis

Gene perturbation or over-expression means artificially mutating or over-expressing one or several specific genes. It can provide a rich variety of different gene expression profiles. Comparing them to the wild type gene expression profile, we can determine the functions of

the perturbed genes and their relationships to other genes. After a series of gene perturbations, we can ascertain the coarse gene network structure. (de la Fuente, Brazhnik et al. 2002; Wagner 2002; Tegner, Yeung et al. 2003) provide methods to design perturbation experiments and algorithms to integrate the results with network modeling. The problem of gene perturbation is that there is a huge number of combinational possibilities of connection. The design of an efficient gene perturbation experiment is still an open problem. In addition, the large expense of such experiments encourages the view that gene perturbation and over-expression are more suitable for the evaluation of results or the inference of networks in a small scale.

1.2.4 Integrating with other prior knowledge

The traditional study of molecular biology has led to remarkable achievements. Many results and much prior knowledge are available. Some of the results are organized in a database and represented in a computer-interpretable way. Gene Ontology (GO: http://www.geneontology.org) is a shared, controlled vocabulary which is being developed to cover all organisms. GO is organized into three categories: molecular function (MF), biological process (BP), and cellular component (CC). Tools for literature mining can be used to retrieve many recent publications. Other public databases of metabolic pathways, protein and protein interactions are also available. Integrating prior knowledge with experimental data for genetic network inference will definitely improve the performance. However, the efficient integration of this knowledge is still an open problem.

1.3 Challenges of genetic network inference

Gene expression network inference is not an easy task. As shown in Figure 1-2, each step of gene expression is regulated by different types of proteins and other factors. For example, the gene transcription by RNA polymerase is regulated by at least three types of proteins: specificity factors, repressors and activators. Other factors like temperature, other environmental stimuli and some metabolites will also affect gene transcription. Therefore, regulation of gene expression is a very complex process. The difficulties of genetic network inference include: (1) Inference of the regulatory networks based on microarray data means

treating other entities, such as proteins and metabolites, as hidden variables. This will produce uncertain results. (2) It is required to consider the combinatorial nature of gene regulation (one gene might be regulated by multiple gene products). (3) The number of measurements (arrays) is very limited compared to the large number of objects (genes). This is true especially for complex models and large-scale networks. For example, there are 13,600 genes for fruit fly, 20,000 to 25,000 for Human, 27,000 for *Arabidopsis* and 45,000 for rice, but the number of samples is very limited for a specific experiment (usually only several or tens of samples). To fit a network model, the number of samples should be at least comparable with the number of parameters in the model. (4) The gene expression measurements are noisy, due to variations among different individuals, low quantities of some RNAs and measurement errors. Also, data from different experiments may not be directly comparable. (5) The gene interactions may happen within specific time periods and conditions instead of across the whole expression profiles. Catching these transient interactions is challenging. (6) The data may be under sampled. Some fast changing information may not be captured. (7) The exact mechanisms of regulatory interactions are usually unclear. (8) It is challenging to integrate prior knowledge or to resolve conflictions during network inference. Much more research is needed in this area.

1.4 Proposed solutions and contributions of this work

Based on previous work, we propose or improve methods for the inference of genetic networks, with special focus on time series gene expression data.

In order to reduce the network scale, we first group the genes with similar expression patterns by clustering. Since the degree of coregulation of different groups of coregulated gene could vary widely, a new Multi-scale Fuzzy K-means clustering algorithm is proposed to discover groups of coregulated genes with different degrees of coregulation. Details will be shown in Chapter 4. Then, we perform genetic network inference based on the cluster centers instead of individual genes. We adopt a linear model with time delay during the network inference. Instead of fitting the linear model directly to experiment data, we use pair-wise correlation or time correlation (for time series data) to detect the linear relationships between expression profiles. This greatly reduces the requirement of the profile

length. Direct and indirect interactions are differentiated by d-separation and partial correlation theory with the combining of time delay and edge direction information. Details will be given in Chapter 5. The prior knowledge expressed in GO (Gene Ontology) is integrated during network inference and evaluation. In order to catch the transient interactions, we propose the use of short-time correlation. The results show when the interactions happen and how the interaction strength changes over time frames (see Chapter 6). Finally, we integrate the genetic networks together with regulatory sequence analysis to refine and interpret the network.

Arabidopsis data, simulated data and yeast cell cycle data are used in network inference. Results show the proposed algorithms are effective. Many identified interactions match literature results and provide hypotheses for future research. Comparisons with other network inference algorithms and models are also made. For details, please refer to chapters 4 to 7.

Major contributions of this work

- A Multi-scale Fuzzy K-means clustering algorithm is proposed to discover groups of coregulated genes in a user controllable way.
- A constraint-based time-correlation (CBTC) network inference algorithm is proposed, which integrates time correlation with d-separation and partial correlation theories to differentiate direct and indirect interactions and identify feedback cycles;
- A short-time correlation algorithm is proposed to catch the transient interactions and show the dynamic changes of network topologies. Detail dynamic interaction is captured by visualizing the short-time correlation coefficients over different parameter dimensions;
- Network inference algorithm with multi-scale resolution is developed;
- These ideas are integrated with analysis of regulatory motifs; more regulatory sequence motifs can be identified by using the Multi-scale Fuzzy K-means clustering algorithm.

1.5 Organization of the report

In Chapter 2, we give a description of preprocessing and clustering of microarray data. In Chapter 3, an introduction to genetic network inference is given. In Chapter 4, we describe using fuzzy logic in genetic network inference, and propose a new Multi-scale Fuzzy K-

means clustering algorithm designed for network inference. In Chapter 5, we describe the genetic network inference based on time series expression profiles. An algorithm for differentiating direct and indirect interactions is proposed. In Chapter 6, we propose a network inference algorithm to catch the transient interactions by using short-time correlation. In Chapter 7, we propose the multi-scale genetic network inference by combining the algorithms proposed in chapters 4 and 5. Regulatory sequence analysis is also integrated with the network inference. Finally conclusions are made in Chapter 8.

CHAPTER 2. PREPROCESSING AND CLUSTERING OF MICROARRAY DATA

2.1 Microarray technology and preprocessing

2.1.1 Microarray Technology

Microarray technology tries to monitor tens of thousands of genes or even the whole genome on a single chip. Terminologies that have been used in the literature to describe this technology include biochip, DNA chip, DNA microarray, gene array. GeneChip®, trademark owned by Affymetrix, Inc., refers to its high density, oligonucleotide-based DNA arrays. The underlining principle of microarray technology is base-pairing (i.e., A-T and G-C for DNA; A-U and G-C for RNA) or hybridization. Microarray chips are fabricated by high-speed robotics, generally on glass but sometimes on nylon substrates. Probes with known identity are planted on the chips in very high density, and used to determine complementary binding. The expression of each gene is reflected by the accumulation level of the corresponding mRNA. There are two major application forms of microarray technology: (1) Identification of sequence (gene / gene mutation); (2) Determination of expression level (abundance) of genes. In genetic network inference, the microarray is used to measure the gene expression levels.

There are two variants of the microarray technology:

The first method is traditionally called cDNA microarray, or spotted microarray. For cDNA microarray, probe cDNA (500~5,000 bases long) is immobilized to a solid surface such as glass using robot spotting and exposed to a set of targets either separately or in a mixture. Usually two samples, dyed with different dyes (Cyanine 3 and Cyanine 5), are hybridized to a single slide. One of the samples is treated as reference. The dyes fluoresce at different wavelengths, so it is possible to get separate images for each dye. The color strength of each spot image on the microarray slide reflects the mRNA accumulation level of the particular gene corresponding to the spot probe. The ratio of the color strength of two dyes reflects the relative change of mRNA accumulation levels between the sample and the

reference sample. Data analysis of cDNA microarray data is usually based on the color strength ratios of the two dyes.

The second method, historically called DNA chips, was developed at Affymetrix, Inc. It is also called Affymetrix GeneChips. For this method, an array of oligonucleotide (20~80-mer oligos) or peptide nucleic acid (PNA) probes is synthesized either in situ (on-chip) or by conventional synthesis followed by on-chip immobilization. The array is exposed to labeled sample DNA, hybridized, and the identity/abundance of complementary sequences is determined. Unlike cDNA microarray, Affymetrix only use one sample during hybridization, and the color strength of the dye reflects the relative level of mRNA accumulation. The manufacture and design of Affymetrix chips is more complex than cDNA mciroarrays. For detailed information, please visit www.affymetrix.com.

2.1.2 Preprocessing and normalization

One major preprocessing step of microarray data is the log transformation. There are several reasons for this. Firstly, the log transformed values are more biologically interpretable. In biology, people are more interested in the fold change instead of the absolute change of expression values. After log transformation, the fold change values will be linear and more easily interpretable. Secondly, after log transformation the distribution of data values will be approximately symmetric and normal.

When analyzing microarray data, it is important to remove the sources of non-biological variations among the arrays. Because each experiment is conducted on a chip at particular conditions, RNA levels may fluctuate a lot from chip to chip due to some uncontrollable non-biological elements. Sources of non-biological variation include dye bias, differences in the amount of labeled cDNA hybridized to each channel in a microarray experiment, variation across replicate slides, variation across hybridization conditions, variation in scanning conditions, variation among technicians doing the lab work and other uncontrollable affects. Normalization is a process for removing these non-biological fluctuations. It is an important step and may directly affect the results of further processing. Many algorithms have been proposed. In general, there are several approaches which can be used separately or in combination to normalize a set of microarrays.

1. Multiply each array by a constant to make the mean (median) intensity the same for each individual array.
2. Adjust the arrays using some control or housekeeping genes that we would expect to have the same intensity level across all of the samples.
3. Match the percentiles of each array.
4. Adjust using a nonlinear smoothing curve, like "Lowess" curve.

The normalization processes of cDNA and Affymetrix have some differences. For cDNA microarray, additional normalization of the dye bias between two dyes should be considered. "Lowess"(LOcally WEighted polynomial regreSSion) normalization usually is adopted to handle intensity-dependent dye bias (Yang, Dudoit et al. 2002). For Affymetrix data, two types of normalization methods are widely adopted, which are Microarray Analysis Suite(MAS) 5.0 Signal proposed by Affymetrix and Robust Multichip Average (RMA) (Irizarry, Hobbs et al. 2003). MAS 5.0 Signal is a systematic normalization process, which includes background adjustment, ideal mismatch computation and signal log value computation. All these computations are done separately for each chip. RMA is a normalization process emphasizing in the statistical view. RMA only uses the PM (Perfect Match) values of the probes. It adopts a quantile normalization process across all Affymetrix GeneChips, perform median polish separately for each probe set with rows indexed by GeneChip and columns indexed by probe id, and finally use the estimated row effects as probe-set specific expression measures for each GeneChip. For some data sets, the difference between two normalization methods is obvious. Whether one method is better than another one is still controversial.

2.2 Overview of widely used clustering algorithms

Clustering analysis has been widely used in many fields, such as engineering, economy, medical, biology and etc. Clustering is appropriate when there is no a priori knowledge about the data set. It is an unsupervised learning process. The purpose of clustering is to group elements (genes) with highly similarity together and separate elements (genes) with low similarity apart.

2.2.1 Distance metrics

Before selecting clustering algorithm, we need to determine which distance (or similarity) metric should be adopted. Widely used distance metrics include Euclidean, City block, Mahalanobis, cosine, correlation. Some people also use mutual information as a distance metric. Different distance metrics measure different features between expression profiles. For microarray data, the most widely used distance metric is Pearson correlation (correlation coefficient) distance.

The Pearson correlation coefficient between any two vectors $x=(x_1,x_2,...,x_n)$ and $y=(y_1,y_2,...,y_n)$ is defined as:

$$r_{xy}=\frac{1}{n}\sum_{i=1,n}\left(\frac{x_i-\bar{x}}{\sigma_x}\right)\left(\frac{y_i-\bar{y}}{\sigma_y}\right) \tag{2-1}$$

where $\bar{x}$ is the mean of x, and σ_x is the standard deviation. Pearson correlation measures the linear relationship between two variables. Pearson correlation coefficient r_{xy} equals 1 or -1 when two variables x and y have a linear relationship: $y=kx+b$, the sign of r_{xy} is the same as k; r close to 0 shows that x and y do not have linear relationship. Therefore Pearson correlation is a good measurement to detect the linear relationship between vectors x and y.

Pearson Correlation distance between vectors x and y is defined as:

$$d_{cor}(x,y)=1-r_{xy} \tag{2-2}$$

where r_{xy} is the Pearson correlation coefficient, $d_{cor}(x,y)\in[0,2]$. $d_{cor}(x,y)=0$ corresponds to $r_{xy}=1$, i.e. vectors x and y are linear correlated; $d_{cor}(x,y)=1$ corresponds to $r_{xy}=0$, i.e. vectors x and y are uncorrelated; $d_{cor}(x,y)=2$ corresponds to $r_{xy}=-1$, i.e. vectors x and y are negative linear correlated. If we are not interested in the sign of correlation coefficients and consider both positive and negative correlation as highly correlated, we can define correlation distance as:

$$d_{cor}(x,y)=1-\left|r_{xy}\right| \tag{2-3}$$

where $d_{cor}(x,y) \in [0,1]$. $d_{cor}(x,y) = 0$ corresponds to $r_{xy} = \pm 1$, i.e. vectors x and y are linear or negative linear correlated; $d_{cor}(x,y) = 1$ represents vectors x and y are uncorrelated.

Sometimes, people also use uncentered correlation, i.e. without subtraction of the variable mean. The uncentered correlation coefficient between vectors x and y can be defined as:

$$r_{xy} = \frac{x \cdot y}{\|x\|\|y\|} \tag{2-4}$$

People use uncentered correlation when they are interested in the difference of variable mean as well as the expression pattern. Actually uncentered correlation represents the angle between vectors x and y, so uncentered correlation distance is also called angle distance.

It is necessary to mention that the Pearson correlation distance between vectors x and y is equivalent to the Euclidean distance between the standardized vectors x and y, which have 0 mean and 1 standard deviation.

In this work, because we are interested in the linear relationships between expression profiles, we adopt Pearson correlation distance, as shown in equation (2-2), to measure the similarity (or distance) between expression profiles.

2.2.2 Clustering algorithms

The most widely used clustering algorithms in microarray data analysis include hierarchical clustering (Eisen, Spellman et al. 1998), K-means (Gasch and Eisen 2002), SOM (Self Organizing Map) (Kohonen 1997), Fuzzy K-means (or C-means) (Gasch and Eisen 2002). (D'Haeseleer, Liang et al. 2000) provide some review of the clustering algorithms for gene expression data. Clustering can be done either over genes or over samples. It can also be done together, e.g., biclustering(Cheng and Church 2000) (Tanay, etc., 2002), and co-clustering (Hanisch, etc., 2002), etc. Next, we will briefly describe some clustering algorithms related to this work.

Hierarchical clustering

Hierarchical clustering organizes the elements in a hierarchy tree structure, in which the height of the branch reflects the similarity between the elements or clusters connected by the

branch. The elements are located at the leaves of the tree. Hierarchical clustering can be done using a bottom-up approach. The algorithms merge similar elements or clusters and compute the new distances for the merged cluster. Finally, all the elements are merged together to form a tree structure. Hierarchical clustering can also be done using a top-down approach, i.e., split the whole data set into two clusters then recursively subdivide clusters until the clusters become single elements. The distance between clusters can be defined as the distance between the closest neighbors (single linkage clustering), furthest neighbors (complete linkage clustering), the distance between the centroids of the clusters (centroid linkage clustering) or the cluster centers, or the average distance of all patterns in each cluster (average linkage clustering). Using a different distance definition will result in different clustering results. Figure 2-1 gives a simple example of a hierarchical tree using single linkage clustering. The labels in the vertical axis represent the distance between the corresponding nodes.

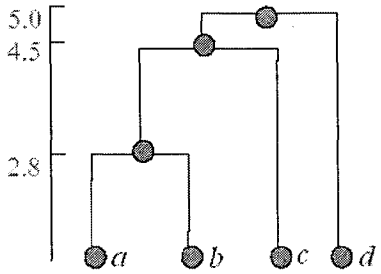


Figure 2-1 An example hierarchic clustering tree. D(a,b) = 2.8, D(a,c)=D(b,c)=4.5, D(b,d) = D(c, d) = 5.0

One advantage of hierarchical clustering algorithm is its high efficiency, because it requires no iterations like other partition clustering algorithms. Hierarchical clustering provides an overview of the element distribution, we can easily distinguish some outliers and groups of genes with high similarity. Also it has a similar representation as used in phylogeny, and so may be intuitive for biologists. However, hierarchical clustering is not an explicit partition into clusters because it has no iterations to converge to some optimal partitioning. Also, the order of the leaf elements of the hierarchical tree has no direct relation with their similarities. For example, in Figure 2-1, the leaf node a and b, can be plotted in either order ab or ba, similarly for node c and d. As a result, for a large hierarchical tree, it is hard to make sense of the data.

K-means

The K-means clustering algorithm partitions a set of elements into K clusters. By adjusting the cluster partitions or cluster center positions through iterations, it gradually converges to a minimum (usually local minimum) of the cost function:

$$E_K = \sum_{j=1}^{K} \sum_{i \in V_j} d_{ij} \qquad (2\text{-}5)$$

where V_j represents the j^{th} cluster, d_{ij} is the distance between the i element and the cluster center of cluster V_j.

The K-means clustering algorithm consists of a simple re-estimation procedure as follows. (1) Randomly initialize K cluster centers. (2) Assign each element to the nearest cluster center and form K clusters. (3) Recompute the cluster centers of each partition. (4) Repeat step 2 and 3 until a stopping criterion is met, e.g., when there is no further change in the assignment of the elements.

Compared with other partition clustering algorithms, the K-means clustering algorithm is very efficient. One disadvantage of the K-means clustering algorithm is that user must set the initial cluster number K, but usually the user has no idea of how many clusters might exist in the data. Of course, the user can guess and try, and find a good solution, but it is computationally expensive. One better way is to first use hierarchical clustering to get an overview of the data and estimate the cluster number, and then use the K-means clustering algorithm. Another problem of K-means is that user cannot get consistent results each time because of the random initialization of cluster centers and convergence to local optima.

Derivatives of K-means

There are lots of derivatives of the K-means clustering algorithm. The idea of most of them is to adapt the cost function (objective function) in equation (2-5), for example by adding some constraints to the cost function. For example, we can add one additional term in equation (2-5) to represent the cost of maximizing the inter-distance among cluster centers when minimizing intra-distance within the same cluster. We can design different constraints for specific problems.

Fuzzy K-means (C-means)

The difference between the fuzzy clustering algorithm and other algorithms (also called crispy clustering algorithms compared with the fuzzy one) is that fuzzy clustering allows one element to belong to multiple clusters instead of a single one. Fuzzy clustering algorithms use a membership function to represent the degree to which this element belongs to different clusters. As a result the cost function can be expressed as (Bezdek 1981; Gasch and Eisen 2002):

$$J(F,V) = \sum_{i=1}^{N}\sum_{j=1}^{K} m_{ij}^2 d_{ij}^2 \tag{2-6}$$

where $F = \{X_i,\ i = 1,...,N\}$ are the N data samples; $V = \{V_j,\ j = 1,...,K\}$ represent the K cluster centers. m_{ij} is the membership of X_i in cluster j, and d_{ij} is the Euclidean distance between X_i and V_j. One commonly used fuzzy membership function is:

$$m_{ij} = \frac{1/d_{ij}^2}{\sum_{k=1}^{K} 1/d_{ik}^2} \tag{2-7}$$

Self-Organizing Map (SOM)

SOM clustering is another popular clustering algorithm. It is similar to K-means. It also tries to minimize the objective function shown in equation (2-5) and discovers K clusters. However, the clustering process of SOM is different from K-means. It is a training process similar to that of neural networks. For details, please refer to (Kohonen 1997). Unlike K-means and hierarchical clustering, SOM clustering is designed to create a plot in which similar patterns are plotted next to each other. SOM, therefore, can be used for visualization.

2.2.3 Evaluation of clustering results

There are multiple ways to evaluate clustering results. One way is to compare the similarity of the element with those in the same cluster and those of the nearest cluster. Silhouette width is one popular measurement of this. For each element i, the *silhouette width* $s(i)$ is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2-8}$$

where $a(i)$ is the average dissimilarity between element i and all other elements of the cluster to which element i belongs. $b(i)$ is the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does *not* belong.

Observations with a large $s(i)$ (almost 1) are very well clustered, a small $s(i)$ (around 0) means that the observation lies between two clusters, and observations with a negative $s(i)$ are probably placed in the wrong cluster.

Other evaluation methods include the compactness of the clusters, ratio of the cluster diameter and the distance to the closest cluster.

All of these evaluation methods do not consider the prior knowledge of the data itself. If we incorporate prior knowledge of the data, we should have a better measurement for evaluation. For gene expression data, gene ontology information is good way to evaluate the clustering results. If the clusters match the biological explanation, we can say it is a good clustering.

2.3 Cluster annotation with Gene Ontology

In order to explain clustering results and explore the functions of unknown genes in the clusters, we need to annotate the clusters. The assumption of cluster annotation is that genes having similar expression patterns have similar or related functions. Usually, functions of some of the genes in the cluster are known. We suppose the common properties of the known genes should also be the properties of the cluster and its unknown genes. Therefore, cluster annotation becomes a problem of finding over represented properties of the genes with known functions.

For the convenience of computer interpretation and sharing among different science fields, the prior knowledge of gene is expressed in the Gene Ontology (GO: http://www.geneontology.org). GO is a shared, controlled vocabulary that is being developed to cover all organisms. GO has three categories: molecular function (MF), biological process (BP), and cellular component (CC). There are multiple algorithms proposed for annotation using GO, including the one proposed in Chapter 4. Here we introduce a simple algorithm by using the Hypergeometric test.

Suppose that there are N genes annotated for all GO categories of interest and that our GO category of interest contains m distinct genes. Then we can imagine an urn with N balls in it and $N-m$ are black while m are white. If we draw k balls from the urn, where k is the gene number in the cluster, we are asking whether the number of white balls in that drawn sample is unusually large. Suppose that there are q white balls (genes in the interested GO category) in the drawn sample, we then ask what is the probability $X \geq q$, where X is a Hypergeometric random variable with parameters as we have described.

$$P = 1 - \sum_{i=0}^{q-1} \frac{\binom{m}{i}\binom{N-m}{k-i}}{\binom{N}{k}} \tag{2-9}$$

For example, suppose the cluster has 20 genes, which are distributed over 25 GO categories. By searching the database, we find 5800 genes under these 25 GO categories. Suppose we want to test the significance of GO category j, which has 50 genes in total and 5 of them in the cluster. In this case, $N = 5800$, $k = 20$, $m = 50$ and $q = 5$. Based on equation (2-2), we can compute $P = 1.10 \times 10^{-8}$.

There are some issues that arise in the interpretation of these p-values. Usually many hypotheses are tested and some form of p-value correction is needed. Also, GO terms belonging to few genes typically have small p-values. (Gentleman 2003) provides more details over these issues.

2.4 Discussion

In this chapter, we briefly described the microarray technology, the preprocessing and normalization process, and make an introduction of commonly used clustering algorithms. Because there is lots of non-biological variation in the measurement process, microarray data is noisy. It is important to preprocess and normalize microarray data before further analysis. The normalization process tries to remove non-biological variations, but it can also remove real biological changes. So we should be cautious when adopting normalization algorithms. We should think about what kind of information we want to keep before performing normalization. For example, if we use quantile normalization, like RMA, for Affymetrix data,

all the chips will have same quantile values after normalization. Definitely, this will lose some real biological variation. If we are interested in this kind of variation, we should not adopt RMA, but instead adopt MAS 5.0 Signal normalization. With the improvement of microarray measurement technology, non-biological variations will be reduced in the microarray measurements. As a result, less normalization procedures are needed.

The clustering algorithms introduced in Chapter 2 are for general purposes. In this research, clustering is used as a preprocessing step for genetic network inference. A general purpose clustering algorithm does not suit this purpose well. In Chapter 4, a new Multi-scale Fuzzy K-means clustering algorithm is proposed for this purpose.

CHAPTER 3. INTRODUCTION TO GENETIC NETWORK INFERENCE

3.1 Introduction

Genetic network inference is to infer the gene (regulatory) relationships based on gene expression profiles. Just like solving other scientific problems, we first select a model, then fit the model with experimental data, and finally evaluate the model. To date, lots of network models and inference algorithms have been proposed for different situations. Most of them are for general purpose, which means they are also widely used in other scientific areas. As we described in the Section 1.3, inference of genetic networks is far from easy; there are lots of special challenges in genetic network inference, which is quite different from other scientific areas. Adapting the general models to the special genetic network inference problem is a challenge. Next, we will briefly review some of the most popular network models.

3.2 Network models

Figure 3-1 shows a generalized genetic network model. Suppose gene Y is regulated by a group of unknown genes X. The regulatory relationships between genes Y and X can be expressed as:

$$Y = F(X, t, b) \tag{3-1}$$

where $X = \{X_i, i = 1, ..., n\}$, X_i represents the i^{th} gene or its product participate in the regulation, t represents time, b represents the effect caused by unconsidered elements and noise.

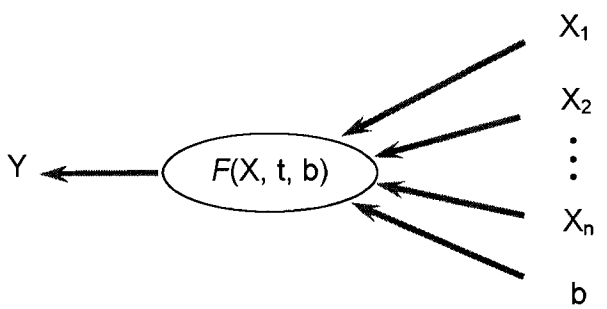


Figure 3-1 A generalized genetic network model

The genetic network model can

be categorized based on the variable states of X and Y and function F in equation (3-1). The variable states of X and Y can be discrete, continuous or fuzzy. The function F can be deterministic, like Boolean and differential equation models, stochastic, like Bayesian models, and Fuzzy, like Fuzzy Cognitive Maps. Next, we will briefly illustrate some type of these models.

Boolean Networks and derivatives

The Boolean network model is the simplest network model, and was first proposed by (Kauffman 1969). It uses a binary variable to define the state of a gene and uses Boolean functions (AND, OR, NOR, NAND) to define the gene relationships. Due to its simplicity, a Boolean network can analyze large-scale networks in an efficient way, but its simplicity makes a Boolean network "waste" a lot of useful information like: the detailed quantity information and time delay information for time series. Several improvements of Boolean networks have been proposed, such as Generalized Logical Networks (Thomas, Thieffry et al. 1995; Mendoza and Alvarez-Buylla 1998; Mendoza, Thieffry et al. 1999), Fuzzy Logic Models (Woolf and Wang 2000) and Probabilistic Boolean Networks (Shmulevich, Dougherty et al. 2002).

Linear Model

The gene regulatory model can be simplified as a linear model (D'Haeseleer, Liang et al. 1999):

$$x_i(k+1) = g(\sum_{j=1}^{J} w_{ij} x_j(k)) + n_{ik} \qquad (3\text{-}2)$$

where function g(.) is a monotonic function, x_i is a vector representing the gene expression profile of gene X_i, w_{ij} is the weight of gene X_j contributes to the expression value of gene X_i, n_{ik} represents other unconsidered elements and the internal transcription noise of gene X_i at k sample.

Many other network models, like coexpression networks (Stuart, Segal et al. 2003; Magwene and Kim 2004) and GGM(Graphical Gaussian Models) (Kishino and Waddell 2000; Toh and Horimoto 2002; Schafer and Strimmer 2004), are based on extensions of the linear model.

Differential Equations Model

The differential equation model (Landahl 1969; Smolen, Baxter et al. 2000) can be expressed as:

$$dx_i / dt = f_i(x_1(t-\tau_{i1}),\cdots,x_n(t-\tau_{in})), 1 \le i \le n \quad (3\text{-}3)$$

where $\tau_{i1},\cdots,\tau_{in} > 0$ denote time delays between the corresponding genes, and $f(x)$ can be a linear or nonlinear function.

Differential equation models show detailed quantities changing over time; they are detailed models and have many network parameters. As a result, they need more measurements to infer the network. Differential equation models have been widely used to model small biochemical networks (Martins, Mendes et al. 2001) which usually have more measurements in time scale and frequently have detailed kinetic information. For genetic network modeling, differential equation models are usually used for some small scale networks, like a group of interesting genes. Some simplified models based on differential equation models have been proposed. (Glass 1975; Edwards, Siegelmann et al. 2001) proposed a piecewise-linear differential equation model. Numerical simulation shows that in many cases that there are no qualitative differences between differential equation solutions and those based on the linear approximation (Edwards, Ibarra et al. 2001).

Stochastic Models

Differential equations assume the concentrations of the substances change continuously and deterministically. However, for gene interactions, there are only a small number of certain molecules and a single DNA molecule carrying the gene in one cell. Also there are many internal fluctuations existing in the cell. Therefore, people use stochastic models to represent the relationships among genes. A simple way of improving a differential equation model is to add an additional random term in the differential equation, which is called a stochastic differential equation (SDE). (Arkin, Ross et al. 1998) proposed a discrete and stochastic model. By including random term in Boolean networks, (Shmulevich, Dougherty et al. 2002) proposed probabilistic Boolean networks, as introduced in Boolean networks section. The Bayesian networks model is another based on stochastic assumptions. Details of Bayesian networks are described in the following section.

Bayesian Networks

Bayesian networks try to infer the causal relationships among genes based on probability theory. It is one of the most widely used models in genetic network inference. Bayesian networks are widely used in statistics and machine learning. They are used to infer the causal relationships among elements in the form of a directed acyclic graph (DAG). The nodes represent the elements (genes in our case), and the edges represent the causal relationships between the linked parent and child elements. It uses conditional probability to express the causal relationship between linked nodes, and joint probability of the network states to represent the network structures. The network structures with higher joint probabilities represent the higher possibilities in reality. The graph with the largest joint probability or likelihood is supposed to be the most probable network structure for the given data set.

By assuming the conditional probability distributions are independent from each other given their parents state (A same assumption as Markov Model), the joint probability distribution of the graph can be expressed as the multiplication of the conditional probabilities of each edge. This greatly simplifies the computation. In general the joint probability can be expressed as:

$$p(X) = \prod_{i=1}^{n} p(X_i \mid parents\ (X_i)) \qquad (3\text{-}4)$$

(Friedman, Linial et al. 2000; Friedman and Koller 2001) first proposed using Bayesian networks to model genetic networks. One problem with Bayesian networks is that they assume the graph is a DAG, which does not allow cycles. But for genetic network, cycles are common. They are the major mechanism to make biological system stable. Also Bayesian networks usually use discrete node states instead of continuous expression values, and do not consider the time delay information for the time series data. As a result, a lot of useful information is lost. As an improvement, a Dynamic Bayesian Network (DBN) was proposed (Murphy 1999; Perrin, Ralaivola et al. 2003; Kim, Imoto et al. 2004; Zou and Conzen 2005).

Dynamic Bayesian Networks (DBN) are an extension of Bayesian networks for time series data, which combines the features of HMM (Hidden Markov Model), utilizes time information and allows cycles in the networks. Most dynamic Bayesian network models are based on discretized expression values that result in the information loss. Also, most dynamic

Bayesian networks only consider first-order Markov relationships, i.e. the transition matrix only considers the connections between the adjacent time slices. However, in cells the time delay among gene interactions can vary over a wide range and the corresponding shift number of the time index is directly related with the sample interval. Dynamic Bayesian networks can be adapted to higher order Markov relation and continuous values at the expense of increased computation and model complexity that is not suited for data with a short time profile, especially in large scale network inference.

Fuzzy Cognitive Maps

Fuzzy cognitive maps (FCMs)(Kosko 1986; Dickerson and Kosko 1993; Dickerson, Cox et al. 2001) is also a graphical model to represent the causal relationships between nodes. It tries to use fuzzy sets to represent the degree of certainty instead of probability. The definition of nodes and edges are similar with Bayesian Networks. FCMs can include feedback, but Bayesian Networks cannot. FCM is a good way to map the expert's knowledge to a graph model, test the hypothesis and perform simulations. Compared with Bayesian networks, for FCM, the theory of constructing the network structure directly from raw data is not so complete.

Rule-based Formalisms

Rule-based or knowledge-based formalisms (Hofestadt 1995; Shimada, Hagiya et al. 1995) were developed in the field of artificial intelligence. The major advantage of rule-based formalisms is their capability to deal with a richer variety of biological knowledge in an intuitive way. It is counteracted by the difficulties in maintaining the consistency of the knowledge base and the problem of incorporating quantitative information.

Other models

There are some other models, like Petri Nets and its derivatives(Goss and Peccoud 1998; Matsuno, Doi et al. 2000), and hybrid models.

(Guet, Elowitz et al. 2002) proposed combinatorial synthesis of genetic networks. This model compares genetic networks as a binary logic circuit, which is composed of well-characterized genetic elements. Thus, genetic network modeling becomes a process of

combinatorial synthesis of these well-characterized genetic elements. Related works are (Hasty, Isaacs et al. 2001; Savageau 2001).

3.3 Network inference

After selecting a genetic network model, the next step is fitting the model to the data set. For most of the models, this is basically equivalent to solving a group of equations. Here, we use linear equations as an example to illustrate this problem. For the linear system of equations

$$Ax = b, \tag{3-6}$$

where A is the $n \times k$ matrix of coefficients, x is the $k \times 1$ column vector of variables (unknown network parameters), and b is the $n \times 1$ column vector of solutions.

When $k > n$, the system is underdetermined and there is infinite solutions.

When $k < n$, the system is (in general) over-determined and there is no solution. In this case, we can compute the optimum solution with least mean square errors.

When $k = n$ and the matrix A is nonsingular, then the system has a unique solution in the n variables.

Unfortunately, for genetic network inference, there are large numbers of variables (network parameters) and very few measurements, i.e., $k \gg n$. In most cases, the system is under-determined: there are infinite possible solutions, but only one of them is real.

One obvious way to deal with the dimension problem is to decrease the parameter number k. The parameter number k can be decreased by adopting simple models or reducing the number of nodes in the network. Clustering groups genes with similar profiles. The network model can be based on the cluster centers. This greatly reduces the network scale. Also we can select a small set of interesting genes and infer the network based on them. The difficulty is finding these genes and ensuring all related genes are included. Apart from decreasing the parameter number, constraints and assumptions can be added during inference to deal with the dimension problem. Next, we will describe some of these methods.

3.2.1 Heuristic methods with constraints on the solution

Just as other scientific areas, optimization based heuristic method is a common way to estimate the model parameters. Usually, it defines an objective function first, and then uses an optimization process to find (usually local) optimal network model parameters by maximizing or minimizing the objective function. Different optimization processes could be adopted by different models. Most of them have already been widely used in engineering and other scientific fields. For example, the Recurrent Neural Network (RNN) (D'Haeseleer 2000) was used for the differential equation model, and Expectation Maximization (EM) was used for the Bayesian Network model. Parallel optimum search algorithms like Genetic Algorithm (GA) (Wahde and Hertz 2000), sequential optimum search algorithm like Simulated Annealing and a lot of other algorithms were used to escape local optima. Different algorithms are better suited for different situations. Here we will not illustrate them in detail.

Since there are so many unknown parameters and hidden variables, the data samples for each gene are far from sufficient. The inferred "optimal networks" could easily over fit the data set and usually are not the real network. This case is equivalent to the underdetermined linear equation systems described previously. One solution is to add constraints and assumptions on the solutions during the optimization process. We can assume the network is sparse, the maximum number of the input or output edges for each node is limited to a small number, etc. If we have other prior knowledge of the networks, it can also be incorporated into the network inference process. With the constraints added, the solution space will be greatly reduced and it is more probable to obtain solutions that are close to the real one. The problem is choosing biologically reasonable constraints.

Another fact, often neglected, is how to design a biologically reasonable objective function. The real genetic networks are evolutionarily and environmentally related, so they are not necessarily optimized in a mathematical way. This means the resultant genetic network structures inferred by the optimization method are not necessary what is expected.

3.2.2 Inference based on constraints

An alternative way of genetic network inference is based on constraints. The basic idea of this is to reduce the solution space by eliminating cellular behaviors that cannot exist due to

known constraints. These constraints include reaction stoichiometry (like mass, energy and redox balance), thermodynamics, enzyme capacity and regulatory constraints. Flux Balance Analysis (FBA), Energy Balance Analysis, Extreme pathways analysis, etc. and their combinations are constraint-based analysis methods (Covert, Schilling et al. 2001; Famili, Forster et al. 2003; Reed and Palsson 2003). The solution space can be further reduced by combining prior expert knowledge and other information.

Therefore, the resultant networks using constraint based inference are not a final genetic network. Instead they are a group of all possible network structures. By combining other information, we can gradually reduce the solution space to a small group of networks. Finally, wet lab experiments can be used to resolve the real network structures.

3.2.3 Network inference by pair wise correlation

For the linear model shown in equation (3-2), the basic goal of the genetic network inference is to determine whether w_{ij} is zero or non-zero, i.e., whether there is a link between genes X_i and X_j or not. This is equivalent to determining whether there is a linear or approximately linear relationship between genes X_i and X_j. The Pearson correlation measures the linear relationship between two variables. The Pearson correlation coefficient r equals 1 or -1 when two variables x and y have a linear relationship: $y = kx + b$, the sign of r is the same as k; r close to 0 shows that x and y do not have a linear relationship. Therefore, instead of fitting the linear model in equation (3-2), we just need to calculate pair wise Pearson correlation to detect the linear relationship between genes X_i and X_j, i.e., to tell whether gene X_j interacts with gene X_i.

Genetic network inference by pair wise correlation will greatly reduce the requirements of profile length, because it is only necessary to calculate the pair wise correlation and involves only two variables. Theoretically, as long as the profile length is larger than 3 samples (include 3 samples), we can infer the network with any number of nodes. However, in order to detect more significant interactions, a longer profile length is required, especially when detecting low-weight interactions for the case where there are multiple independent variables x_j on the right of equation (3-2). Another problem of using correlation is that it

will result in lots of indirect links because correlation is transitive. In Chapter 4, we will propose some algorithms to deal with this problem. Also, just like the linear model, correlation based algorithms cannot deal with complex situation which cannot be approximated by linear relationships.

3.4 Discussion

In this chapter, we briefly described the genetic network models and inference algorithms. It is difficult to say one is better than another. The selection of the model depends on the purpose of the network inference and availability of the data set. If we have a long profile length and are interested in a small group of genes, we can select more detailed models, or else simplified models can be adopted. If the measurements are very noisy, using discrete values is perhaps the better choice. It is possible to combine different models to form hybrids, like linear stochastic model and stochastic differential equation models. We can also use a simple model first, and then use more complex models for sub-networks. With constraints and assumptions added, using complex models over limited profile lengths is also possible. The difficulty is how to ensure the constraints and assumptions are biological reasonable.

For the time series profile, the length is usually less than 20. Actually, the length of most profiles is even less than 10. The complex models like differential equation models are not suit for large scale networks. Instead, we adopt pair wise time correlation to detect linear relationships and estimate the time delay information. For details, please refer to Chapter 5.

CHAPTER 4. MODELING GENETIC NETWORKS USING FUZZY LOGIC[1]

4.1 Introduction

The behavior of biological systems is inherently fuzzy. Genes influence one another and are active at different level to different degrees. Many organisms have had their genomes completely sequenced, making it possible to begin to identify all the genes and their function in the organism. The major challenge in the post-genome era is to understand how interactions among molecules in a cell determine its form and function. This points to the need to develop methodologies to identify and analyze the complex biological networks that regulate metabolism. Metabolic networks form the basis for the net accumulation of biomolecules in living organisms. Regulatory networks modulate the action of these metabolic networks, leading to physiological and morphological changes. Even though new high-throughput transcriptomic, proteomic, and metabolomic analysis technologies give biologists vast amounts of valuable data, techniques that model uncertainty are needed to cope with the many genes of uncertain function and to understand complex interactions.

Gene expression (or transcriptomic) data in the form of high-throughput microarray experiments measures the amount of RNA associated with each of thousands of genes in parallel. The expression of each gene, as reflected by level of accumulation of the corresponding RNA, is not just turned on and off like a light switch. Clustering analysis has been used to hypothesize gene function under the assumption that genes that show similar expression patterns must be coregulated or part of the same regulatory pathway. Fuzzy clustering methods allow genes to belong to multiple clusters and participate in multiple pathways, thus reflecting the known biological reality of cellular metabolism. Fuzzy systems also aid in incorporating known information about some genes into the network.

---

[1] This chapter is the extended version of the paper: Du, P., J. Gong, et al. (2005). "Modeling Gene Expression Networks using Fuzzy Logic." IEEE Trans. on SMCB (Systems, Man and Cybernetics, Part B) **35**(6): (in press).

Gene expression networks show how genes regulate metabolism. Previous work used different machine learning methods to construct hypothetical networks. These methods produced high numbers of false positive connections due to inadequate sampling of the biological process in time and the on/off assumption described previously. In order to get biological meaningful results, information must be combined from a variety of sources to construct networks. Such fuzzy expert knowledge includes databases of genes and their products, as well as information about the interactions that occur between them. This work models the interactions between genes in gene regulatory pathways using fuzzy weights.

4.2 Background

4.2.1 Transcriptomics data

Gene expression describes the transcription of the information contained within the DNA, the repository of genetic information, into messenger RNA (mRNA) molecules. mRNA molecules are then translated (Here "translate" means that messenger RNA directs the amino acid sequence of a growing polypeptide during protein synthesis) into the proteins that perform most of the critical functions of cells. The analysis of the types and quantities of mRNAs produced by a cell (transcriptomics) indicates which genes are transcribed under specific conditions. Gene expression is a highly complex and tightly regulated process that allows a cell to respond dynamically both to environmental stimuli and to its own changing needs. This mechanism controls which genes are expressed in a cell and acts as a "volume control" that increases or decreases the level of expression of particular genes as necessary (National Center for Biotechnology Information (NCBI) 2004). Fuzzy metrics can express both concepts simultaneously. The challenge currently facing biological researchers is to discover the functions of the genes and how they interact.

DNA microarray technology exploits the ability of a given mRNA molecule to bind specifically to, or hybridize to, the DNA template from which it originated. Microarrays allow scientists to measure, in a single experiment, the expression levels of thousands of genes within a cell. The amount of mRNA bound to the spots on the microarray is precisely measured, generating a profile of RNAs accumulated in the cell. This work uses microarray

data from the Affymetrix *Arabidopsis* ATH1 genome array, that analyzes 22K genes at a time (Affymetrix Inc. 2001).

Researchers use microarrays to detect expression patterns— the extent to which each particular gene(s) is being expressed more or less under a set of specific circumstances. These gene expression patterns can give insights into the gene functions and the underlining gene regulatory networks.

4.2.2 Finding patterns in microarray data

In related biological processes, many genes are highly coregulated (i.e., their gene expression patterns are similar). Figure 4-1 shows an example of highly coregulated gene expression profiles in the diurnal biological process of the model plant *Arabidopsis* (a member of the mustard family, widely used as a model organism in plant biology)(Fatland, Ke et al. 2002; Foster, Ling et al. 2004). Clustering is widely used to find these coregulated genes(Eisen, Spellman et al. 1998; Spellman, Sherlock et al. 1998; Iyer, Eisen et al. 1999; Hastie, Tibshirani et al. 2000). Many popular cluster algorithms are hard clustering algorithms, e.g., hierarchical clustering or K-means clustering. In these algorithms, a gene can only belong to one cluster. In actuality, a single gene may be involved in different biological processes. Furthermore, gene expression patterns may be similar only under a subset of conditions. Hard clustering algorithms cannot extract the gene relationships described above. Fuzzy K-means uses membership values to measure the relationship between a gene and its clusters (Bezdek 1981; Gasch and Eisen 2002). As a result, a gene can belong to several clusters to a degree.

Clustering, by itself, does not delineate the causal relationship between genes. RNA profiles are very noisy and may be unequally sampled in time. Using cluster centers, instead of individual gene expression profiles, smoothes by averaging individual gene profiles within the cluster. This is equivalent to a low-pass filter. Thus, clusters of highly coregulated genes can be modeled as a single entity when inferring the gene regulatory relations. A gene transcription response usually can occur in from tens of minutes to several hours, so time delay correlation can help determine the causal relationship.

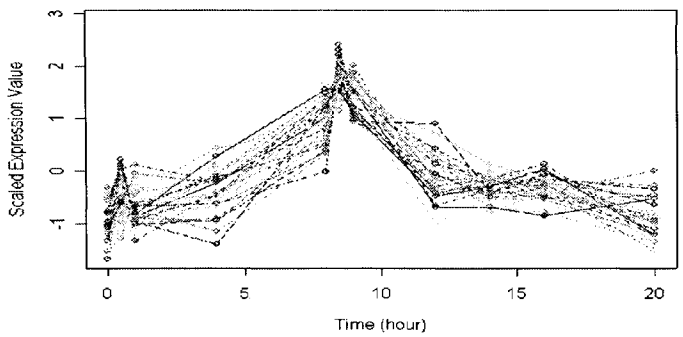


Figure 4-1 Coregulated gene expression patterns behave similarly across a range of conditions. In this example, the index is hours into a short growth day. The expression values are normalized to a mean of zero and a standard deviation of one. A cluster window scale of *sc* = 0.1 was used.

4.2.3 Gene regulatory networks

Regulatory networks reflect causal interactions among biomolecules in living systems. Gene regulatory networks can be defined as regulatory networks that consider transcriptomics data. Several types of models have been proposed for representing regulatory networks in biological systems, including Boolean networks (Liang, Fuhrman et al. 1998; Akutsu, Miyano et al. 1999), linear weighting networks (Weaver, Workman et al. 1999), differential equations (Akutsu, Miyano et al. 2000), and Bayesian Networks (Murphy 1999; Murphy 2002; Perrin, Ralaivola et al. 2003). Circuit simulations and differential equations require detailed information that is not yet known about the regulatory mechanisms between entities. Boolean networks analyze binary state transition matrices to look for patterns in gene expression. Each part of the network is either on or off depending on whether a signal exceeds a pre-determined threshold. Generalized Logical Networks (Thomas, Thieffry et al. 1995; Mendoza and Alvarez-Buylla 1998; Mendoza, Thieffry et al. 1999) allow the variables in Boolean networks to have more than two values and use generalized Boolean functions to define the relationship. Probabilistic Boolean Networks combine several promising predictors or Boolean functions together, so that each makes a contribution to the prediction of a target gene. A probabilistic model randomly selects one of these promising predictors. Linear weighting networks have the advantage of simplicity since they use simple weight matrices to additively combine the contributions of different regulatory elements.

Bayesian networks model probabilistic transitions between network states. Bayesian networks assume that there are no cycles in a network. However cycles are the major mechanism to ensure stability or homeostasis. Dynamic Bayesian Networks combine the features of Hidden Markov Models to incorporate feedback (Murphy 1999; Murphy 2002; Perrin, Ralaivola et al. 2003).

This work models interactions (also referred to as edges or links) in the network as fuzzy functions that depend on the detail known about the network. Fuzzy cognitive maps are fuzzy digraphs that model causal flow between concepts (Dickerson and Kosko 1994) or, in this case, biomolecular entities, including RNAs, metabolites, and proteins (Dickerson, Cox et al. 2001; Cox, Fulmer et al. 2002). Entities stand for causal fuzzy sets where events occur to some degree. The entities are linked by interactions that show the degree to which these entities depend on each other. Interactions stand for causal flow. The sign of an interaction (+ or -) shows causal coregulation between entities. The fuzzy structure allows the entities levels to be expressed as continuous values. This modeling has demonstrated regulation in the *Arabidopsis* network, in the case of gibberellin conversion from an inactive form to an active form (Dickerson, Cox et al. 2001). Fuzzy cognitive maps (FCMs) have the potential to deal with the lack of quantitative information on how different variables interact. The FCModeler tool uses fuzzy methods for modeling networks and interprets the results using fuzzy cognitive maps. The FCModeler tool is intended to capture the intuitions of biologists, help test hypotheses, and provide a modeling framework for assessing the large amounts of data captured by RNA microarrays and other high-throughput experiments (Dickerson, Berleant et al. 2001).

For regulatory network modeling, there are a number of significant problems. All of these models are based on information about the quantities of one or more classes of entities. However, these values alone cannot give a complete picture of how the metabolism of living things works (Hatzimanikatis and Lee 1999). The number of measurements for each object is very limited due to experimental constraints. This is true especially for the complex models and large-scale networks. This makes it difficult to get enough data to use classical machine learning approaches.

Another difficulty is that different models and algorithms often produce different results. It is important to interpret the resulting network model from a biological viewpoint. The Gene Ontology (GO: http://www.geneontology.org) provides a way to do this (Ashburner and Lewis 2002; Blake and Harris 2003). GO is a shared, controlled vocabulary that is being developed to cover all organisms. GO has three categories: molecular function (MF), biological process (BP), and cellular component (CC). The existence of GO is not only providing us a controlled vocabulary, but paved another way to gene function prediction, clustering interpretation, and evaluation (Al-Shahrour, Diaz-Uriarte et al. 2004). This work uses an additive fuzzy system to assess the evidence for gene function in a cluster and for the interactions in gene regulatory networks.

4.3 Analysis methods

The analysis and creation of gene regulatory networks involves first clustering the data at different levels, then searching for weighted time correlations between the cluster center time profiles. The link validity and strength is then evaluated using a fuzzy metric based on evidence strength and co-occurrence of similar gene functions within a cluster.

4.3.1 Multi-scale Fuzzy K-Means Clustering

The Fuzzy K-means algorithm minimizes the objective function (Bezdek 1981; Gasch and Eisen 2002):

$$J(F,V) = \sum_{i=1}^{N}\sum_{j=1}^{K} m_{ij}^2 d_{ij}^2 \qquad (4\text{-}1)$$

where $F = \{X_i,\ i = 1,...,N\}$ are the N data samples; $V = \{V_j,\ j = 1,...,K\}$ represent the K cluster centers. m_{ij} is the membership of X_i in cluster j, and d_{ij} is the Euclidean distance between X_i and V_j. One commonly used fuzzy membership function is adapted as:

$$m_{ij} = \frac{1/d_{ij}^2}{\sum_{k=1}^{K} 1/d_{ik}^2} W\left(d_{ij}\right) \qquad (4\text{-}2)$$

where $W(d)$ is the window function centered at V_j and can take any form. Adding a window function $W(d)$ to the membership function limits the effects of cluster members far away from cluster centers. This work uses truncated Gaussian windows with values outside the range of 3σ set to zero:

$$W\left(d_{ij}\right)=\begin{cases} e^{-(d_{ij})^2/2\sigma^2} & d_{ij}<3\sigma \\ 0 & elsewhere \end{cases} \tag{4-3}$$

The window function $W(d)$ insures that genes with distances larger than 3σ will have no effect on the cluster centers. In the future analysis, we define σ as the window scale, *sc*, of $W(d)$ in equation (4-3), i.e. $sc=\sigma$.

Multi-scale Algorithm

The multi-scale algorithm is similar to the ISODATA algorithm with cluster splitting and merging(Ball 1965; Ball and Hall 1965). There are four parameters: K (initial cluster number), *sc* (the window scale of $W(d)$, $sc=\sigma$), T_{split} (split threshold), $T_{combine}$ (combine threshold). Whenever the genes are further away from the cluster center than T_{split}, the cluster is split and faraway genes form new clusters. Also, if two cluster centers are separated by less than $T_{combine}$, then the clusters are combined. Usually $T_{combine}\le\sigma$ and $2\sigma\le T_{split}\le 3\sigma$. The algorithm is given in Table 4-1. ε_1 and ε_2 are small numbers to determine whether the clustering converged. The advantage of this algorithm is that it dynamically adjusts the number of clusters based on the splitting and merging heuristics.

Table 4-1 Multi-scale Fuzzy K-Means Algorithm

1	Initialize parameters: K, *sc*, T_{split} and $T_{combine}$
2	Iterate using Fuzzy K-means until convergence to threshold ε_1
3	Split process: do split if there are elements farther away from cluster center than T_{split}.
4	Iterate using Fuzzy K-means until convergence to threshold ε_1
5	Combine Process: combine the clusters whose distance between cluster centers is less than $T_{combine}$. If the cluster after combining has elements far away from cluster center (distance larger than 3σ), stop combining.
6	Iterate steps 1-5 until converging to a given threshold ε_2.

Effects of window scale

Changing the window scale can affect the level of detail captured in the clusters. If $sc << 1$, then clusters are individual elements. As *sc* increases, the window gets larger. The result is a hierarchical tree that shows how the clusters interact at different levels of detail. This work uses three level of Multi-scale Fuzzy K-means clustering (*sc* = 0.1, 0.2 and 0.3). The initial number of clusters is $K = N$, the total number of data points, $T_{combine}=\sigma$, and T_{split} $=3\sigma$. Clustering results with different window scales provide different levels of information. At *sc* = 0.1, the cluster sizes are very small. These clusters represent very highly correlated profiles or just the individual gene profiles because many clusters only contain a single element. At *sc* = 0.2, smaller clusters are combined with nearby clusters. Highly correlated profiles are detected. The *sc* =0.3 level is the coarsest level.

Better estimation of cluster center

Since we want to use cluster centers to represent the expression patterns of the whole group of genes, high accuracy of cluster center estimation is required. Next, we will describe the cluster center estimation by given an example. Suppose there are two clusters as shown in Figure 4-2. Because of the clear cutting of clusters, algorithms like K-means will result in the cluster centers skew to the outer side of real cluster centers in the case of overlapping clusters. On the other hand, for Fuzzy K-means algorithms, because it allows overlapping of clusters, the cluster centers will skew to the inner side of the real cluster centers in the case of overlapping clusters. Also the genes far away from the cluster centers may also make the cluster center skew from real center. By multiplying $W(d)$, it limits the effects of cluster elements far away from cluster center, the genes with distances larger than 3σ will have no effect on the cluster centers. This helps Multi-scale Fuzzy K-means to make better estimation of cluster centers. Figure 4-2 shows the result of a simulation. There are two overlapping clusters. The clusters are produced by two dimensional normal random variable (x_0, y) and (x_1, y), x_0 and y are $N(1, 1)$ distributed and x_1 is N(3, 1) distributed. Figure 4-2.b shows the estimated cluster center position in x axis direction (y = 1) using three different cluster algorithm: K-means, windowed Fuzzy K-means (window scale *sc* =1) and Fuzzy K-means.

From the results, we can see windowed Fuzzy K-means got the best result. For the higher dimension and more complex cases, we can get similar results.

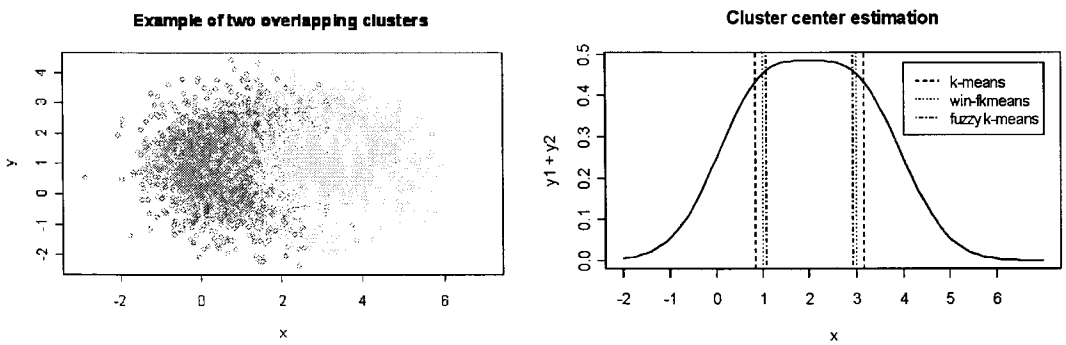


Figure 4-2 Accuracy comparison of cluster centroid estimation by three clustering algorithm. The real cluster centers locate at x =1 and 3, y = 1

Initialization of the cluster centers

To avoid the uncertainty of the clustering results because of random cluster center initialization, we can start clustering with individual genes as initial cluster centers. This makes us implement windowed Fuzzy K-means like the hierarchical clustering, i.e., starting from individual genes.

4.3.2 Construction of gene regulatory networks

Clustering provides sets of genes with similar RNA profiles. The next step is finding the relationships among these coregulated genes. If gene A and gene B have similar expression profiles, there are several possible relationships: 1. A and B are coregulated by other genes; 2. A regulates B or vice versa; 3. There is no causal relationship, just coincidence. Here, the regulation may be indirect, i.e., interaction through intermediates. These cases cannot be differentiated solely by clustering. Cubic spline interpolation generates equally sampled profiles as in (D'Haeseleer 2000).

The gene regulatory model can be simplified as a linear model (D'Haeseleer, Liang et al. 1999):

$$x_A(t) = \sum_B w_{BA} x_B(t - \tau_{BA}) + b_A \qquad (4\text{-}4)$$

where $x_A(t)$ is the expression level of gene A at time t, τ_{BA} is the time delay of gene B regulating gene A, w_{BA} is the weight indicating the inference of gene B to A, b_A is a bias indicating the default expression level of gene A without regulation. The gene expression profile $\mathbf{x}_A$ is a series of time samples of $x_A(t)$.

Standardizing gene expression profiles to 0 mean and 1 standard deviation removes b_A from equation (4-4). The goal is to find out if genes A and B have a regulatory relationship, the weight $w_{BA} = [-1,0,1]$ (0 means no regulatory relation, 1 or -1 means strongly regulated or negatively regulated). The time correlation between genes A and B can be expressed in discrete form as:

$$R_{AB}(\tau) = \mathrm{cov}(\mathbf{x}'_A, \mathbf{x}'_B) \tag{4-5}$$
$$\mathbf{x}'_A[k] = \mathbf{x}_A[k],\ \mathbf{x}'_B[k] = \mathbf{x}_B[k+\tau],\ k = 1,\ldots,N,\ 1 \le k+\tau \le N$$

where $\mathbf{x}_A$ and $\mathbf{x}_B$ are the standardized (zero mean, standard deviation of unity) expression profiles of genes A and B. τ is the time shift. For a periodic time profile, we can use circular time correlation, i.e., the time points at the end of the time series will be rewound to the beginning of series after time shifting. For multiple data sets, the time correlation results of each data set are combined as:

$$R^C_{AB}(\tau) = \sum_k w_k R^k_{AB}(\tau) \tag{4-6}$$

where $R^C_{AB}(\tau)$ is the combined time correlation result, $R^k_{AB}(\tau)$ is the time correlation result of the k^{th} data set, w_k is the weight of k^{th} data set that depends on the experiment reliability and the length of the expression profile.

The value $\max|R^C_{AB}(\tau)|$ can be used to estimate the time delay τ' between expression profiles of genes A and B. Given a correlation threshold T_R, if $\max|R^C_{AB}(\tau)| > T_R$, there is significant regulation between genes or clusters. By defining the clusters as nodes and significant links as edges, we can get the gene regulation network of these clusters. We can define four types of regulation:

$R^C_{AB}(\tau') > 0,\ \tau' \neq 0$, positive regulation between genes A and B;

$R^C_{AB}(\tau') < 0,\ \tau' \neq 0$, negative regulation between genes A and B;

$R^{C}_{AB}(\tau') > 0$, $\tau' = 0$, genes A and B are positively coregulated;

$R^{C}_{AB}(\tau') < 0$, $\tau' = 0$, genes A and B are negatively coregulated.

The sign of τ' determines the direction of regulation. $\tau' > 0$ means gene B regulates gene A with time delay τ'; $\tau' < 0$ means gene A regulates gene B with time delay τ'.

4.3.3 Network evaluation using fuzzy metrics

The available gene ontology (GO) annotation information can estimate a fuzzy measure for the types or functions of genes in a cluster. The GO terms in each cluster are weighted according to the strength of the supporting evidence information and the distance to cluster center. An additive fuzzy system is used to combine this information (Kosko 1992). Every GO annotation indicates the type of supporting evidence. This evidence is used to set up a bank of fuzzy rules for each annotated data point. Different fuzzy membership values are given to each evidence code. For example, evidence inferred by direct assays (IDA) or from a traceable author statement (TAS) in a refereed journal has a value of one. The least reliable evidence is electronic annotation which is known to have high rates of false positives.

Table 4-2 Evidence codes and their weights (http://www.geneontology.org/GO.evidence.html)

Evidence Code	Meaning of the Evidence Code	Evidence Weight, wevi
IDA	Inferred from direct assay	1.0
TAS	Traceable author statement	1.0
IMP	Inferred from mutant phenotype	0.9
IGI	Inferred from genetic interaction	0.9
IPI	Inferred from physical interaction	0.9
IEP	Inferred from expression pattern	0.8
ISS	Inferred from structural similarity	0.8
NAS	Non-traceable author statement	0.7
IEA	Inferred from electronic annotation	0.6
	Other	0.5

Each gene in a cluster is weighted by the Gaussian window function in equation (4-3). This term weights the certainty of the gene's GO annotation using product weighting. Each gene and its associated GO term are combined to find the possibility distribution for each single GO term that occurs in the GO annotations in one cluster. One gene may be annotated by several GO terms, and each GO term has one evidence code. Each GO term may occur K

times in one cluster, but with a different evidence code and in different genes. For the n^{th} unique GO term in the j^{th} cluster, the fuzzy weight is the sum of the weights for each occurrence of the term:

$$W_{GO}(j,n) = \sum_{i=1}^{K} w_{GO,j}(i,n) \tag{4-7}$$

where $w_{GO,j}(i,n) = w_{evi}(i,n) \cdot W(d_{ij})$, w_{evi} is given in table II, and $W(d_{ij})$ is the same as equation (4-3).

This provides a method of pooling uncertain information about gene function for a cluster of genes. This gives an additive fuzzy system that assesses the credibility of any GO terms associated to a cluster (Kosko 1992). The results can be left as a weighted fuzzy set or be defuzzified by selecting the most likely annotation. For each cluster, the weight is normalized by the maximum weight and the amount of unknown genes. This is the weighted percentage of each GO term p_{weight}:

$$p_{weight}(j,n) = \frac{W_{GO}(j,n)}{W_{root}(j) - W_{unknown}(j)} * 100\% \tag{4-8}$$

where $W_{GO}(j,n)$ represents the weight of the n^{th} GO term in the j^{th} cluster. $W_{unknown}(j)$ is the weight of GO term in cluster j: xxx unknown, e.g., GO: 0005554 (molecular_function unknown). $W_{root}(j)$ is the weight of root in cluster j. GO terms are related using directed acyclic graphs. The root of the graph is the most general term. Terms further from the root provide more specific detail about the gene function and are more useful for a researcher. The weight of each node is computed by summing up the weights of its children (summing the weights of each of the N GO terms in a cluster):

$$W_{root}(j) = \sum_{n=1}^{N} W_{GO}(j,n) \tag{4-9}$$

The higher weighted nodes further from the root are the most interesting since those nodes refer to specific biological processes.

4.4 Clustering results

The tested data set compared Arabidopsis thaliana plants, wild-type (WT) and transgenic plants containing antisense ACLA-1 behind the constitutive CaMV 35S promoter (referred to

as aACLA-1). The microarray type was an Affymetrix GeneChip. The data consisted of two replicates; each with eleven time points (0, 0.5, 1, 4, 8, 8.5, 9, 12, 14, 16, 20 hours), and changing from light (from 0 to 8 hours) to dark (from 8 to 20 hours) (Fatland, Ke et al. 2002; Foster, Ling et al. 2004). Only ACLA-1 seedlings exhibiting features characteristic of the antisense phenotype were used. Total RNA was extracted from leaves and used for microarray analyses.

The Affymetrix microarray data were normalized with the Robust Multichip Average (RMA) method (Gautier, Cope et al. 2004). The replicates of each gene expression profile are standardized to zero mean, one standard deviation. The data was filtered by comparing the expression values between the WT and ACLA1 gene mutated at 1, 8.5 and 12 hours. Differentially expressed genes having fold changes larger than 2 times at any of the time points 1, 8.5 and 12 hours were kept. 484 genes remained after filtering. The gene expression patterns used for clustering are the time point measurements for the wild-type plant.

The data was combined so that each data point consists of a gene evaluated at a series of time points. Three-level Multi-scale Fuzzy K-means clustering was used, with window scale of $sc = 0.1$, 0.2, and 0.3. The initial number of clusters, K, was the number of genes. There were 236 clusters at $sc = 0.1$; 28 clusters at $sc = 0.2$; and 5 clusters at the $sc = 0.3$ level.

Figure 4-1 and Figure 4-3 show typical cluster patterns for at window scale $sc = 0.1$ and 0.2 respectively. The cluster in Figure 4-1 is much more tightly coregulated than Figure 4-3 with less variation. Figure 4-4 shows the cluster center profiles of 5 cluster centers at the $sc = 0.3$ level. At this coarse level, information such as whether the gene expression level increases or decreases in the day or night is given. Figure 4-4 shows that clusters 2 and 3 decrease in the day and increase at night, while cluster 1, 4 and 5 are opposite. At $sc = 0.2$, the regulatory relationships can be studied at a more detailed level. There are 28 clusters at this level. Figure 4-5 shows their relationship with the $sc = 0.3$ clustering. Several clusters from $sc = 0.2$ belong to more than one cluster at $sc = 0.3$. This is due to genes in these sub-clusters being involved in multiple related biological processes. In Figure 4-4, clusters 2 and 3 represent the genes active at night, and clusters 1, 4, and 5 are active in the day. Figure 4-5 shows that genes active at night are more tightly coregulated than those active in the day. Biologically, this indicates the ACLA1 related genes are mainly active in the day and their

expressions are diversified. At $sc = 0.1$, clusters were further subdivided into 236 clusters. Many of these clusters only included 1 or 2 genes. Given the noise in microarray experiments and the small number of genes in each cluster, we did not further study at this level.

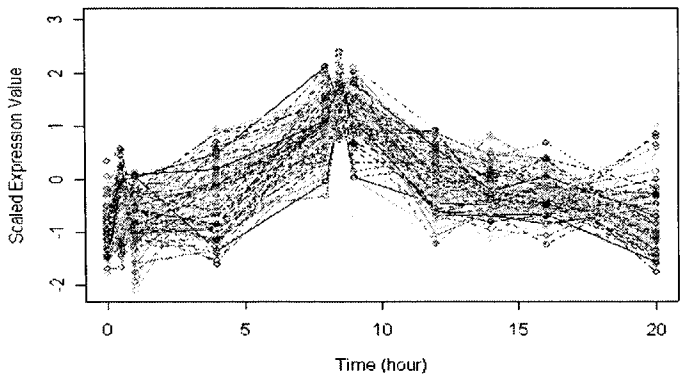


Figure 4-3 Coregulated gene expression patterns behave similarly across a range of conditions. In this example, the index is hours into a short growth day. The expression values are normalized to a mean of zero and a standard deviation of one. A cluster window scale of $sc = 0.2$ was used.

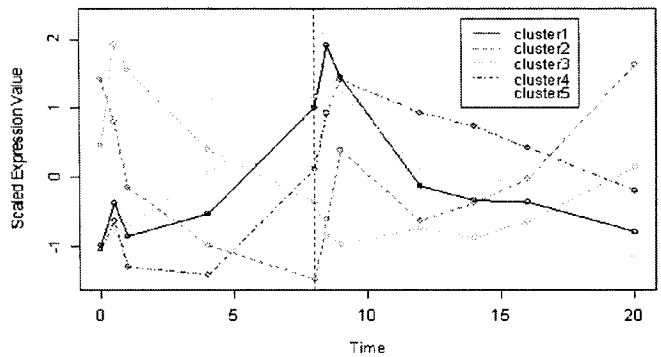


Figure 4-4 Cluster center profiles for the window scale $sc = 0.2$ level.

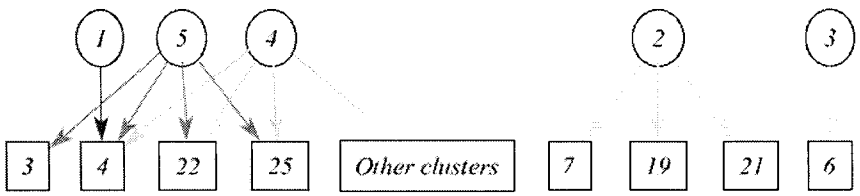


Figure 4-5 Relationship between the clusters from the $sc = 0.2$ case (cluster numbers in rectangles) to the clusters in the $sc = 0.3$ case (cluster numbers in circle).

4.5 Inferring and modeling gene regulatory networks

4.5.1 Construct the genetic network using time correlation

The genetic networks among the clusters of highly coregulated genes can be constructed based on their cluster center profiles. Since the data used were unequally sampled with 0.5h as minimum interval, we interpolated the gene expression profiles as equally sampled 41 time points with 0.5h intervals using cubic spline interpolation. The time correlation of each replicate $R_{ij}^{k}(\tau)$, k=1, 2 was computed using equation (4-7), then combined using equation (4-8) as $R_{ij}^{C}(\tau)$ with weight $w_k = 0.5, k = 1,2$. τ was limited to the range of [-4h, 4h] because the light period only lasted 8 hours in this data set. The genetic networks were constructed with a correlation threshold of $T_R = 0.65$. The strength of correlation was mapped into three categories: [0.65, 0.75), [0.75, 0.85), and [0.85, 1]. Three types of line thickness from thin to thick represent the strength of the correlation. Blue dashed lines represent positive coregulation; red dashed lines represent negative coregulation; solid lines with bar head represent negative regulation; solid lines with arrowheads represent positive regulation.

Figure 4-6 shows the constructed gene regulatory networks based on the cluster center profiles shown in Figure 4-4. The networks indicate clusters 1 and 5 are highly coregulated (0 time delay), clusters 1 and 5 positively regulate cluster 4 with time delays of 2.5h and 3h, and both negatively regulated cluster 3 with a time delay of 1.5h; cluster 4 is negatively regulated by cluster 3 with delay 1h, the correlation between cluster 2 and cluster 4, and cluster 1 and 3 is not strong. All of these relationships are correspond to the cluster center profiles. This means the algorithm correctly resolved the relationships between cluster centers.

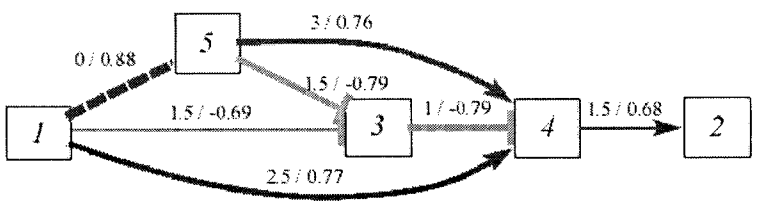


Figure 4-6 Gene regulatory networks inferred at level *sc* = 0.3. The numbers on each link show the time delay for the interaction on top and the correlation coefficient of the interaction on the bottom.

Figure 4-7 shows the constructed regulatory networks of the 28 cluster centers at *sc* = 0.2 level. The graph notations are the same as in Figure 4-6. The graph shows that there is one highly connected group of clusters. The other clusters at the upper right corner are less connected. The relationships between clusters may become complex with a large number of edges. Simplification of the networks is necessary when there are many highly connected clusters.

Figure 4-7 shows possible duplicate relationships. This can be analyzed using the path search function in FCModeler. In Figure 4-7, from cluster 15 to 19, there are two paths: one is directly from cluster 15 → 19 with time delay 1h and correlation coefficient, ρ = -0.85; another path is cluster 15 → 7 with time delay 0.5h and correlation coefficient, ρ = -0.89, and then from 7→ 19 with time delay 0.5h and ρ = 0.81. The total time delays of both paths are the same. So it is very possible one of the paths is redundant. Figure 4-8 shows part of the simplified graph of Figure 4-7.

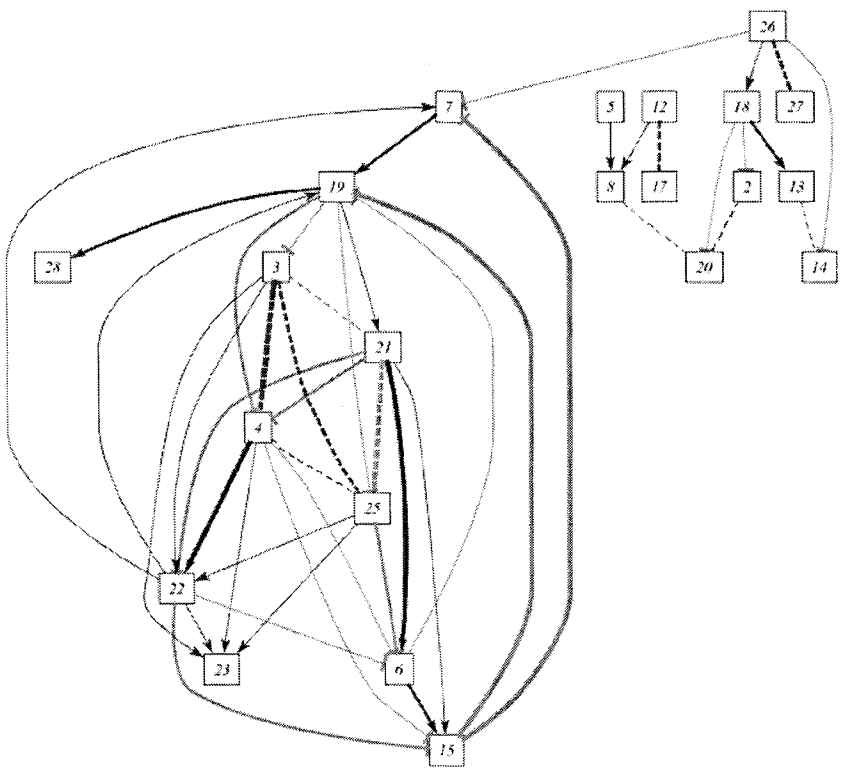


Figure 4-7 Regulatory networks among cluster centers at the window scale *sc* = 0.2 level. The graph annotations are the same as in Figure 4-6.

4.5.2 Cluster and network evaluation using weighted GO terms

Cluster evaluation makes use of the available GO information to find out what kind of functions or processes a cluster involves. In Figure 4-7, the graphs in the upper right corner are less connected. The Gene Ontology shows most of these clusters are not annotated. This means these clusters have no biological evidence of direct relation with the highly connected group. It also shows how the multi-scale fuzzy algorithm successfully separates those unrelated genes.

Figure 4-8 shows that cluster 3 and 4 are highly coregulated (correlation coefficient between cluster centers is 0.91). The cluster is split because the combined cluster 3 and 4 has a cluster radius larger than 3σ. Table III shows the fuzzy weights for the GO terms in each cluster. The BP (Biological Process) GO annotations show that clusters 3 and 4 involve many similar biological processes. For example, both clusters involve "Carboxylic acid metabolism", "Regulation of transcription, DNA-dependent", and "Protein amino acid phosphorylation". Cluster 3 has more emphasis on "Regulation of transcription, DNA-dependent" and cluster 4 emphasizes "Protein amino acid phosphorylation". Also cluster 3 involves "water derivation", but cluster 4 mainly involves another BP "Response to desiccation, hyper osmotic salinity and temperature". Clusters 3 and 4 provide a good example of the overlapping of fuzzy clusters, while the separation of two clusters does make sense.

Clusters 21 and 25 are two highly negatively coregulated clusters. Cluster 21 involves "Photosynthesis, dark reaction" which is active at night, while cluster 25 mainly involves "Carboxylic acid metabolism" and other metabolism usually active in the day. Cluster 21 contains genes for "Trehalose biosynthesis". Trehalose plays a role in the regulation of sugar metabolism, which has just been identified for *Arabidopsis*(Eastmond and

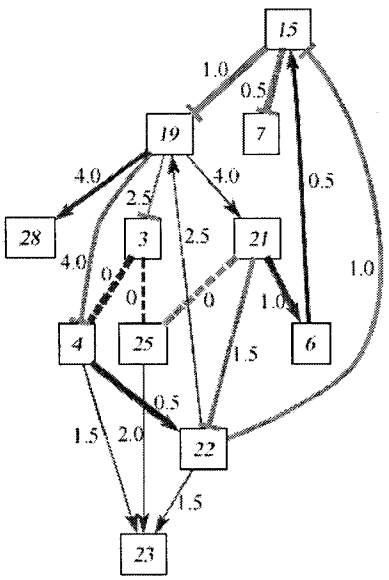


Figure 4-8 Simplified regulatory networks with redundant edges removed for the window scale $sc = 0.2$ level. The number on each link represents the estimated time delay.

Graham 2003). Clusters 6 and 21 involve sugar metabolism (carbohydrate metabolism in GO term). This is a significant biological result for understanding regulation in this experiment.

Figure 4-7 and Figure 4-8 show that cluster 19 regulates clusters 3, 4, 21, 22, 25 and 28. After checking the BP GO annotations, we found the annotated genes in cluster 19 fall in three categories: "Protein Metabolism" ("N-terminal protein myristoylation", and "Protein folding"), "Response to auxin stimulus" and "Cell-cell signaling". "N-terminal protein myristoylation", and "Protein folding" are two major protein regulation mechanisms, while "Response to auxin stimulus" and "Cell-cell signaling" involve the processes of receiving stimulus or signals from others. Therefore these BP GO annotations match our network structures.

Clusters 23 and 28 have no out-going edges, which implies that they are not involved in regulatory activity. Clusters 3, 4, 6, 7, 15, 19 21 22, and 25 involve one or several of "Regulation of transcription, DNA-dependent", "Protein amino acid phosphorylation" or "N-terminal protein myristoylation" biological processes. The later two are two major protein regulation mechanisms. Also cluster 21 involves Trehalose regulation as shown earlier. The BP annotations for clusters 23 and 28 are "Response to stimulus" and "Carbohydrate metabolism" which are non-regulatory.

Table 4-4 shows that the molecular functions in clusters 23 and 27 are all unknown, and cluster 28 has only one function: hydrolase activity. The remaining clusters mainly have the following molecular functions: nucleic acid binding, nucleotide binding, transferase activity, hydrolase activity and transcription regulator activity. Also we found that the MF annotation of cluster 19 is focusing on only two functions: hydrolase activity, specifically hydrolyzing O-glycosyl compounds, and signal transducer activity.

Most of the clusters have the following molecular functions: binding, catalytic activity, and transcription regulator activity. Clusters 3 and 4 are the most similar clusters in the sense of molecular function. The largest weight is on DNA binding, and they both include: purine nucleotide binding, oxygen binding, and carbohydrate binding. Also, both clusters contain active genes that attend transferase activity (transferring phosphorus-containing groups), hydrolase activity (acting on glycosyl bonds), and oxidoreductase activity. The only difference is that cluster 4 contains genes acting in transporter activity.

Table 4-3 Cluster annotation of Biological Process GO (W_{root}, $W_{GO}(j,n)$ and $p_{weight}(j,n)$ as defined in equation (4-9))

Cluster Index (W_{root})	Major GO term	$W_{GO}(j,n)$	$p_{weight}(j,n)$
Cluster 3	Response to water derivation	4.11	16.6
(24.81)	Regulation of transcription, DNA-dependent	3.16	12.7
	Carboxylic acid metabolism	2.82	11.4
	Protein amino acid phosphorylation	2.63	10.6
Cluster 4	Protein amino acid phosphorylation	8.34	23.1
(36.03)	Carboxylic acid metabolism	3.58	9.9
	Response to abiotic stimulus	3.35	9.3
	Regulation of transcription, DNA-dependent	2.44	6.8
Cluster 6	Regulation of transcription, DNA-dependent	1.99	23.5
(8.48)	myo-inositol biosynthesis	0.95	11.2
	Abscisic acid mediated signaling	0.83	9.8
	Protein amino acid phosphorylation	0.57	6.7
Cluster 7	Carbohydrate metabolism	3.02	22.2
(13.58)	Cell surface receptor linked signal transduction	1.71	12.6
	Nucleobase, nucleoside, nucleotide, and nucleic acid metabolism	1.62	11.9
	Protein amino acid phosphorylation	1.59	11.7
Cluster 15	Regulation of transcription, DNA-dependent	1.32	52.4
(2.52)	Electron transport	0.7	27.8
Cluster 19	Cell-cell signaling	0.78	23.5
(3.32)	Response to auxin stimulus	0.68	20.5
	Protein folding	0.65	19.6
	N-terminal protein myristoylation	0.61	18.4
Cluster 21	Carbohydrate metabolism	2.93	29.1
(9.71)	Response to gibberellic acid stimulus	1.86	19.2
	Photosynthesis, dark reaction	0.91	9.4
Cluster 22	Protein amino acid phosphorylation	6.74	28.4
(23.76)	Macromolecule biosynthesis	3.38	14.2
	Regulation of transcription DNA-dependent	2.50	10.5
	Signal transduction	2.30	9.7
Cluster 23	Response to endogenous stimulus	2.79	60.5
(4.61)	Response to biotic stimulus	1.83	39.7
Cluster 25	Carboxylic acid metabolism	8.19	20.9
(39.16)	Response to pest/pathogen/parasite	5.66	14.5
	Lipid biosynthesis	3.55	9.1
	Transport	3.52	9.0
Cluster 28 (0.95)	Carbohydrate metabolism	0.95	100

Table 4-4 Summary of molecular function for each cluster

MF	Cluster Index											
Level 2	**3**	**4**	**6**	**7**	**15**	**19**	**21**	**22**	**23**	**25**	**27**	**28**
Carbohydrate binding	1.3	3.6	0	3.4	0	0	0	3.8	0	0	0	0
Nucleic acid binding	23.7	10.4	41.6	4.4	55.6	0	0	16	0	3.5	0	0
Nucleotide binding	6.14	7.5	0	12.4	0	0	17.6	0	0	4.8	0	0
Protein binding	0	0	0	0	0	0	10.4	0	0	2.8	0	0
Oxygen binding	5.5	4	0	0	15.4	0	0	0	0	9.3	0	0
Lipid binding	0	1.6	0	0	0	0	0	2.7	0	3.4	0	0
Metal ion binding	2.9	1.4	8	0	0	0	0	0	0	0	0	0
Kinase activity	27.2	15.7	6.4	0	0	0	0	0	0	12	0	0
Transferase activity	23.3	19.6	0	28.2	11	0	0	59.3	0	8.6	0	0
Hydrolase activity	15.8	12.8	16.5	27	0	71.9	32.1	8	0	8.1	0	100
Oxidoreductase activity	8.2	8.3	16.8	6	0	0	23.8	11	0	14.1	0	0
Signal transducer activity	0	0	0	7.8	0	28.5	0	0	0	0	0	0
Isomerase activity	0	0	10.7	0	0	0	0	0	0	0	0	0
Transcription regulator activity	14.2	5.7	25.6	0	27.9	0	0	8.9	0	3.5	0	0
Transporter activity	0	1.5	0	0	17.8	0	23.8	0	0	5.4	0	0

4.6 Conclusions and future work

Fuzzy logic can be applied to all aspects of gene regulatory network analysis from clustering to assessing network credibility. Multi-scale Fuzzy K-means clustering provides the cluster information in different scales and captures interactions in terms of gene function and across regulatory pathways. It makes the results more reliable. The regulatory network construction algorithm uses the cluster centers efficiently to evaluate the time delay information. The algorithm also allows feedback in the networks, which most qualitative regulatory network algorithms cannot provide at present. Visualizing the cluster relationships helps show biological interactions. GO and pathway evaluations indicate the algorithm is promising and demonstrate that it yields detailed biological hypotheses of the regulatory connections with known metabolic networks. Future work will focus on integrating the regulatory network model with existing metabolic networks to simulate cellular processes.

CHAPTER 5. GENETIC NETWORK INFERENCE BASED ON TIME SERIES EXPRESSION PROFILES[2]

5.1 Introduction

Genetic network inference infers gene relationships based on mRNA accumulation levels. Due to transcription regulation and mRNA degradation, the mRNA accumulation levels dynamically change over time. This dynamic information reflects the internal regulation mechanisms, and is crucial for inferring gene regulatory relationships (Bar-Joseph 2004). Time series microarraky data is a series of mRNA accumulation level measurements sorted in time order.

The profile length of most publicly available time series data is usually less than 20 samples. This is not enough to estimate parameters for detailed models such as differential equation models (Chen, He et al. 1999; Smolen, Baxter et al. 2000). Bayesian networks (Friedman, Linial et al. 2000) and correlation-based models, like coexpression networks (Stuart, Segal et al. 2003; de la Fuente, Bing et al. 2004; Magwene and Kim 2004) and GGM(Graphical Gaussian Models) (Kishino and Waddell 2000; Toh and Horimoto 2002; Schafer and Strimmer 2004) treat time series data as if it were static data, i.e., treat the time samples as if they were independently distributed. Dynamic Bayesian Networks (Murphy 1999; Murphy 2002; Perrin, Ralaivola et al. 2003; Kim, Imoto et al. 2004; Zou and Conzen 2005) are an extension of Bayesian networks that are designed for time series data. Most dynamic Bayesian network models are based on discretized expression values that result in the information loss. Also, most dynamic Bayesian networks only consider first-order Markov relationships, i.e. the transition matrix only considers the connections between the adjacent time slices. This is not true in reality, because the time delay of gene interactions can vary over a wide range and the corresponding shift number of the time index is directly related with the sample interval. Dynamic Bayesian networks can be adapted to higher order

---

[2] This chapter is the extended version of the paper: Du, P., E. S. Wurtele, et al. (2005). "Genetic Network Inference based on Time Series Expression Profiles." Bioinformatics **(submitted)**.

Markov relation and continuous values at the expense of increased computation that is not suited for data with a short time profile, especially in large scale network inference. The model proposed in this work extends correlation-based models to time series data that uses the actual time differences.

Due to delays in gene regulation and transcription responses, time delay information is important for resolving the causal relationships in gene regulation. (Arkin, Shen et al. 1997) first proposed using time delay information in construction of reaction pathways. (Kato, Tsunoda et al. 2001; Shaw, Harwood et al. 2004) described using time delay information to infer genetic networks. One of the important tasks in genetic network inference is to differentiate direct and indirect interactions, which can be treated as equivalent to determining the conditional independence among variables. GGM is a multivariate analysis that infers the variable relationships using the idea of conditional independence among variables (Edwards 2000). GGM cannot check the conditional independence when the conditional variables include common descendents (de la Fuente, Bing et al. 2004). Instead, d-separation (directed-separation) theory (Pearl 2000; Shipley 2002) checks the conditional independence among variables based on the network topology. (de la Fuente, Bing et al. 2004) proposed the use of partial correlation and d-separation theory to differentiate direct and indirect interactions. However, the d-separation check in (de la Fuente, Bing et al. 2004) has problems with a high edge false deletion rate.

Another important task in genetic network inference is identifying the feedback loops. Feedback plays an important role in the control of the biological systems. Most current models and algorithms do not detect feedback loops. Correlation-based models allow feedback, however, existing algorithms like GGM and de la Fuente et al.(2004), which are based on undirected graphs, do not perform well in dealing with cycles, as shown in our simulation results.

The constraint-based time-correlation (CBTC) algorithm uses time correlation to estimate the time delays and edge directions, and then uses partial correlation and d-separation theory to tell the direct and indirect interactions. By combining the time delay and edge direction information during the d-separation check, feedback loops can be easily identified. In the CBTC algorithm, edge direction information with additional constraints such as time delay

and sign of the path decreases the number of false deletions. The algorithm was evaluated with simulated data and yeast cell cycle data and compared with other algorithms. The results show the CBTC algorithm is effective at differentiating direct and indirect interactions and identifying feedback loops.

5.2 Methodology

5.2.1 Network model

The linear genetic network model (D'Haeseleer, Liang et al. 1999; van Someren, Wessels et al. 2001) with time delays takes the form:

$$x_i(t) = g(\sum_{j=1}^{J} w_{ji} x_j(t - \tau_{ji})) + n_i(t) \qquad (5\text{-}1)$$

where g(.) is a monotonic function, $x_i(t)$ is the gene expression value of gene X_i at time t, w_{ji} is the weight of gene X_j contributes to the expression value of gene X_i, τ_{ji} is the time delay of gene X_j activated (or depressed) by gene X_i, $n_i(t)$ models other unconsidered elements and the internal transcription noise of gene X_i at the time t. The gene expression profile $\mathbf{x}_i$ is a series of time samples of $x_i(t)$.

Linear genetic network inference estimates the weight w_{ij}. This estimation is often simplified to a binary decision to detect if the weight is zero or non-zero $w_{ij} \in \{-1, 0, 1\}$; i.e., whether there is a link (positive or negative) between genes X_i and X_j or not. This is equivalent to determining whether there is a linear or approximately linear relationship between gene expression profiles $\mathbf{x}_i$ and $\mathbf{x}_j$. The Pearson correlation can estimate the linear relationship between two variables. The Pearson correlation coefficient, r, equals 1 or -1 when two variables x and y have a linear relationship: $y = kx + b$. The sign of r is the same as k; r close to 0 shows that x and y do not have linear relationship. Therefore, instead of fitting the linear model in equation (5-1), we just need to calculate the pairwise Pearson correlation to detect if a linear relationship between gene expression profiles. This greatly simplifies the network inference and reduces the requirement of long time profiles.

5.2.2 Determine the time delay and edge directions

Correlation can find the linear relationships between gene expression profiles, but it cannot tell the direction of causality. Knowledge of the time delay can help determine the direction of causality. Time correlation can estimate time delay τ_{ij} over l sample intervals:

$$r_{ij}(\tau) = r_{ij}(l\Delta t) = \mathrm{cov}(\mathbf{x}'_i, \mathbf{x}'_j) / \sqrt{\mathrm{var}(\mathbf{x}'_i)\,\mathrm{var}(\mathbf{x}'_j)} \tag{5-2}$$

$$\mathbf{x}'_i[k] = \mathbf{x}_i[k],\ \mathbf{x}'_j[k] = \mathbf{x}_j[k+l], k = 1,\ldots,N, 1 \le k+l \le N$$

where l is the number of sample intervals shifted between gene expression profiles $\mathbf{x}_i$ and $\mathbf{x}_j$, the real time delay, $\tau = l \cdot \Delta t$, Δt is the sample interval, N is the profile length, $\mathbf{x}'_i$ and $\mathbf{x}'_j$ represent shifted profiles of $\mathbf{x}_i$ and $\mathbf{x}_j$.

The shift number l corresponding to $\max|r_{ij}(\tau)|$ can be used to estimate the time delay, τ_{ij}, from i to j. Given a correlation threshold T_r, if $\max|r_{ij}(\tau)| > T_r$, we assume there is an edge between i and j. The edge direction is determined by τ_{ij}. If $\tau_{ij} > 0$, then there is an edge from i to j; if $\tau_{ij} < 0$, then there is an edge from j to i; if $\tau_{ij} = 0$, then we cannot determine the edge direction or we can treat the edge as bi-directional. The correlation threshold T_r can be determined based on the statistic significance of correlation. The statistic for the correlation coefficient test ($H_0 : r = 0$) can be defined as:

$$t = r\sqrt{df/(1-r^2)} \tag{5-3}$$

where r is the correlation coefficient, df is the degree of freedom of the t-distribution. For the standard correlation, $df = N-2$, where N is the sample number. For time correlation (not circular time correlation), $df = N-l-2$. The partial correlation coefficient, described later, has the same distribution as standard correlation (Hotelling 1953) with $df = N-l-k-2$, where k is the order of partial correlation. A p-value can then be converted into a corresponding correlation threshold T_r based on t-distribution and equation (5-3).

This algorithm limits time delay within a positive finite range, i.e. $0 \le \tau \le D$. This will ensure that each pair of nodes has edge estimations in both directions. If the time delay in one

direction is zero, the algorithm assumes that the edge with strongest correlation is correct. If the time delays in both directions are zero, then the edge is treated as bi-directional with 0 delay. Limiting the maximum delay $(\tau \le D)$ reduces the chance of false discovery. Typically, $D \le N/2$, where N is the profile length.

5.2.3 Differentiating direct and indirect interactions

If gene X and gene Y have similar expression profiles, there are several possibilities: (1) X directly interacts with Y or vice versa. (2) X and Y interact through other intermediates. (3) X and Y have a common cause. (4) No causal relationship, just coincidence. Here, the interaction may be through some other unobserved intermediates, like proteins and etc. Case (4) may be treated by defining a correlation significance threshold. An important question is how to differentiate case (2) or (3) from case (1), i.e., tell the indirect correlations from the direct ones.

The partial correlation coefficient measures the linear relationship between two variables after controlling for conditioning on other variables. The order of the partial correlation is the number of conditional variables. The first-order partial correlation coefficient between variables X and Y conditioned on variable Z can be calculated by:

$$r_{XY|Z} = (r_{XY} - r_{XZ}r_{YZ}) / \sqrt{(1-r_{XZ}^2)(1-r_{YZ}^2)} \qquad (5\text{-}4)$$

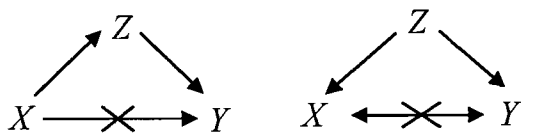
Higher order partial correlations can be computed in similar ways (Shipley 2002).

For time correlation, r_{XY}, r_{XZ} and r_{YZ} in equation (5-4) are replaced by time correlation coefficients with τ set as the estimated time delay τ_{XY}, τ_{XZ} and τ_{YZ}.

In GGM, the variable relationships are characterized by a partial correlation matrix. The partial correlation coefficients describe the correlation between any two variables conditioned on all the remaining variables, which can be computed by the inverse of the standard correlation coefficient matrix (Edwards 2000). If the partial correlation coefficient is close to zero, then two variables are considered to be conditionally independent, i.e., the correlation between two variables is indirectly caused by the conditioned variables. D-separation theory states that two variables are dependent when they are conditioned on their common descendents (Pearl 2000; Shipley 2002). Thus GGM cannot check the conditional

independency when the conditional variables include their common descendents. As a result, some indirect interactions cannot be detected. This will be severer when the network having strong feedbacks. Instead, de la Fuente et al. (2004) proposed combining partial correlation and d-separation theory (de la Fuente, Bing et al. 2004) to identify indirect interactions in undirected genetic networks.

D-separation theory can be used to determine the causal independence of two nodes upon conditioning on a third node set (Shipley 2002). If two variables are d-separated when conditioned on a third node set, their corresponding partial correlation coefficient is close to zero. But the converse may not hold. A zero partial correlation coefficient does not always imply d-separated. Instead of conditioning on all other variables in the network as in GGM, d-separation theory selects the conditional variables based on the network topology. For genetic network inference, only two situations need to be considered as shown in Figure 5-1. The correlation of variables X and Y is caused through intermediate cause Z or caused by a common cause Z. If the partial correlation of X and Y conditioned on Z is close to zero, then we assume that X and Y are d-separated, i.e., the correlation between X and Y is indirectly caused by Z. As a result, edge XY can be deleted as shown.



(a) Intermediate cause **(b) Common cause**

Figure 5-1 d-separation types

In practice, the link between X and Z or Y and Z can be through multiple steps and Z can represent multiple variables. For the "intermediate cause" d-separation case, we assume our model satisfies Markov conditions. In the case of multiple steps, only the node directly linking the end node Y needs to be considered. Also, deleting the edge XY does not affect the connectivity between node X and Y, i.e., there is still a path linking node X and Y. This property still holds for the whole graph after the "intermediate cause" d-separation check. If two nodes are accessible before the "intermediate cause" d-separation check, they will also be accessible after the check. But this is not true for "common cause" d-separation. If an edge is deleted by "common cause" d-separation check, the two nodes connected by the edge

perhaps may not be linked any more. Thus, false deletion during the "common cause" d-separation check can cause problems in the interpretation of the resultant networks.

In order to reduce the false deletion rate, constraints can be added to the conditional variables to avoid unnecessary d-separation checks. For example, given the "intermediate cause" d-separation check of edge XY as shown in Figure 5-2, then r_{XY} and $r_{XZ}r_{ZY}$ should have the same sign. The delay of path X→Z→Y: $\tau_{XZ}+\tau_{ZY}$ needs to be estimated. Only the nodes Z satisfying $|\tau_{XZ}+\tau_{ZY}-\tau_{XY}|<\tau_{error}$, where τ_{error} represents the estimation error of path delay, are used. For "common cause" d-separation, we do the similar processing.

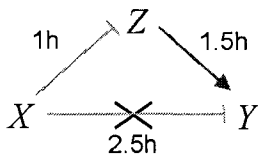


Figure 5-2 Constraints in d-separation check, the red edge with bar head represents negative correlation, the blue edge represents positive correlation, edge labels represent time delays.

Another case to be considered is the possible conflict between "intermediate cause" and "common cause" d-separation checks. For the case shown in Figure 5-2, both the "intermediate cause" d-separation check of edge XY conditioned on node Z, i.e., compute $r_{XY|Z}$, and the "common cause" d-separation check of edge ZY conditioned on node X, i.e., compute $r_{ZY|X}$ need to be computed. It is possible both $r_{XY|Z}$ and $r_{ZY|X}$ are less than the significant threshold. In that case, we delete the edge with the lowest partial correlation coefficient and keep the other.

In undirected graphs, we cannot tell the difference between the feedback cycle shown in Figure 5-3 and the d-separation cases shown in Figure 5-1. As a result the feedback loops may be deleted by mistake during the d-separation check.
Since time correlation learns the edge directions, the feedback loops shown in Figure 5-3 will not be eliminated by the d-separation check because it does not satisfy the "intermediate cause" or "common cause" d-separation conditions.

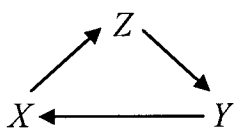


Figure 5-3 A feedback cycle

5.2.4 Algorithm for genetic network inference:

Table 5-1 presents the CBTC genetic network inference algorithm. In the algorithm, the "intermediate cause" d-separation check is done first because it will not affect the overall connectivity of the graph. Because all the processing is done sequentially, if some edges are falsely deleted in previous steps, it will affect the subsequent steps. In order to reduce this chance, two thresholds: T_{pCor1} and T_{pCor2} with $T_{pCor1} < T_{pCor2}$ are used (default p-values of T_{pCor1} and T_{pCor2} are 0.2 and 0.1). After finishing all d-separation checks with T_{pCor1}, post processing of the network with the higher partial correlation coefficient threshold T_{pCor2} is performed. During post processing, we only delete the edges with "intermediate cause" d-separation partial correlation coefficient less than T_{pCor2} and the deletion of the edge will not affect the network connectivity.

Table 5-2 shows the detailed d-separation check algorithm. The algorithm treats the edges with zero time delay as bi-directional edges. In the case of undirected graph, all edges are bi-directional. When deleting a d-separated edge, we delete the edges in both directions, i.e. both XY and YX.

Table 5-1 CBTC genetic network inference algorithm

1. Compute pairwise time correlation as shown in equation (5-2), estimate the time delay τ_{ij} and time correlation coefficient, and get a time correlation matrix R, in which each element r_{ij} represents time correlation coefficients with time delay τ_{ij};
2. Select significant correlated edges above a certain correlation coefficient threshold T_r (default p-value is 0.05);
3. Sort the significant edges from the weakest to strongest (based on $|r_{ij}|$);

For each significant edge in the network, perform "intermediate cause" d-separation check, as shown in

4. Table 5-2, with partial correlation coefficient threshold T_{pCor1} and record parameters;

For each significant edge remained in the network, perform "common cause" d-separation check, as shown in

5. Table 5-2, with partial correlation coefficient threshold T_{pCor1} and record parameters;
6. Post processing with partial correlation coefficient threshold T_{pCor2} and output the final network.

Table 5-2 Algorithm of d-separation check of edge XY

1. Find alternative paths of edge XY
   a. For "intermediate cause" d-separation check, find all nodes Z_i directly pointing to node Y, among them select those which node X can access within N steps (suppose the indirect induced correlation through larger than N steps is rare, by default N = 4);
   b. For "common cause" d-separation check, find all nodes Z_i which are accessible to node X and Y within N steps without going through edge XY;
2. Filtering the candidate conditional nodes Z_i with sign and delay constraints τ_{error};
3. Sort Z_i based on $|r_{XZ_i} r_{Z_iY}|$ (for "intermediate cause") or $|r_{Z_iX} r_{Z_iY}|$ (for "common cause") from the strongest to weakest;
4. Compute the first order partial correlation $r_{XY|Z_i}$, if the partial correlation coefficient close to 0 (less than significant threshold T_{pCor1}), delete edge XY and go to step 7, or else continue step 4 until check all Z_i;
5. Increase the partial correlation order by 1 and compute higher order partial correlation conditioned on the nodes Z_i. If it is close to 0, delete edge XY and go to step 7, or else continue step 5 until the partial correlation order is larger than M (suppose higher order partial correlation is rare, by default M is 3);
6. Mark edge XY as a direct interaction between X and Y;
7. Save the partial correlation information.

Note: For computational efficiency, in step 1, an access matrix recording the shortest path length between nodes is created before the d-separation check, so the path search just needs to check this matrix. In step 5, we do not consider all combinational possibilities of higher order partial correlation, we increase the order by gradually adding one conditional variable at a time in decreasing order of $r_{XY} - r_{XY|Z_i}$, which represents the effect of correlation coefficient by the conditional variable Z.

5.3 Results

5.3.1 Simulation Results

(de la Fuente, Bing et al. 2004) evaluated the performance of the d-separation check algorithm for undirected graphs on large scale artificial networks. This study focuses on the evaluation of the two types of d-separation check and feedback loop identification in directed graphs. The simulation is based on the linear model in equation (5-1) in discrete form and with measurement noise.

$$\mathbf{x}_i[k] = g(\sum_{j=1}^{J} w_{ji}\mathbf{x}_j[k-l_{ji}]) + n_{ik} \quad (5\text{-}5)$$

$$\mathbf{x}'_i[k] = \mathbf{x}_i[k] + m_{ik}$$

$$g(x) = 2/(1+\exp(-2x)) - 1$$

where $\mathbf{x}'_i$ is the transcription level measured by microarray chips, $\mathbf{x}_i$ is the real transcription level and m_{ik} is the normally distributed measurement noise, $g(.)$ is a hyperbolic tangent sigmoid transfer function, and the other parameters are the same as equation (5-1).

Figure 5-4 shows the network topology of the simulation. The edges with arrow heads represent activation, and the ones with bar heads represent inhibition. The network includes positive and negative feedback loops, duplicate paths, "intermediate cause" and "common cause" d-separation cases. All of them are typical in real biological systems. For simplicity, we set all the edge weights $w_{ij} = 1$, time delays $\tau_{ij} = 1$, sample interval $\Delta t = 1$, n_{ik} is N(0, σ_n^2), by default $\sigma_n = 0.2$ and m_{ik} is N(0, σ_m^2), by default $\sigma_m = 0$.

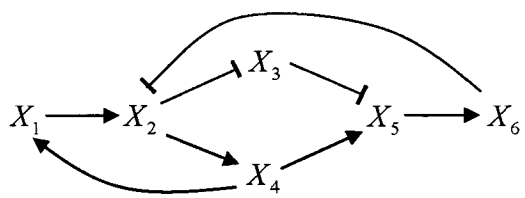


Figure 5-4 Simulation network topology. The edges with arrow ends represent activation, bar ends represent depression.

The network shown in Figure 5-4 is a self-oscillation system. With the initial values of x_i are set as 0.5, the system will automatically produce periodic oscillation signals, just like some periodic biological process, e.g. cell cycle process. The simulated data was extracted after the system in stable oscillation state (after 20 iterations).

To evaluate the performance of the algorithm, we iterated the procedure of producing simulation data and creating networks for K (K = 1000) times under each particular parameter setting. Then we computed the detection rate of the edge (the count of detecting the edge divided by the iteration number K). The sensitivity of the algorithm is defined as the proportion of true links identified among all true links and is average over all K iterations. The specificity of the algorithm is defined as the proportion true negative links of all the

negative links ($J^2 - J$ for directed graph and $(J^2 - J)/2$ for undirected graph, where J is the number of nodes in the network) and is averaged over all K iterations.

Effects of profile length

For the commonly used time series expression data, the length of the profile usually ranges from 10 to 20, or less. Thus, it is important to check how the profile length will affect the network inference performance. Table 5-3 shows the results of different profile length N under default parameter settings. The time correlation delay range D is [0, 2]. The p-value of T_r is 0.001. In order to keep the detection rate high, we set the maximum correlation coefficient threshold T_r as 0.7, i.e., if the T_r corresponding to the p-value is larger than 0.7, we set $T_r = 0.7$. From the results, it can clearly be seen that the performance becomes worse as signal length becomes shorter, especially for edges X_1X_2 and X_4X_5. One reason is that the statistical significance of correlation becomes lower when the profile length becomes shorter. In other words, noise and other interference become more likely to induce the correlation

Table 5-3 Results of different profile length *N*

Edge	Signal Length (N)				Interpolation			
	64	**32**	**16**	**8**	**8 (15) (linear)**	**8 (22) (linear)**	**8 (15) (spline)**	**5 (13) (linear)**
X_1X_2	65.0	51.5	46.5	44.4	50.3	55.1	41.0	50.9
X_2X_3	100	99.4	91.5	75.0	81.5	85.3	70.7	65.1
X_2X_4	100	99.2	91.7	72.6	80.2	84.6	67.8	66.0
X_3X_5	98.6	91.2	74.9	55.5	58.3	64.3	55.2	57.5
X_4X_1	100	97.7	85.1	63.8	68.1	74.0	60.5	57.2
X_4X_5	92.6	76.4	57.6	42.9	53.8	58.0	46.6	55.9
X_5X_6	100	99.8	96.1	76.3	81.0	82.7	78.5	73.3
X_6X_2	86.3	70.8	72.5	67.4	64.8	67.0	60.5	88.3
X_3X_4	27.0	24.1	30.5	49.3	48.7	53.5	61.8	71.4
X_2X_5	16.4	20.8	27.0	31.3	43.4	48.4	36.9	30.4
Sens.	92.8	85.8	77.0	62.2	67.3	71.4	60.1	64.3
Spec.	80.3	85.4	87.3	82.2	78.1	72.7	75.0	67.5

The number shown in the table is the percentage of the detection rate over 1000 iterations. The shaded rows represent the edges not existing in the network. In the title line of right "Interpolation" part, the first number is the profile length before interpolation, the number in the parenthesis is the length after interpolation. "linear" represents linear interpolation method, "spline" represents spline interpolation method, "Sens." represents sensitivity and "Spec." represents specificity.

with same significance by chance. Another reason is that the time delay estimation becomes more unreliable and inaccurate when the profile length becomes short, especially when there are interactions among multiple variables, like node X_2 and X_5. But even when signal length $N = 8$, the detection rate of most edges is still larger than 50%. The simplified network can still help users capture the major interactions and the general view of genetic networks, especially in large scale network inference. The specificity of $N = 64$ and $N = 32$ are lower than that of $N = 16$. The reason is that we use the same p-value for all of them. In practice, the p-value should be adjusted based on the profile length. If the user cares more about the sensitivity, lower p-value (higher correlation threshold) should be adopted, especially when the profile length is short.

For short profiles, the profile can be interpolated to extend the profile length. The results are also shown in Table 5-3. When signal length $N = 5$, the result after interpolation is not bad, but the algorithm specificity is also decreased. Interpolating more than one sample within one sample interval gives a slight increase of algorithm sensitivity with a decrease of specificity. That means interpolation does help to improve the performance at the expense of the decreased algorithm specificity because interpolation brings uncertainties in the sample values. Linear interpolation performed better than "spline" interpolation method for this data set.

Effects of time delay estimation

Time delay estimation will affect the network inference results. Table 5-4 shows how delay range [0, D] and measurement noise affect the results. From Table 5-4, as the delay range D decreases, the algorithm specificity increases. This means if the user has prior knowledge of the time delay range of biological systems, the algorithm will work better by limiting the potential solutions. Unfortunately, we still know little of the real time delays of transcription, translation, and different kinds of regulation. Table 5-4 also shows the effects of measurement noise. By comparing the results with and without correct delay information, which is estimated based on the simulation network topology, we can tell measurement noise affects the time delay estimation and network inference. This means normalization, as a preprocessing step to reduce the measurement noise, is important.

Table 5-4 Effects of time delay estimation under different settings (signal length $N = 16$)

Edge	Delay range [0, D]				Measurement noise	
	Correct delay	$D = 2$	$D = 3$	$D = 4$	($\sigma_m = 0.1$)	($\sigma_m = 0.1$) Correct delay
X_1X_2	63.2	46.5	42.1	40.4	32.0	39.9
X_2X_3	100	91.5	89.8	90.0	94.9	94.3
X_2X_4	99.3	91.7	89.8	90.8	92.2	89.8
X_3X_5	84.0	74.9	72.0	71.4	75.3	80.8
X_4X_1	89.0	85.1	85.0	82.7	34.1	70.9
X_4X_5	72.4	57.6	58.8	61.1	54.0	73.1
X_5X_6	95.0	96.1	94.9	93.0	80.4	80.6
X_6X_2	74.3	72.5	67.1	73.0	66.5	55.8
X_3X_4	6.4	30.5	33.5	32.9	13.9	11.7
X_2X_5	31.8	27.0	26.6	29.8	20.3	33.9
Sens.	84.7	77.0	74.9	75.3	66.2	73.2
Spec.	90.9	87.3	86.2	85.5	83.0	87.2

The column named as "Correct delay" represents the result with correct time delay info estimated based on simulation network topology. For other columns, the time delays were estimated by time correlation. Other notations are the same as Table 5-3.

Comparison with other algorithms

Finally, we compare the CBTC algorithm with GGM and undirected d-separation check algorithm, as shown in Table 5-5. Because GGM requires the correlation matrix to be positive definite, we select a longer signal length $N = 32$. In order to remove the effects of time delay estimation error, we used correct delay information to estimate the time correlation coefficients. The time correlation matrix was modeled as a symmetric matrix (equivalent to an undirected graph) by only keeping the strongest correlation, i.e., $\max\left(|r_{ij}|,|r_{ji}|\right)$. The p-value corresponding to T_r is 0.001. For GGM, the p-value of the partial correlation threshold is 0.05.

Table 5-5 shows that the performance of the CBTC "Directed" algorithm is much better than the "Undirected" and GGM algorithms. For the "Undirected" algorithm, the major problem is the detection rate of feedback links like X_1X_2 and X_6X_2 is very low. Because undirected algorithms cannot tell the difference between the feedback loops and "intermediate cause" or "common cause" d-separation cases, the feedback loops tend to deleted by mistake. The problem with GGM is the low specificity. Indirect links like X_3X_4, which is caused by "common cause" X_2, are rarely deleted, because the conditional variables

include the common descendents of X_3 and X_4. Table 5-5 also compares CBTC algorithm with standard correlation. The result of standard correlation is poor as the time profiles are misaligned. This indicates the potential problem of using standard correlation for time series data.

Table 5-5 Comparing CBTC with other algorithms (signal length $N = 32$)

Edge	Directed	Undirected	GGM	Standard Correlation
X_1X_2	82.8	9.5	87.6	0
X_2X_3	100	93.5	91.7	1.4
X_2X_4	100	89.8	63.9	0.6
X_3X_5	92.3	49.9	43.5	0.6
X_4X_1	99.7	97.3	73.7	0.7
X_4X_5	86.4	51.3	81.8	0.8
X_5X_6	99.4	98.7	93.0	0.3
X_6X_2	84.2	15.4	41.8	5.9
X_3X_4	4.2	44.7	97.9	99.7
X_2X_5	63.2	45.0	47.8	0
Sens.	93.1	63.2	72.1	1.3
Spec.	91.3	89.5	71.7	86.5

Column "Directed" is the result using the CBTC algorithm in this work,, the column "Undirected" is the result without using time delay information, which is equivalent to the algorithm proposed in (de la Fuente, Bing et al. 2004). GGM column is the result using Graphical Gaussian Modeling (GGM) algorithm. Other notations are the same as Table 5-3.

5.3.2 Results using yeast cell cycle microarray data

We used the public yeast (*Saccharomyces cerevisiae*) mitotic cell cycle data set (Cho, Campbell et al. 1998). The data was synchronized by arresting cdc28-13 cells in late G1. 17 time points with 10 min intervals were collected. The Affymetrix Genechip was used for measuring mRNA accumulation levels. Our analysis was based on 140 genes listed at the paper companion website (Simon, Barnett et al. 2001), including both cell cycle TFs (Transcriptional Factor) and their target genes. The relationships between the TF and the target genes were identified using genome-wide location analysis(Simon, Barnett et al. 2001).

Figure 5-5 shows the expression profiles of transcriptional factors SWI4, MCM1 and some of their target genes. We can see the time delays between transcriptional factors and their regulated genes. The delays between SWI4 and its regulated genes are quite small. Improving the results requires either taking more samples or decreasing measurement noise. For MCM1, the delays are very obvious. In this case, standard correlation methods without

time delay will give misleading results between MCM1 and its regulated genes, especially for gene YOR066W. This indicates the importance of using time correlation for time series expression profiles.

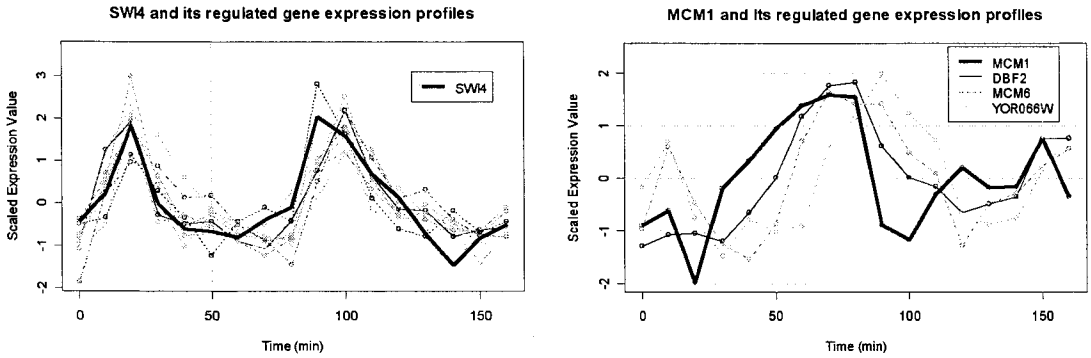


Figure 5-5 Expression profiles of Transcriptional Factors and their regulated genes. (Only shows genes with time correlation coefficient larger than 0.8)

As shown in Figure 5-5, the expression profiles of SWI4 regulated genes are highly correlated. The situations are similar for some other genes. Considering the measurement noise and limited sample number, it is hard to tell the difference between the expression profiles of these highly correlated genes. Therefore, we do clustering first, and suppose the genes within the same clusters having similar expression profiles and each cluster can be treated as a single entity. Then we can infer the genetic networks based on the cluster center profiles instead of individual genes'. Another advantage of using cluster centers is suppressing of noise. Because the cluster center profile is equivalent to a signal passing a lower pass filter, the faster changing noise will be filtered and minimized. In this work, we adopted the Multi-scale Fuzzy K-means algorithm to cluster the data (Du, Gong et al. 2005). Multi-scale Fuzzy K-means is a clustering algorithm particularly designed for genetic network inference. The clustering parameter, Gaussian window scale, is set to 0.3. This gave 6 clusters over 140 genes. The genetic network inference was based on the 6 cluster center profiles shown in Figure 5-6.

Figure 5-7 shows the inferred network based on cluster center profiles. The network inference parameters were set as following. The p-value of T_r was set as 0.05, p-values of threshold T_{pCor1} and T_{pCor2} were set as 0.2 and 0.1 respectively; the time delay range [0, D]

was limited as [0, 40min] (half cycle period). Figure 5-7.a shows the network before the d-separation check, i.e. shows all the links with correlation coefficients larger than T_r. Figure 5-7.b shows the network after the d-separation check with $\tau_{error} = 0$ minutes. For this time error, the d-separation check requires that the alterative paths have exactly the same time delay. Considering the time delay estimation error and sampling interval, this restriction was loosened to be $\tau_{error} = 10$ minutes giving the network shown in Figure 5-7.c.

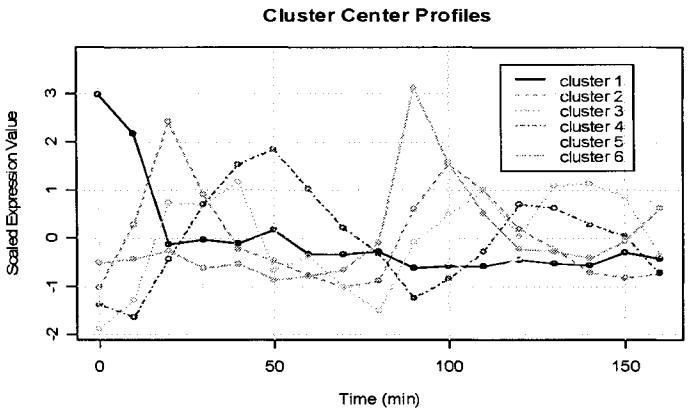


Figure 5-6 Cluster center expression profiles
(The profiles were standardized as 0 mean and 1 standard deviation.)

The d-separation check simplifies the network as can be seen in Figure 5-7 (a-c). First, we will check the positive cycles in the network. Figure 5-7.c is the simplest network, it clearly shows one positive cycle 2 → 4 → 5 → 6 → 2. The period of the cycle is 70 min, which is nearly the same as the real yeast cell cycle period, around 80 min. Figure 5-7.b is the network with more severer constraint, it indicates three more cycles 2 → 4 → 5 → 2 (period = 80 min), 2 → 3 → 5 → 2 (period = 90 min) and 2 → 3 → 5 → 6 → 2 (period = 80 min). Figure 5-8 shows the frequency distribution of the genes in different cell cycle stages. The cell cycle stage information came from the paper companion website (Cho, Campbell et al. 1998). Because the development stage information for the genes in cluster 1 is not available, it is not shown. Next, we will illustrate the cell cycle development stage information based on the network shown in Figure 5-7.b. Figure 5-8 indicates the majority of cluster 2 is in *late G1*; cluster 2 activates cluster 3 with a 10min delay, which corresponds to the *S phase*, and cluster 2 activates cluster 4 with a 20min delay, which corresponds to the *S phase* and *G2 phase*. Clusters 3 and 4 activate cluster 5, which corresponds to the *M phase*.

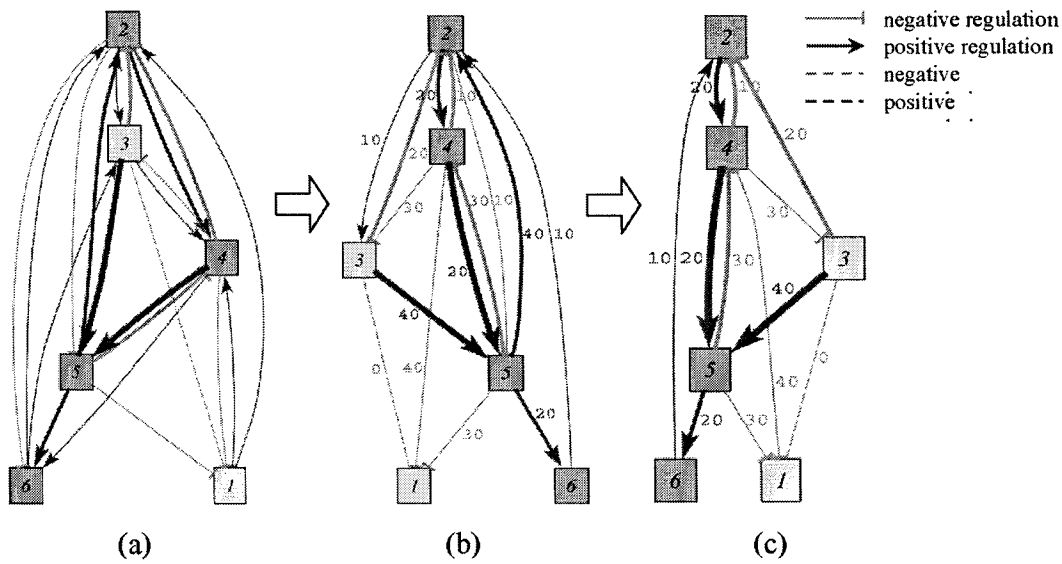


Figure 5-7 Inferred genetic networks based on 6 cluster center profiles. (a) Genetic network before d-separation check; (b) Genetic network after d-separation check (τ_{error} = 0min); (c) Genetic network after d-separation check (τ_{error} = Δt = 10min). The width of the line represents the significance of correlation. The wide line represents the p-value of correlation coefficient is less than 0.0001, the mid-wide line represents between 0.0001 and 0.001, the thin line represents larger than 0.001. The number within the node box represents the cluster index. The edge label represents the time delay (in minutes) between the nodes the edge connecting. The node filled with red color represents the corresponding cluster has more than 15 elements, brown color represents having 6 to15 elements.

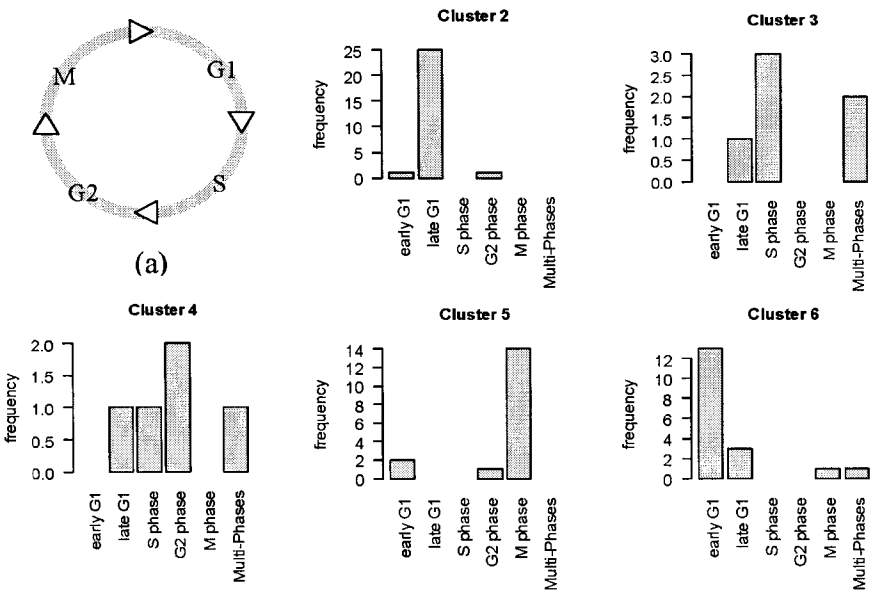
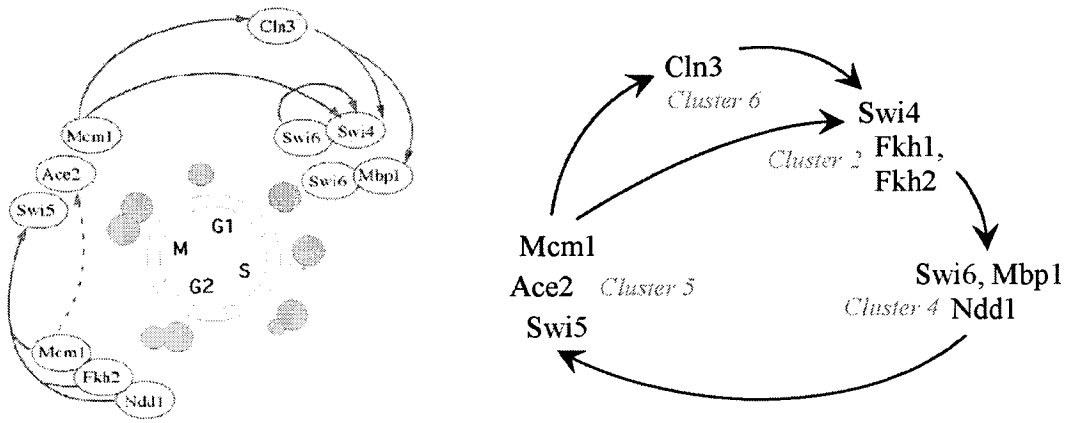


Figure 5-8 Frequency distribution of the genes in different cell cycle stages in the same cluster

Cluster 5 activates cluster 6 after a 20min delay, which corresponds to the early *G1 phase*. Finally cluster 5 activates cluster 2 with a 10min delay and the next cycle begins. Comparing with the yeast cell cycle stages shown in Figure 5-8.a, we can see the development stage information matches the cycle we found. Interestingly, the network in Figure 5-7.c also identifies another two strong negative feedback loops, each with two clusters: clusters 2 and 4 and clusters 4 and 5. Negative feedback loops are important biological regulation mechanisms. The network clearly indicated that the genes in *late G1* activated genes in *S* and *G2 phase*, then some of these activated genes in return depressed the genes in *late G1*. The situation is similar between *G2 phase* and *M phase* as indicated by the negative loop between cluster 4 and 5. All of these match the real cell cycle development process (Wittenberg and Reed 2005).



a. Organized literature results. Picture is adapted from (Simon, Barnett et al. 2001)

b. Inferred relations of transcription factor genes

Figure 5-9 Transcription regulation of cell cycle transcription factor genes. The inferred relationships are based on the network shown in Figure 5-7.b

The cell cycle related transcription factors include Mbp1, Swi4, Swi6, Mcm1, Fkh1, Fkh2, Ndd1, Swi5 and Ace2 (Simon, Barnett et al. 2001). Based on their distribution over clusters, we can infer their regulatory relations, as shown in Figure 5-9.b. We found all of 9 TFs are located within the identified cell cycle loops in the network, i.e., 2 → 4 → 5 → 2 and 2 → 4 → 5 → 6 → 2. Comparing with the organized literature results (Simon, Barnett et al. 2001; Lee, Rinaldi et al. 2002), as shown in Figure 5-9.a, the regulatory relationships Mcm1 → Cln3 → Swi4, Mcm1 → Swi4 and Mcm1,Fkh2,Ndd1 → Swi5 are clearly indicated in the inferred networks. One difference is that the inferred network separates Swi4 and Swi6,Mbp1

into cluster 2 and cluster 4, because Swi4 and Swi6,Mbp1 have obvious time delays. As shown in Figure 5-8, cluster 4 also includes genes in *late G1* stage, so the inferred network still matches Figure 5-9.a.

(Simon, Barnett et al. 2001) provide 227 TF binding interactions between 132 cell cycle-related genes and cell cycle-related TFs using genome wide location analysis. Among these genes, 80 genes have only one TF binding. Table 5-6 shows the TF binding evaluation results of the inferred networks. $P_{correct}$ of Figure 5-7.b (after d-separation checks with τ_{error} = 0) is almost the same as $P_{correct}$ of Figure 5-7.a (before d-separation checks). This indicates the d-separation checks keep almost all key interactions. For network Figure 5-7.c (after d-separation checks with τ_{error} = 10 min), $P_{correct}$ decreases, i.e., some interactions were falsely deleted during the d-separation checks. This indicates that including constraints during d-separation checks helps to reduce the chance of false deletion. The results shown in Table 5-6 assume that all TF binding interactions take regulation roles during gene transcription in the cell cycle. However, it is possible that only some of the binding TFs take major regulation roles. Under this assumption, 81.1% genes (107 genes) have one of the binding TFs located either in the same cluster or in their positively correlated parent clusters for networks Figure 5-7.a and b; and 93 genes for network Figure 5-7.c. Other reasons for the unidentified TF interactions include: factors other than the nine TFs listed above may have major regulatory roles; the linear model in equation (5-1) does not capture complex interactions when several TFs regulate one gene in a nonlinear fashion; or the genome wide location analysis itself is noisy and has high positive detection rate.

Table 5-6 Network evaluation by TF binding information

Network	Figure 5-7.a	Figure 5-7.b	Figure 5-7.c
I_{same}	42	42	42
I_{parent}	112	108	80
$P_{correct}$	67.8%	66.1%	53.7%

I_{same} represents the number of TF binding interactions whose TF and regulated gene(s) are in the same cluster; I_{parent} represents the number of TF binding interactions whose TF locates in the direct positive parent cluster of the regulated gene(s); $P_{correct}$ represents the percentage of TF binding interactions reflected in the network, i.e. $P_{correct} = (I_{same} + I_{parent}) / I_{total}$, where I_{total} = 227 is the total number of TF binding interactions.

The work of (Zou and Conzen 2005) used a DBN (Dynamic Bayesian Network) on the same data set. The DBN correctly identified 17 interactions without using prior knowledge and 46 interactions by using prior knowledge of TFs. Comparing with the results shown in Table 5-6, the results of CBTC algorithm are much better. However, their inferred network is based on the individual genes instead of clustering first and then doing network inference based on the clustering center profiles.

Finally, we evaluated the network with the biological process information came from the paper (Simon, Barnett et al. 2001). Figure 5-10 shows the frequency distribution of genes in different biological processes within the same cluster. Based on the network topology, time delay information and the gene development stage information of the clusters, we can clearly see the biological processes change over different clusters. One interesting finding is that only cluster 1 has no cell cycle control related genes. If we check Figure 5-7.b or c, we found only cluster 1 has no edges coming out from it. This indicates the result network does capture this relation. Detailed explanations of other biological process changes are beyond the purpose of this paper.

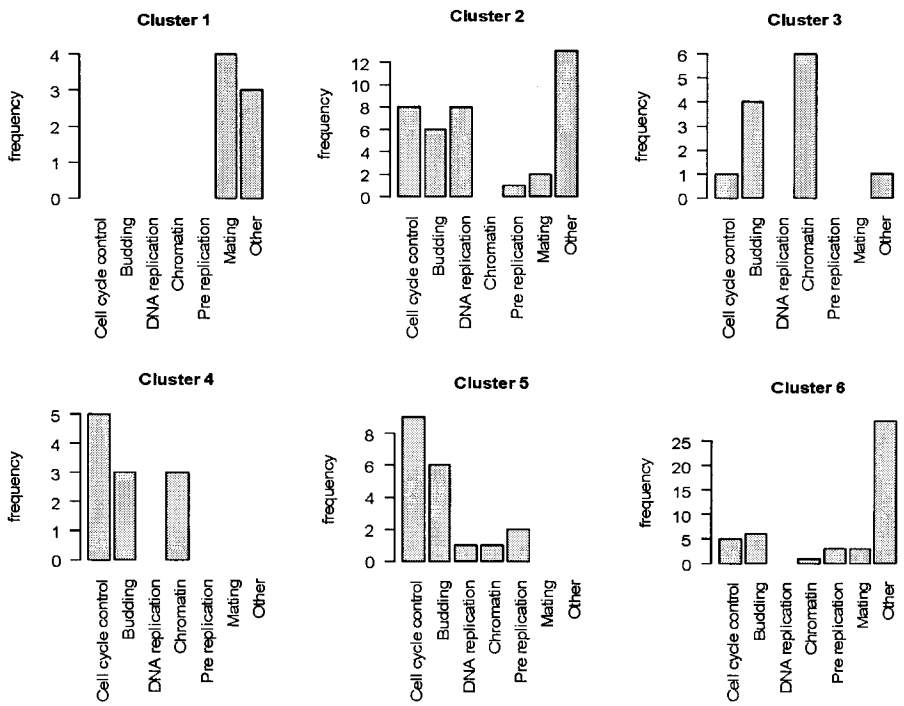


Figure 5-10 Frequency distribution of the genes in different biological processes in the same cluster

Therefore, the algorithm successfully identified the yeast cell cycle development stages, cell cycle loop, negative feedback loops, the relationships between cell cycle related TFs and their regulated genes. The network greatly helps us to understand how gene activities changing over development stages. It also provided hypothesis for further biological studies.

5.4 Discussion

The results of simulated and yeast cell cycle data show that the CBTC algorithm is effective for estimating networks with feedback loops as well as avoiding the problems encountered by the GGM algorithm when strong feedback loops are present. The CBTC algorithm is straightforward (without iterations) and exhaustive (checks every significant edge in the network). The computation load is linear with the number of the significant edges in the network. Comparatively, dynamic Bayesian networks algorithms are computationally expensive. The CBTC algorithm is based on continuous values, can easily deal with the interactions with different time delays and have no limitation of input or output edge number of the node.

In the CBTC algorithm, the d-separation check is based on local network topology and typically uses low order partial correlation. Therefore, the algorithm also works well for dealing with large scale networks and the data with short profile lengths. In practice, increasing the correlation threshold can increase the algorithm specificity and help identify the major interactions.

The d-separation check may also lead to the false deletion of correct edges. However, constraints, like time delay and sign of the path, on the conditional variables during d-separation check can reduce that problem. Other prior knowledge of the genes or clusters can also be added. The processing order of the sequential d-separation check affects results. A two-step correlation threshold check was used to minimize the effect. Multiple iterations with gradually increasing correlation thresholds may help to get more uniform results.

The estimation of the time correlation matrix and time delay information is a critical step. Time delay estimation becomes unreliable when there are interactions among more than two variables or the correlation profile is multimodal. Also, the time correlation was based on the overall profiles, but the real regulation may happen only in a specific time period. In this case,

the algorithm may miss the interactions. In future work, we will try to capture this kind of interaction, but it requires more sample points with shorter sample intervals.

Our model is based on a linear (or approximately linear) assumption. This approximation works well in many cases, as shown in the results of yeast cell cycle data, but it may not hold in some complex cases. One potential solution is to use the CBTC algorithm as a preprocessing step, then use detailed models to infer the detail network parameters.

5.5 Conclusion

The CBTC algorithm shows good performance using both simulated and real yeast cell cycle data. Also it provides the time delay and edge direction information and finds feedback loops. Correlation and time delay information is not enough to determine the real regulatory relationships, especially when the sample rate, signal length and resolution of microarray technology are limited. Combination with other biological prior knowledge and information, such as genome sequence information, will be necessary to get more detailed relationships and hypotheses.

CHAPTER 6. GENETIC NETWORK INFERENCE WITH SHORT-TIME CORRELATION

6.1 Introduction

In Chapter 5, we described genetic network inference based on pair wise time correlation. Time correlation is based on the entire expression profiles and tries to catch the linear relationships between entire expression profiles. However, the real gene interactions usually happen within specific time periods and conditions. Correlation based on entire profiles could miss some short-duration interactions. In this chapter, we propose the use of short-time correlation to catch the transient interactions. The idea of short-time correlation is derived from the STFT (Short-Time Fourier Transform), which is widely used to catch frequency information in a short period of time and represent frequency changes over time. In STFT, a slide window function is multiplied with the time series signal, and Fourier transform is done only over the segment of the profile under the nonzero window function. By sliding the window function over the entire time period, we can observe how the frequency distribution changes over time. Based on the same idea, short-time correlation can also observe how the correlation changes over time and calculates the correlation at specific time periods. Another advantage of using short-time correlation is that we can visualize the distribution of correlation coefficient to see how correlation changes over time, time delay and the size of window function. These graphs will help us understand and discover useful features of interactions.

The genetic networks can be constructed based on the short-time correlation matrix at each time interval. Comparing the networks of adjacent time intervals, we can see how network parameters change over time. The d-separation check can be adapted to differentiate the direct and indirect interactions. Because the networks of adjacent time intervals are closely related, the d-separation check in short-time correlation case will combine the network information at adjacent time frames. We use the same yeast cell cycle data set as in

Chapter 5. The results show how network parameters change over time. New significant interactions were identified and match the biological interpretations.

6.2 Methodology

6.2.1 Short-time correlation coefficient

In short-time correlation, a slide window function is multiplied with the expression profiles. Time correlation is computed over the profiles under the nonzero window function. Therefore, the definition of short-time correlation coefficient is similar with time correlation. The short-time correlation coefficient $r_{ijmM}(\tau)$ can be defined as:

$$r_{ijmM}(\tau) = r_{ijmM}(l\Delta t) = \mathrm{cov}(\mathbf{x}'_{imM}, \mathbf{x}'_{jmM}) / \sqrt{\mathrm{var}(\mathbf{x}'_{imM})\,\mathrm{var}(\mathbf{x}'_{jmM})} \tag{6-1}$$

$$\mathbf{x}'_{imM}[k] = w_M(k-m)\mathbf{x}_i[k],\ \mathbf{x}'_{jmM}[k] = w_M(k+l-m)\mathbf{x}_j[k+l],$$
$$k = 1,\ldots,N,\ m = 1,\ldots,N,\ 1 \le k+l \le N$$

$$w_M(k) = \begin{cases} 1, & -[M/2] \le k \le [M/2] \\ 0, & otherwise \end{cases}$$

where $r_{ijmM}(\tau)$ represents short time correlation coefficient between expression profiles $\mathbf{x}_i$ and $\mathbf{x}_j$ with time delay τ, window size M at time frame m. m is the time index where the center of window function $w_M(k)$ is located, M is the size of the window function, τ is the time shift between gene profiles $\mathbf{x}_i$ and $\mathbf{x}_j$, $\tau = l\Delta t$, Δt is the sample interval, l is the number of sample intervals shifted between two profiles, N is the profile length. For periodic time profiles, circular time correlation is effective, i.e., the time points at the end of the time series are rewound to the beginning of the series after time shifting.

The estimation of time delay τ and edge direction during time interval m is exactly the same as time correlation. Since the window size is changeable, in order to make the time correlation with different window size comparable, we translate all the short-time correlation coefficients $r_{ijmM}(\tau)$ as p-values. The p-value of the correlation can be computed by equation (5-4).

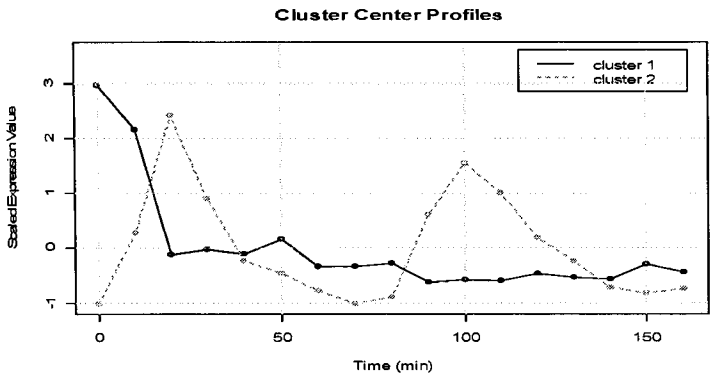
6.2.2 Visualizing the short-time correlation coefficient distribution

By visualizing the short-time correlation coefficient distribution, we can easily observe how short-time correlation coefficients change over different parameters. For short-time correlation coefficients $r_{ijmM}(\tau)$ between expression profiles $\mathbf{x}_i$ and $\mathbf{x}_j$, there are three parameters: time frame m, window size M, and time delay τ. By fixing one parameter, we can visualize the short-time correlation coefficients over another two parameters in 2-D graph. If we fix window size M, we can obtain graphs of time delay τ v.s. time m; if we fix time delay τ, we can obtain graphs of window size M v.s. time m; by using estimated time delay τ', $r_{ijmM}(\tau')$ will produce a graph of window size M v.s. time m. Next we will give examples for all these cases.

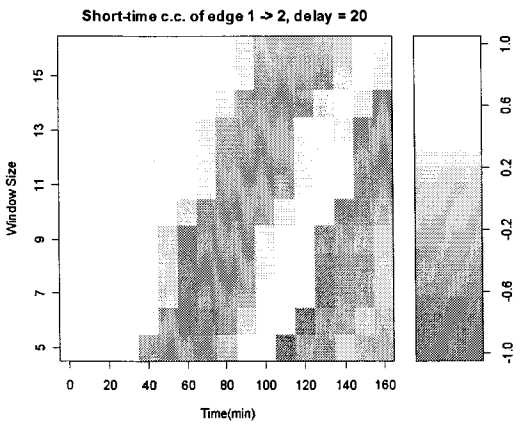
Figure 6-1 shows an example of visualizing short-time correlation coefficients. Figure 6-1.a shows two example expression profiles, which are cluster center profiles of cluster 1 and 2 in Figure 5-6. Figure 6-1.b shows the short-time correlation coefficients distribution for window size v.s. time with fixed delay $\tau = 20$ min. From Figure 6-1.b, we can easily observe that the expression profiles of cluster 1 and 2 have the most significant correlation at m =10 min with window size $M = 6$; the correlation decreases gradually as the window size increases. We can also identify two peaks over the time frame m, which correspond to two period of yeast cell cycle. Figure 6-1.c shows the short-time correlation coefficients distribution for time m v.s. time delay τ with fixed window size $M = 6$. We can see that expression profiles of cluster 1 and 2 have significant correlation with time delay $\tau = 20$ min, and significant negative correlation with time delay $\tau = 0$ min, which match the expression profiles shown in Figure 6-1.a. Based on Figure 6-1.c, we can estimate the time delay τ' at each time frame and window size, and produce a graph of $r_{ijmM}(\tau')$ for time m v.s. window size M, as shown in Figure 6-1.d. Figure 6-1.e shows the estimated time delay τ' at each time m and window size M. From Figure 6-1.d, we can easily identify at which window size M and time m the correlation is significant and we can observe its dynamic behavior.

6.2.3 Differentiate direct and indirect interactions

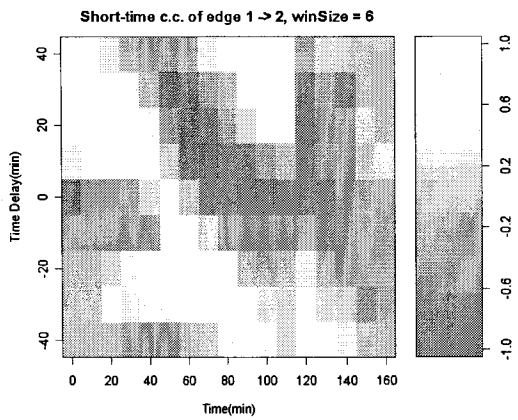
In Chapter 5, we described the algorithm of differentiating direct and indirect interactions



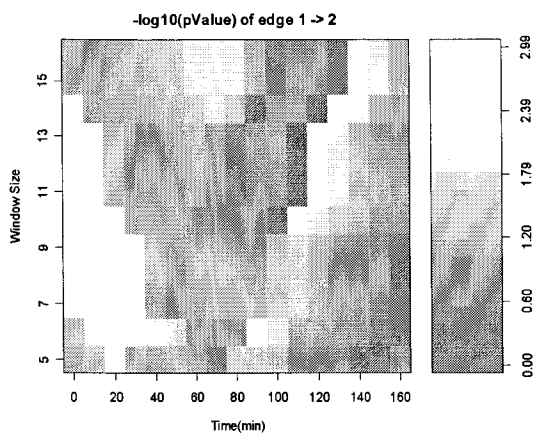
a. Example expression profiles for short-time correlation coefficient estimation



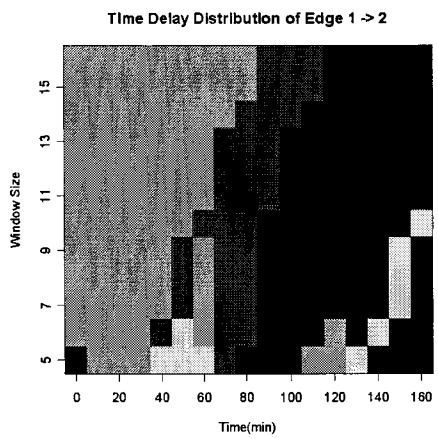
b. Window Size v.s. Time with fixed Delay



c. Delay v.s. Time with fixed Window Size



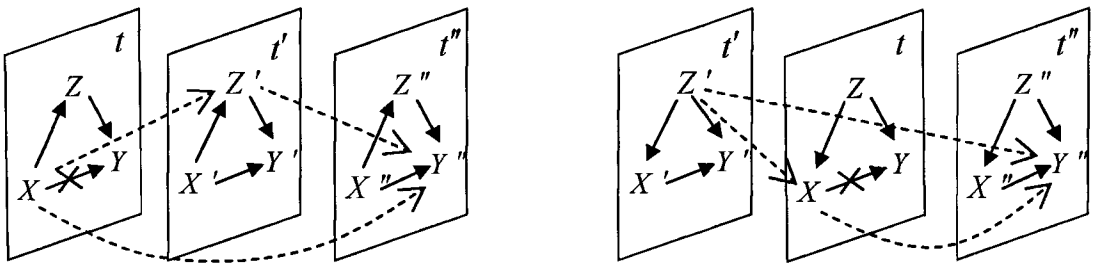
d. Log(pValue), Window Size v.s. Time



e. Estimated Delay, Window Size v.s. Time

Figure 6-1 Visualizing short-time correlation coefficients

by using d-separation and partial correlation theories. The same algorithm can also be applied for the short-time correlation based algorithm. Because short-time correlation based network inference will result in a series of networks at different time frames and the networks at adjacent time frames are closely related, the two types of d-separation cases, as shown in Figure 5-1, are adapted to best utilize this information.



(a) Intermediate cause (b) Common cause

Figure 6-2 d-separation check involving different time frames. t, t', t'' **represent the corresponding time frame of the network,** X, X', X'' **are the windowed expression profiles at time frame** t, t', t''.

For the intermediate d-separation case shown in Figure 6-2.a, if we only consider the time frame t and check whether edge XY is d-separated, the partial correlation $r_{XY|Z}$ is:

$$r_{XY|Z} = (r_{XY} - r_{XZ}r_{ZY}) / \sqrt{(1-r_{XZ}^2)(1-r_{ZY}^2)} \tag{6-2}$$

and check whether $r_{XY|Z}$ is close to zero. If we consider the information at adjacent time frames, we need to modify equation (6-2). Suppose the time delay from X to Y, $d_{XY} = t'' - t$, and the time delay from X to Z, $d_{XZ} = t' - t$, so the actual positions of Y and Z are in time frame t'' and t' respectively. The real partial correlation should represent the correlation of between X and Y'' conditioned on Z', which corresponds to the edges shown in dotted lines in Figure 6-2.a. As a result, partial correlation $r_{XY|Z}$ should be calculated as

$$r_{XY|Z} = (r_{XY} - r_{XZ}r_{Z'Y'}) / \sqrt{(1-r_{XZ}^2)(1-r_{Z'Y'}^2)} \tag{6-3}$$

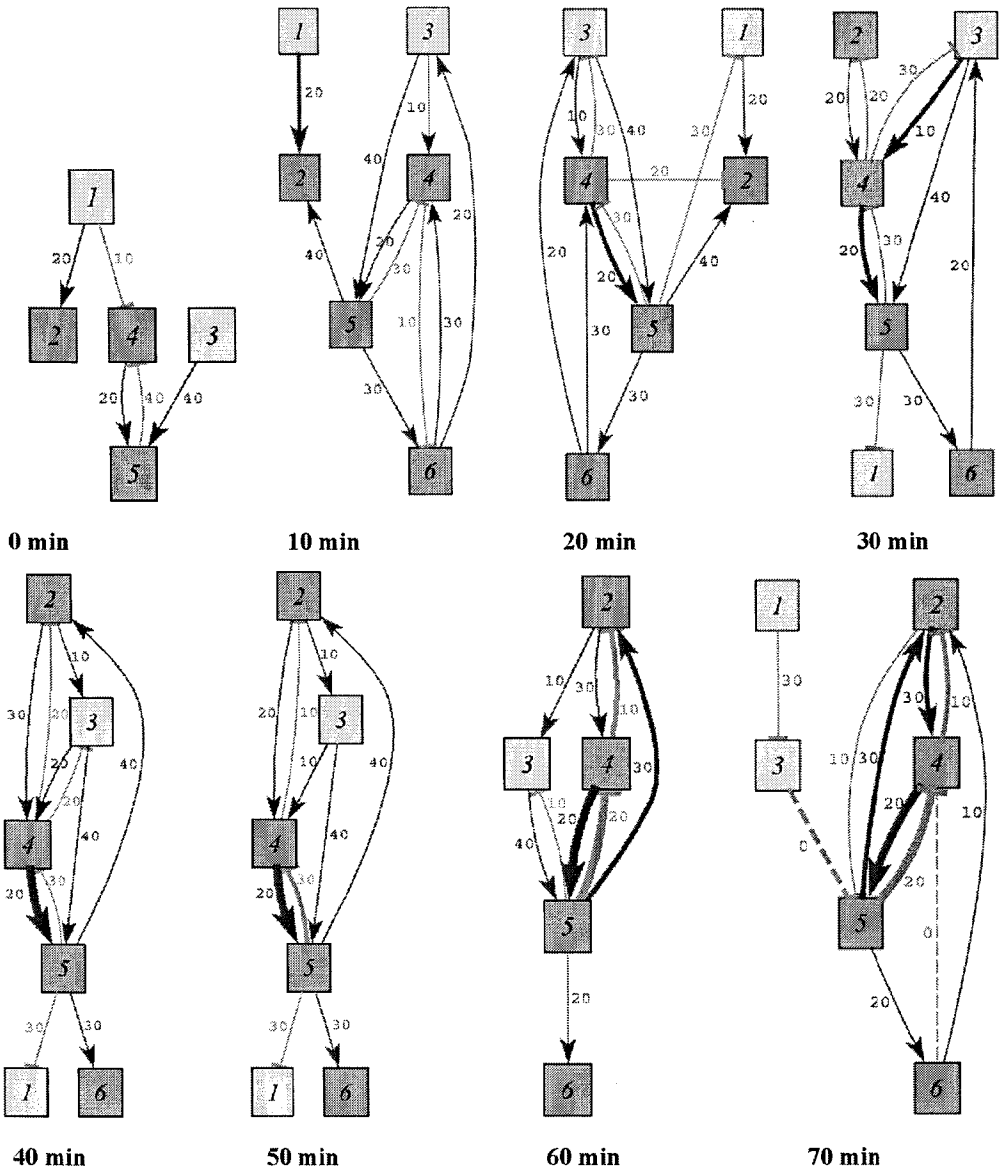
Similarly, for the d-separation case shown in Figure 6-2.b, partial correlation coefficient $r_{XY|Z}$ should be calculated as

$$r_{XY|Z} = (r_{XY} - r_{Z'X'}r_{Z'Y'}) / \sqrt{(1-r_{Z'X'}^2)(1-r_{Z'Y'}^2)} \tag{6-4}$$

Time delay constraints can also be imposed to reduce the false deletion rate of edge XY. For Figure 6-2.a, $|d_{XY}-(d_{XZ}+d_{Z'Y'})|\leq \tau_{error}$. For Figure 6-2.b, $|d_{XY}-(d_{Z'X'}-d_{Z'Y'})|\leq \tau_{error}$. Other d-separation procedures are the same as described in Chapter 5.

6.3 Results

We used the same data set as Chapter 5. The network inference is based on 6 cluster center profiles as shown in Figure 5-6.



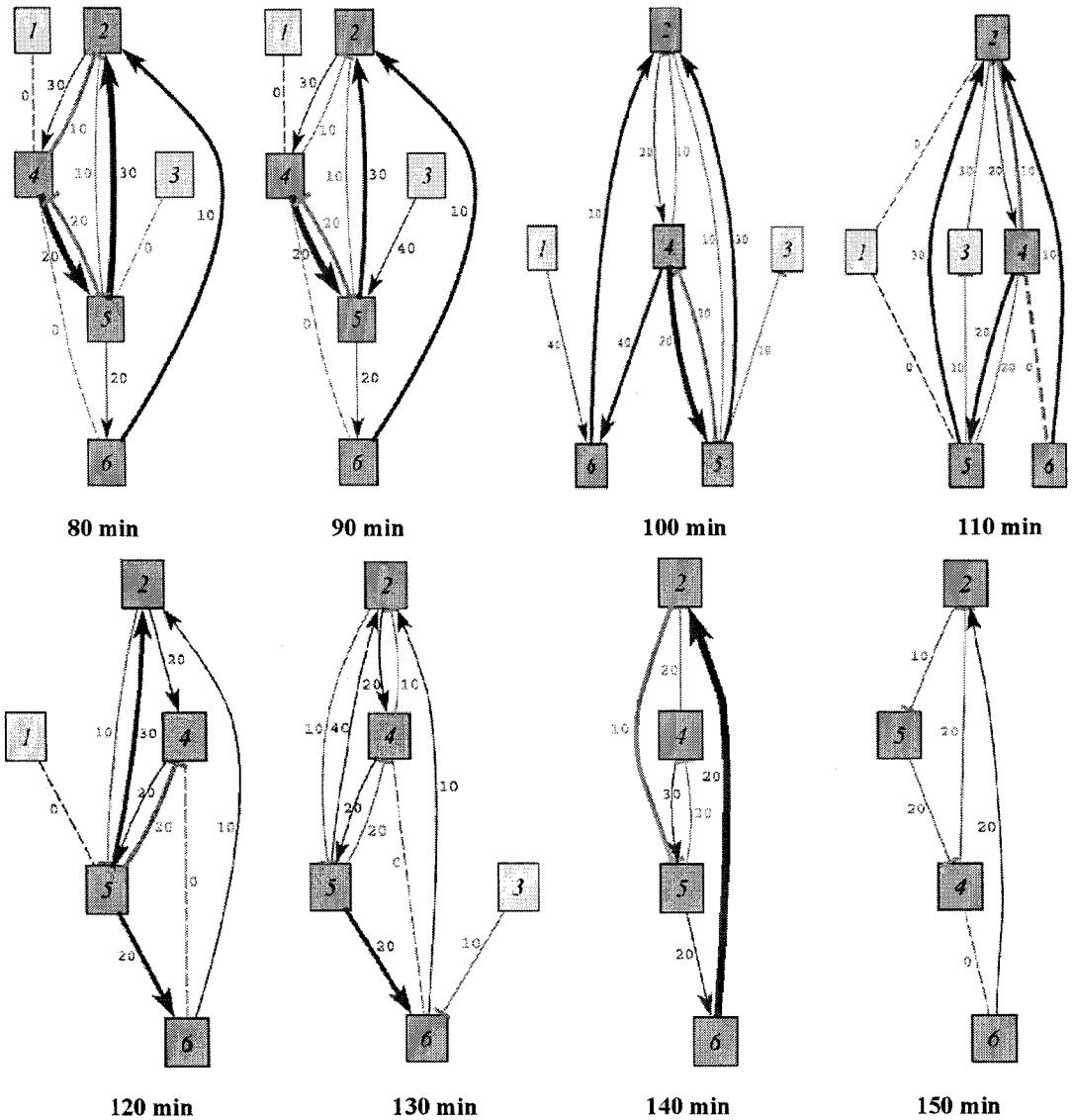


Figure 6-3 Networks at each time frame. Graph annotation is the same as Figure 5-7. Network inference setting: tError = 0, pValueT = 0.01

6.3.1 Network inference results with fixed window size

Figure 6-3 shows the networks at each time frame with the window size $M = 10$. We can see the network topology and edge significance gradually change over time.

Edge 1 → 2 exists only within the networks at time frame 0, 10 and 20 min, and has most significant correlation at time frame 10 min with 20 min time delay. By checking the GO Biological Process of the genes in cluster 1, we find the most significant biological process of cluster 1 is "response to pheromone during conjugation with cellular fusion"; for cluster 2, it is "cell cycle" and "axial budding". So the reasonable biological explanation is that the genes in cluster 1 respond to pheromone and activate the genes participating in cell cycle processes in cluster 2 at the very beginning of the cell cycle profiles. The interaction 6 → 2 exists at the end of first cell cycle and through the second cycle. It has most significant correlation at 140min time frame. Since most genes in cluster 6 are in the early G1 stage, and genes in cluster 2 are in late G1 state, the correlation significance change of interaction 6 → 2 matches the cell cycle stage information. One interesting finding is that interaction 2 → 5 always exists together with 6 → 2. This means whenever the genes in cluster 6 (early G1 phase) activates the genes in cluster 2 (late G1 phase), the genes in cluster 2 will depresses the genes in cluster 5 (M phase) in reverse. Another finding is that the interactions between clusters 4 and 5 are consistent. They almost always exist throughout two cell cycles. And the cycle 2 → 4 → 5 exists at most of the time frames. We also find some interesting phenomena, for example, cluster 3 participates in the cycle 3 → 4 → 5 → 6 → 3 during time frames 10, 20, 30min. Meanwhile the interaction between 2 → 4 is weak. When the interaction between 2 → 4 becomes stronger, interaction 3 → 4 becomes weaker. It seems 3 → 4 is replaced by 2 → 4. The biological explanation is still under exploration.

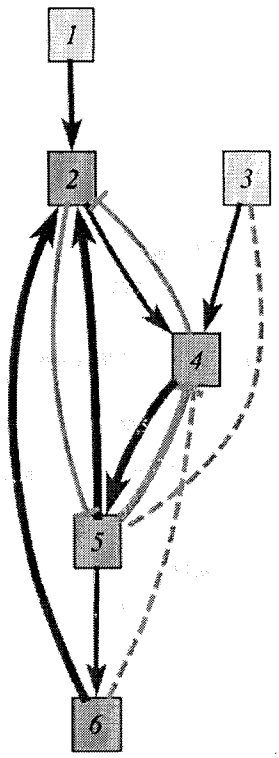


Figure 6-4 Combined networks of Figure 6-3. Only show the edges with p-values less than 0.001. The edge labels represent time delay and most significant time frame, unit is in minute.

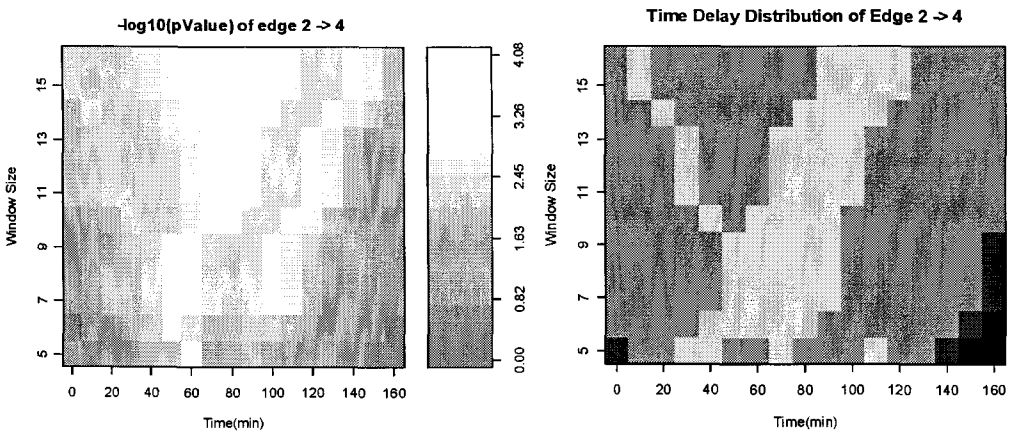
The networks shown in Figure 6-3 provide much detailed information regarding network topology change. Because there is so much information, it is not easy to capture the major interactions. Figure 6-4 shows the

combined network which only keeps the most significant edges over all time frames with p-values less than 0.001. Each edge is labeled with the time delay and the most significant time frame. So we can easily observe the overall network information. If we want to know more details of interaction between two nodes, we can visualize the short-time correlation coefficients under different settings, as shown in Figure 6-5. Comparing with the networks in Figure 5-7, we can see short-time correlation based networks, as shown in Figure 6-5, capture additional interactions: 1 → 2, 2 → 5, 3 → 5 and 4 → 6. And the significance of the edges, especially edge 6 → 2, increases greatly.

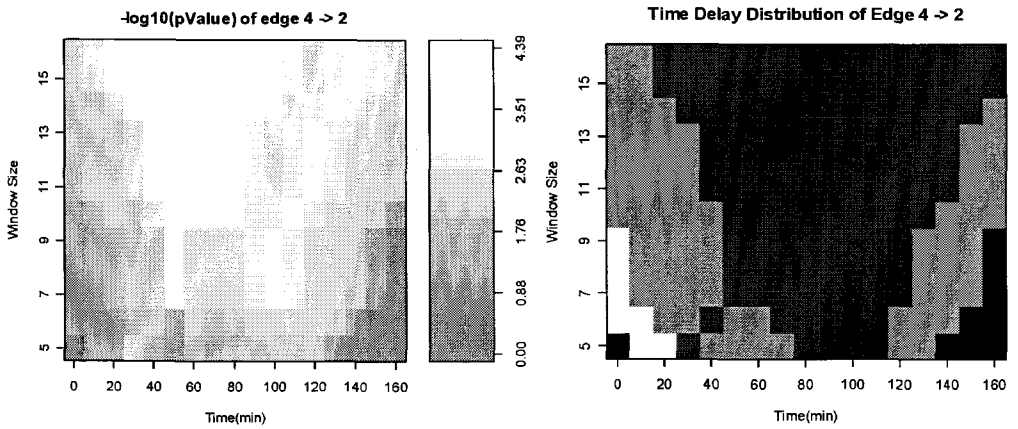
6.3.2 Network inference by visualizing interactions

Figure 6-3 and Figure 6-4 show the networks produced with fixed window size ($M = 10$). But in practice, we do not know what window size should be used in advance. One way is to visualize short-time correlation coefficients $r_{ijmM}(\tau)$ in a graph of window size M vs. time frame m with the time delay τ set as estimated time delay τ', as the example shown in Figure 6-1.d. Figure 6-5 shows such graphs of the significant edges shown in Figure 6-4. In order to make correlation coefficients under different window size comparable, we translate all correlation coefficients as corresponding p-values based on equation (5-4), then visualize the negative logarithm of the p-values in Figure 6-5. Based on Figure 6-5, we can easily identify the window size, time delays and time frames corresponding to the most significant correlation. We can see for most edges, there are significant correlations when window size M is 10. This gives us the idea that a window size of 10 is a good choice.

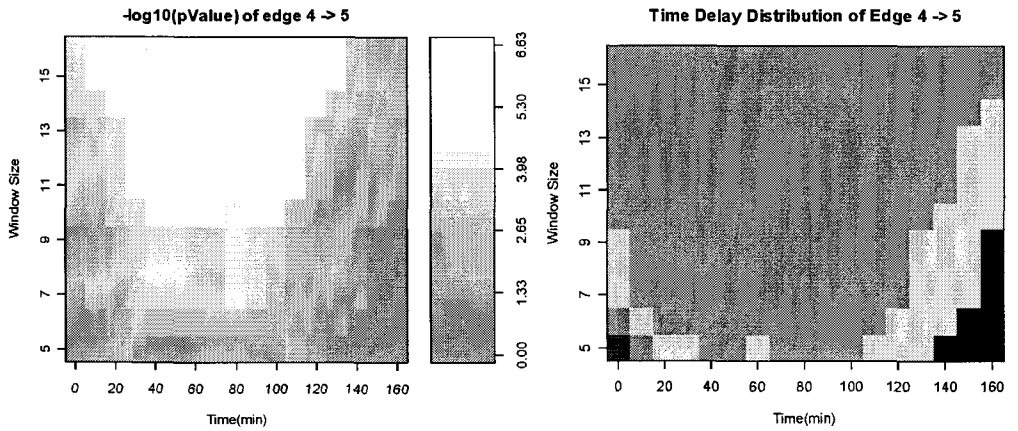
In Figure 6-5, we can also observe both the p-value and time delay distributions change gradually over time and window size. This is reasonable because the slide windows are overlapped between adjacent time frames. If there are big changes between adjacent estimations, it is very possible that some of the estimations may be unreliable over the border area. Another benefit of visualization is that we can identify possible misidentifications or candidate solutions. For example, in Figure 6-5.H, the most significant settings are delay τ = 0 min, window size $M = 10$, time frame $m = 70$ min and minimum p-value $p_{\min} = 4.79 \times 10^{-4}$. But if we check the graph, we find another setting maybe is better: delay $\tau = 40$ min, window size $M = 11$, time $m = 60$ min and $p = 4.92 \times 10^{-4}$, because under this setting, the



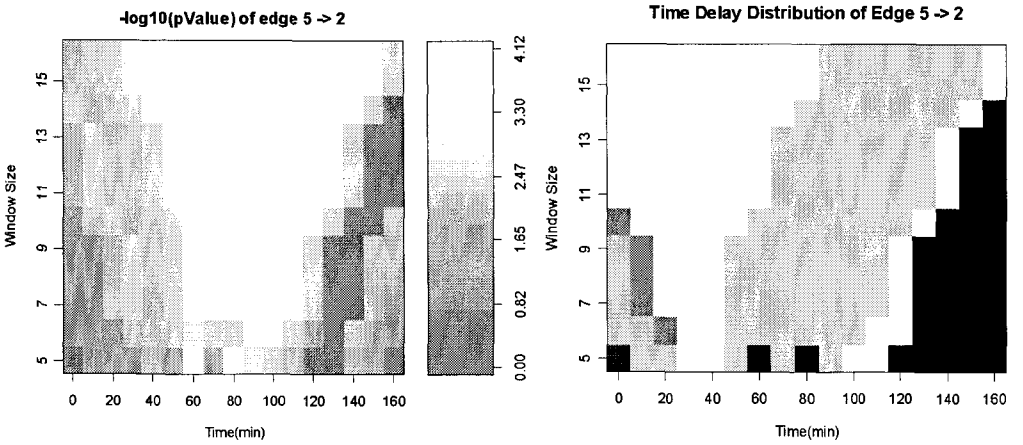
A. Edge 2 → 4, $p_{\min} = 8.40 \times 10^{-5}$, delay = 30min, time = 80min, window size = 11



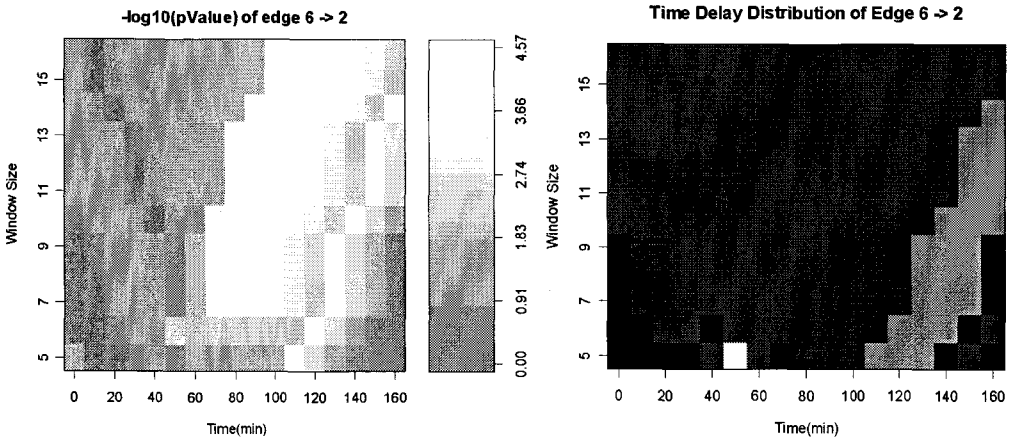
B. Edge 4 → 2, $p_{\min} = 4.08 \times 10^{-5}$, delay =10min, time = 70min, window size = 11



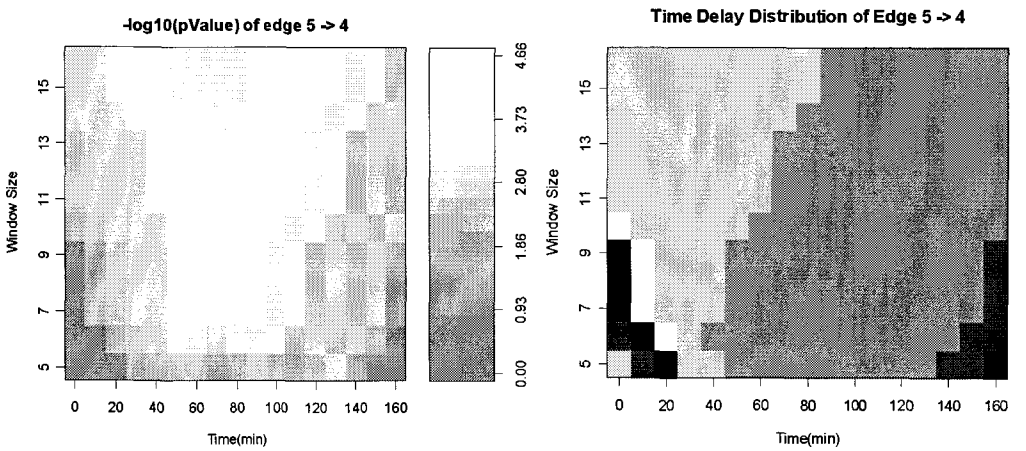
C. Edge 4 → 5, $p_{\min} = 2.35 \times 10^{-7}$, delay = 20min, time = 70min, window size = 14



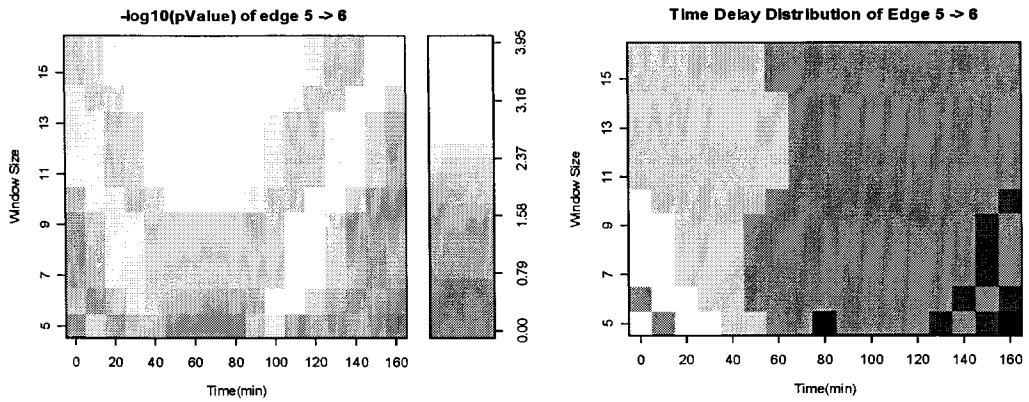
D. Edge 5 → 2, $p_{\min} = 7.59 \times 10^{-5}$, delay = 30min, time = 80min, window size = 10



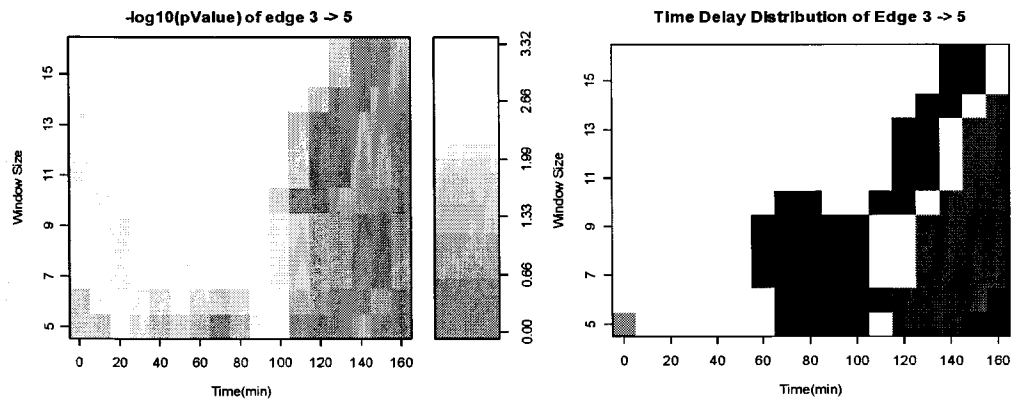
E. Edge 6 → 2, $p_{\min} = 2.69 \times 10^{-5}$, delay = 20min, time = 120min, window size = 6



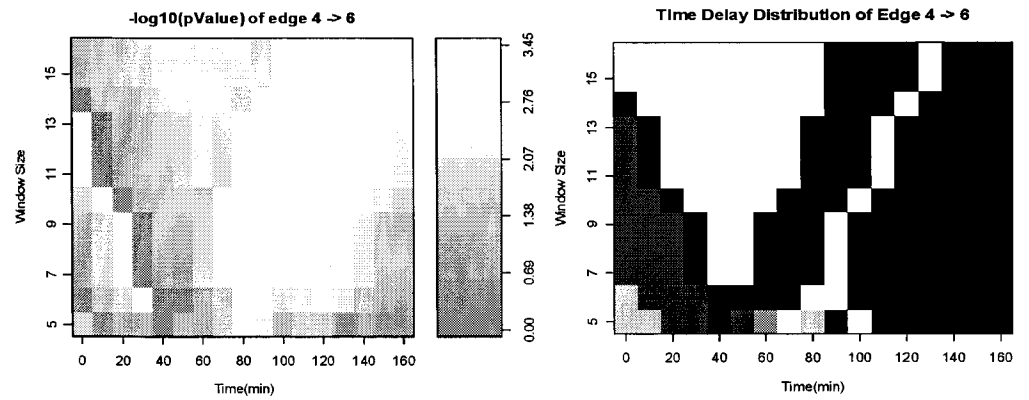
F. Edge 5 → 4, $p_{\min} = 2.19 \times 10^{-5}$, delay = 20min, time = 80min, window size = 14



G. Edge 5 → 6, $p_{\min} = 1.13 \times 10^{-4}$, delay = 20min, time = 110min, window size = 6



H. Edge 3 → 5, $p_{\min} = 4.79 \times 10^{-4}$, delay = 0min, time = 70min, window size = 10



I. Edge 4 → 6, $p_{\min} = 3.52 \times 10^{-4}$, delay = 0min, time = 70min, window size = 14

Figure 6-5 Correlation significance and time delay distribution over window size v.s. time. Left column corresponds to the distribution of -Logarithm of p-values, right column is the estimated time delay used to estimate correlation coefficients and their p-values. White color, light grey, grey, dark grey and black respectively represent time delays 40, 30, 20, 10 and 0min.

estimated time delay (40min) is the same as adjacent time estimated time frames, while under the optimal setting, the corresponding time delay (0min) is different from adjacent time frames (40min). A similar situation applies to edge 4 → 6. Based on the p-value and time delay distribution shown in Figure 6-5.I, we can find the optimal setting is located in the border area, and the estimation may be unreliable. The alternative estimation with time delay 40min is very possibly caused indirectly by 4 → 5 → 6.

Table 6-1 Parameter settings of the most significant edges with p-values ≤ 0.001

Edge	p-value	τ (min)	m (min)	M
1 → 2	1.02×10^{-3}	20	30	6
1 — 3	5.49×10^{-4}	0	10	5
2 → 4	8.40×10^{-5}	30	80	11
2 —| 5	1.15×10^{-4}	10	90	14
3 → 4	6.30×10^{-4}	10	40	7
3 — 5	4.79×10^{-4}	0	70	10
4 —| 2	4.08×10^{-5}	10	70	11
4 → 5	2.35×10^{-7}	20	70	14
4 — 6	3.52×10^{-4}	0	70	14
5 → 2	7.59×10^{-5}	30	80	10
5 —| 4	2.19×10^{-5}	20	80	14
5 → 6	1.13×10^{-4}	20	110	6
6 → 2	2.69×10^{-5}	20	120	6

"—|","→" and "—" represents negative regulation, positive regulation and negative coregulation respectively.

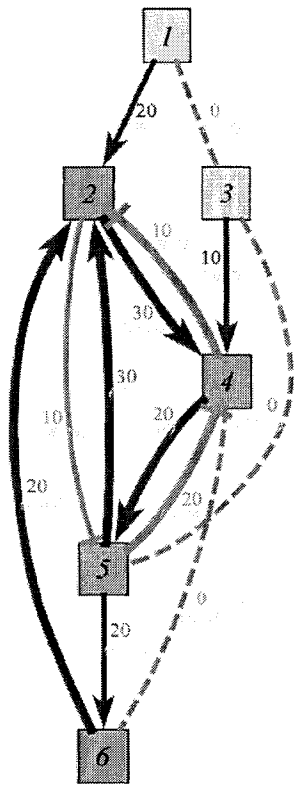


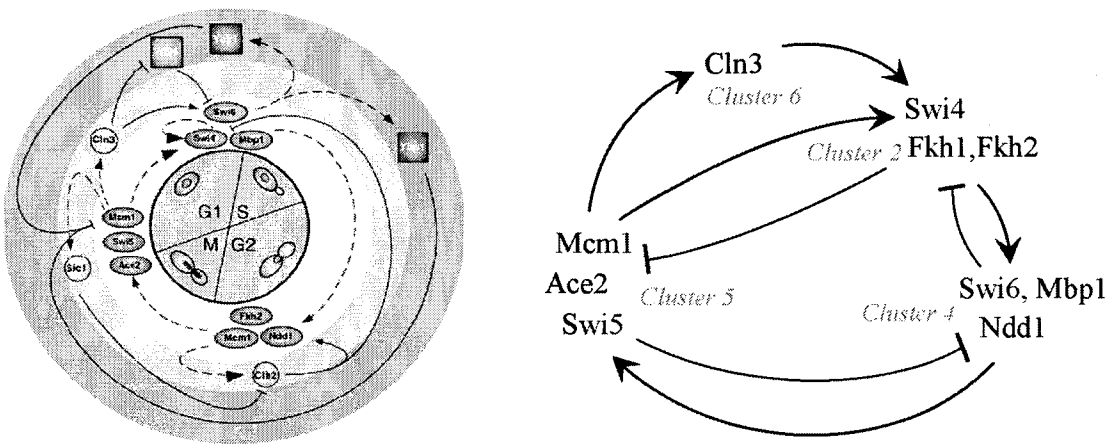
Figure 6-6 Combined network with the most significant edges over all window size, time and time delay. Only show the edges with p-values less than 0.001. Edge labels at the top represent time delay. Edge labels at the bottom represent the most significant time frame and window size.

Table 6-1 shows the parameter settings of the most significant edges with p-value ≤ 0.001. By combining these edges together, we obtain a network as shown in Figure 6-6. Comparing Figure 6-5 and Figure 6-6, we find they are almost the same except for an additional link between cluster 1 and 3 and increased edge significance. This means window size $M = 10$ is a good choice.

6.3.3 Comparison with literature results

Figure 6-7 shows the integrated transcription regulatory networks during the cell cycle.

Figure 6-7.a is the organized literature results from (Wittenberg and Reed 2005). Figure 6-7.b is the inferred network based on the distribution of the cell cycle related transcription factors (Mbp1, Swi4, Swi6, Mcm1, Fkh1, Fkh2, Ndd1, Swi5 and Ace2) over the cluster network shown in Figure 6-4 or Figure 6-6. We find all of 9 TFs are located within the identified cell cycle loops in the network, i.e., 2 → 4 → 5 → 2 and 2 → 4 → 5 → 6 → 2. Compared with the organized literature results (Wittenberg and Reed 2005), as shown in Figure 6-7.a, the regulatory relationships Mcm1 → Cln3 → Swi4, Mcm1 → Swi4 and Mcm1,Fkh2,Ndd1 → Swi5 are clearly indicated in the inferred networks. Three negative feedbacks 2 → 4 —| 2, 4 → 5 —| 4, 5 → 2 —| 5 are identified, which also matches Figure 6-7.a. One difference is that the inferred network separates Swi4 and Swi6,Mbp1 into cluster 2 and cluster 4, because Swi4 and Swi6,Mbp1 have obvious time delays. As shown in Figure 5-8, cluster 4 also includes genes in *late G1* stage, so the inferred network still matches Figure 6-7.a.



a. Organized literature results **b. Inferred relations of transcription factor genes**

Figure 6-7 Integration of transcriptional regulatory networks during the cell cycle. The inferred relationships are based on the network shown in Figure 6-4 and Figure 6-6. Picture shown in Figure 6-7.a is adapted from (Wittenberg and Reed 2005).

6.4 Discussion

In this chapter, we proposed short-time correlation as a method to identify the transient interactions. The results show it successfully identifies some links which cannot be identified by time correlation. Biological information indicates these additional edges are reasonable.

Also the edge significance increases greatly by considering short-time correlation. By creating networks at individual time frames, we can observe the network topology changing over time. Visualization of the short-time correlation coefficients helps us perform in-depth analysis on the behavior of the correlation over time, window size and time delay, and provides some alternative parameter settings which probably are more biologically reasonable.

By using short time correlation, it is possible to capture some nonlinear interactions which cannot be captured by using time correlation. This is similar to approximating a nonlinear function by pieces of linear segments. However, because of the limited profile length and sample interval, more detailed information cannot be shown by short-time correlation. As the microarray chips become cheaper, hopefully sample intervals will become shorter and profile length will be much longer. Then short-time correlation could provide more dynamic information and achieve better performance. Also, more useful signal processing algorithms could be applied in the analysis of time series expression profiles.

CHAPTER 7. GENETIC NETWORK INFERENCE WITH MULTI-SCALE RESOLUTION

7.1 Introduction

Gene expression data are noisy, large scale and with groups of gene expression profiles coregulated. The behavior of biological systems is inherently fuzzy. The same gene may participate in different biological processes at different times and conditions with different expression levels. In Chapter 4, we proposed the Multi-scale Fuzzy K-means clustering algorithm to group genes with similar patterns, and then infer the genetic networks based on the cluster centers. Multi-scale Fuzzy K-means clustering can derive clusters with different degrees of similarities. Based on these cluster center profiles, we can further create networks at different levels of detail. In this chapter, we will cover this topic in more depth and try to uncover the real coregulated genes by integrating regulatory sequence analysis.

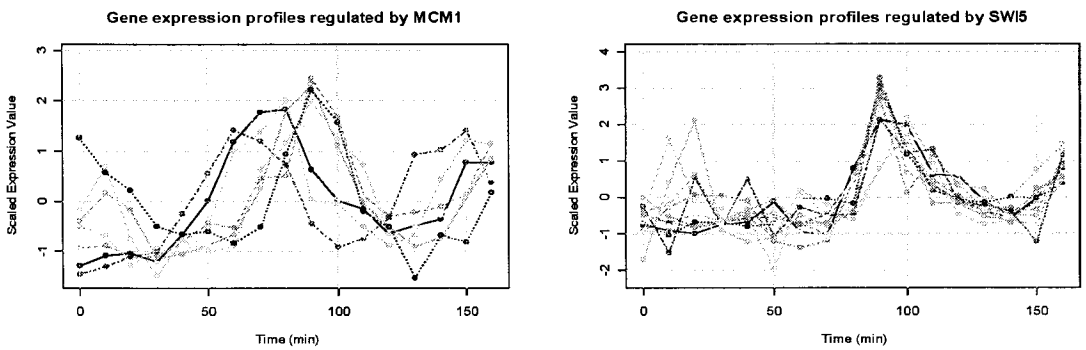


Figure 7-1 Expression profiles of coregulated genes

Coregulated genes usually have similar gene expression patterns. Clustering is widely used to group genes with similar patterns, which are supposed to be coregulated. But because of the complexity of regulation, the profile similarity among coregulated genes can vary widely. Figure 7-1 shows two groups of coregulated genes regulated by transcription factor Mcm1 and Swi5. We can see the similarities among the coregulated gene expression profiles have wide variation. The profiles of coregulated genes regulated by Swi5 have very similar patterns, however, for the genes regulated by Mcm1, their patterns spread out and have

different time delays. Common clustering algorithms are not effective at identifying these kinds of coregulated genes that have quite different group similarities. Multi-scale Fuzzy K-means clustering algorithm can identified them by using different window scales.

In this chapter, we adopt Multi-scale Fuzzy K-means clustering algorithm to perform clustering at different window scales, which correspond to different detailed levels. Then we carry out regulatory sequence analysis over all clusters at different levels. Significant motifs can be identified. Among them, we select the most significant and record the corresponding clustering window scale, which characterizes the degree of coregulation among the genes. Genetic networks are created based on the cluster center profiles at different levels. The networks can be refined by combining the results with regulatory sequence analysis. Gene ontology information is then used to annotate these groups of coregulated genes.

7.2 Methodology

7.2.1 Multi-scale Fuzzy K-means clustering algorithm

The details of Multi-scale Fuzzy K-means clustering algorithm are described in Chapter 4. Please refer to Section 4.3 for details.

7.2.2 Methods to identify regulatory sequence motifs

There are several methods to search over-represented motifs at the sequence upstream of coregulated genes (Tompa, Li et al. 2005). These algorithms can roughly be categorized into two classes: word frequency based (van Helden, Andre et al. 1998; Jensen and Knudsen 2000; van Helden, Andre et al. 2000; van Helden, Rios et al. 2000; van Helden 2003; van Helden 2004) and probabilistic sequence models based (Lawrence, Altschul et al. 1993; Bailey and Elkan 1994; Roth, Hughes et al. 1998; Hughes, Estep et al. 2000; Thijs, Lescot et al. 2001; Marchal, Thijs et al. 2003).

The word frequency based methods are based on the frequency analysis of oligonucleotides in the upstream regions of coregulated genes. The statistical significance of a site is calculated based on oligonucleotide frequency tables observed in all non-coding regions of the specific organism's genome. Usually, the length of oligonucleotide is varied

from 4 to 9. Hexanucleotide (with oligonucleotide length equal to 6) analysis is most widely used. The identified significant oligonucleotides can be grouped as longer consensus motifs. The strengths of word counting based methods include its simplicity and efficiency; also, it is rigorous (compared with heuristic methods) and exhaustive (all over-represented patterns of chosen length are detected). The cost of the simplicity is that it is limited to the detection of short and relatively conserved motifs and is not effective at identifying complex motif patterns.

For the probabilistic based methods, the motif is represented as a position probability matrix, Position Specific Scoring Matrix (PSSM), and the motifs are assumed to be hidden in the noisy background sequences. Maximum likelihood estimation is used to estimate model parameters. Heuristic methods, like Expectation Maximization (EM) (Bailey and Elkan 1994) and Gibbs sampling methods(Lawrence, Altschul et al. 1993), are usually adopted to perform optimization. Actually Gibbs sampling is a stochastic equivalent of EM. One of the strengths of probabilistic based methods is the capability to identify motifs with complex patterns. Many potential motifs can also be identified, which actually is also a weakness, because it is difficult to distinguish the real one among them. Other limitations include: longer computational time, lack of unique solution due to the inherent randomness of the procedure, and the requirement of multiple runs.

In this work, we will adopt the word frequency based method and use the online regulatory sequence analysis tool (van Helden 2003). The analysis results show significant motifs and consensus with a significant coefficient $sig \geq 0$. $sig = 0$ means one expects one pattern to occur at random within each family. The increment of 1 for the significant coefficient sig represents a drop of 10 times in the occurrence probability. A higher significant coefficient indicates a more significant motif.

7.3 Results

7.3.1 Clustering results with different window scales

We select the same data set as in Chapter 5 and perform four level Multi-scale Fuzzy K-means clustering with window scales equal 0.15, 0.2, 0.25 and 0.3. The cluster relationships

between adjacent levels are shown in Figure 7-2. The figure only shows the links having fuzzy membership larger than 0.2. From, we can easily identify the cluster relationships and evaluate the compactness of the clusters. For example, Figure 7-2 shows cluster 5 is always separated from the other clusters across the different layers and there are only 2 clusters at the most detailed level (sc = 0.15), therefore, we can say cluster 5 at level 4 (sc = 0.3) is tightly clustered and very distinct from the other clusters. We can also see clusters 1, 3 and 4 are loosely clustered.

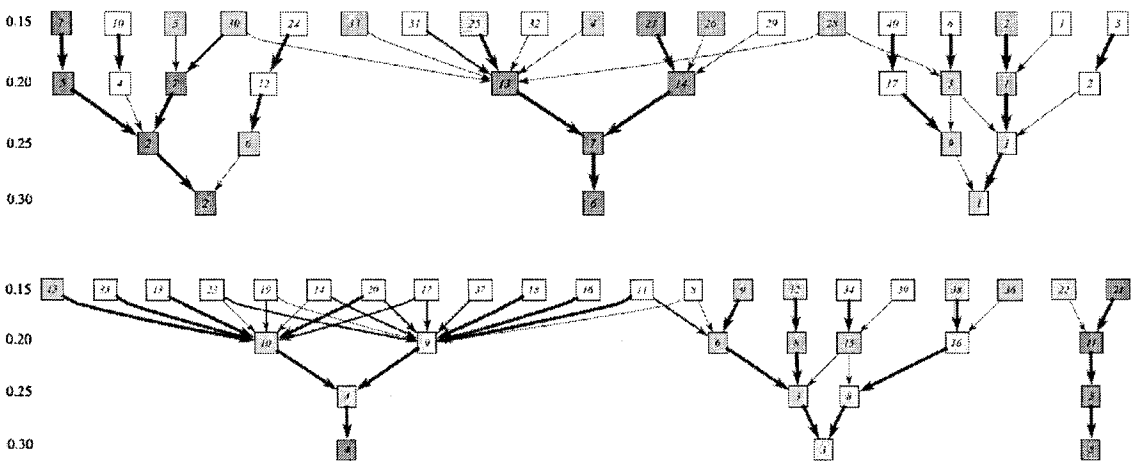


Figure 7-2 Cluster relationships between adjacent levels. The width of the line represents the significance of correlation. The wide line represents the p-value is less than 0.0001, the mid-wide line represents between 0.0001 and 0.001, the thin line represents larger than 0.001. The numbers at the left of the figure represent the clustering window scale used in this level. The number within the node box represents the cluster index in that level. The node filled with red color represents the corresponding cluster has more than 15 elements, brown color represents having 6 to15 elements.

Figure 7-3 shows some selected cluster expression profiles at different levels. We can see that the gene expression profiles of cluster 14 and 25 at level 1 (sc=0.15) are very similar. At level 2 (sc=0.2), more genes with less similar expression profiles join the clusters. At level 3 (sc=0.25), two clusters merge as one cluster. At level 4 (sc=0.3), several more genes join the cluster. This process clearly shows how the window scale controls the cluster group similarity. When the window scale is small, the algorithm has high resolution and can differentiate small difference among clusters.

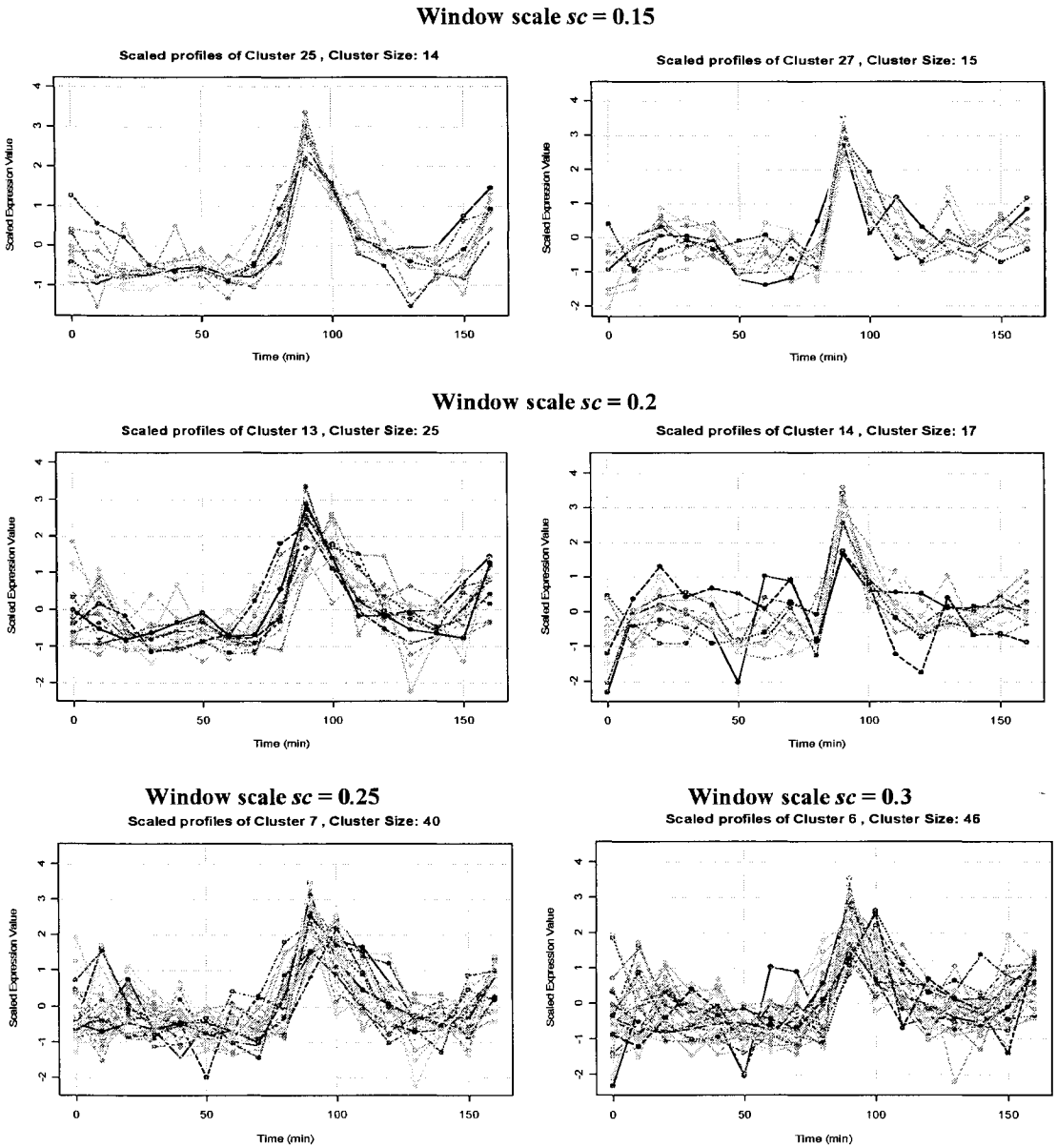


Figure 7-3 Selected cluster expression profiles in different levels

7.3.2 Cluster annotations with Gene Ontolgoy

Table 7-1 shows the cluster annotation with GO (Gene Ontology). For each cluster, we carried out the Hypergeometric test over the Biological Process GO terms, and only p-values less than 0.05 are returned. The details of cluster annotation using the Hypergeometric test are described in Section 2.3. The GO terms with the highest significance are shown in bold.

We can see almost all GO terms of big clusters (cluster size larger than 3) at the detailed level (sc = 0.15) have the highest significance among the clusters having same GO functions. This means the genes in the clusters at the detailed level usually have very similar profiles and are involved in the same or related biological processes. It also indicates that clustering is successful. In general, the clusters at the detailed level have higher specificity than the less detailed levels. However, some cluster GO annotations have higher significance at the less detailed level, e.g., "axial budding" of cluster 2 at level 3 and "chromatin assembly/ disassembly" of cluster 3 at level 3 (sc = 0.25). This indicates that the multi-scale clustering algorithm is effective at discovering clusters of genes with similar functions and different degrees of coregulation. By integrating the GO information with the networks shown in Figure 7-4, we can understand how the biological process changes over clusters in the network. Table 7-2 shows the GO Biological Process of the clusters with single genes.

Table 7-1 Cluster annotation with GO Biological Process

Cluster	Size	GO Id	GO Term	p-Value	N
A. Window scale *sc* = 0.15, GO level higher than 3					
2 (1)	3	GO:0000749	**response to pheromone during conjugation with cellular fusion**	**5.01e-07**	3
		GO:0007157	heterophilic cell adhesion	6.72e-03	1
4 (13)	2	GO:0007047	cell wall organization and biogenesis	4.78e-02	1
5 (7)	2	GO:0007157	heterophilic cell adhesion	4.48e-03	1
		GO:0000749	response to pheromone during conjugation ...	1.62e-02	1
		GO:0007047	cell wall organization and biogenesis	4.78e-02	1
7 (5)	30	GO:0007049	**cell cycle**	**7.12e-15**	20
		GO:0007120	axial budding	5.61e-03	2
		GO:0000726	non-recombinational repair	7.22e-03	2
		GO:0006302	double-strand break repair	1.24e-02	2
9 (6)	3	GO:0051053	negative regulation of DNA metabolism	5.69e-03	1
		GO:0000018	regulation of DNA recombination	6.20e-03	1
		GO:0006493	O-linked glycosylation	7.75e-03	1
		GO:0006487	N-linked glycosylation	2.31e-02	1
		GO:0006970	response to osmotic stress	2.41e-02	1
		GO:0030447	filamentous growth	3.73e-02	1
12 (8)	3	GO:0006333	**chromatin assembly or disassembly**	**7.24e-05**	2
15 (10)	2	GO:0009894	regulation of catabolism	5.17e-03	1
		GO:0051244	regulation of cellular physiological process	7.53e-03	2
		GO:0043283	biopolymer metabolism	3.51e-02	2
21 (11)	15	GO:0000074	**regulation of cell cycle**	2.42e-04	4
		GO:0007067	**mitosis**	2.97e-04	4
		GO:0007010	**cytoskeleton organization and biogenesis**	4.25e-04	5

Cluster	Size	GO Id	GO Term	p-Value	N
22 (11)	6	GO:0006268	**DNA unwinding**	**1.58e-10**	4
		GO:0006270	**DNA replication initiation**	**3.38e-09**	4
		GO:0006267	**pre-replicative complex formation and maintenance**	**1.76e-07**	3
25 (13)	14	GO:0009250	**glucan biosynthesis**	3.80e-03	1
		GO:0008151	cell growth and/or maintenance	3.86e-02	9
27 (14)	15	GO:0016043	**cell organization and biogenesis**	1.02e-05	11
		GO:0006073	glucan metabolism	3.74e-03	2
28 (3,	2	GO:0006267	pre-replicative complex formation and ...	4.48e-03	1
13)		GO:0000750	signal transduction during conjugation ...	7.58e-03	1
30 (7)	9	GO:0000910	**Cytokinesis**	1.72e-04	3
		GO:0030468	**establishment of cell polarity** (sensu Fungi)	5.61e-03	2
36 (16)	2	GO:0000114	G1-specific transcription in mitotic cell cycle	4.83e-03	1
		GO:0007047	cell wall organization and biogenesis	4.78e-02	1
38 (16)	5	GO:0006333	chromatin assembly/disassembly	2.17e-09	4
B. Window scale *sc* = 0.2, GO level higher than 3					
5 (2)	21	GO:0007049	cell cycle	**1.39e-13**	16
		GO:0006468	*protein amino acid phosphorylation*	3.29e-05	5
		GO:0007120	axial budding	2.77e-03	2
		GO:0000726	**non-recombinational repair**	3.57e-03	2
		GO:0006302	**double-strand break repair**	6.17e-03	2
7 (2)	18	GO:0030468	establishment of cell polarity (sensu Fungi)	2.21e-02	2
		GO:0008283	cell proliferation	2.86e-02	5
9 (4)	8	GO:0016043	cell organization and biogenesis	9.63e-03	5
10 (4)	6	GO:0000074	regulation of cell cycle	1.91e-03	3
		GO:0000086	*G2/M transition of mitotic cell cycle*	4.65e-03	2
		GO:0000082	*G1/S transition of mitotic cell cycle*	9.46e-04	2
11 (5)	20	GO:0006270	DNA replication initiation	**1.05e-06**	4
		GO:0006268	DNA unwinding	**7.58e-06**	3
		GO:0000074	regulation of cell cycle	7.88e-04	4
		GO:0006267	pre-replicative complex formation and ...	8.63e-04	2
		GO:0007067	mitosis	9.66e-04	4
		GO:0000114	G1-specific transcription in mitotic cell cycle	1.00e-03	2
13 (7)	25	GO:0000750	*signal transduction during conjugation ...*	3.92e-03	2
		GO:0008151	cell growth and/or maintenance	1.86e-02	15
14 (7)	17	GO:0016043	cell organization and biogenesis	6.26e-05	11
		GO:0006073	glucan metabolism	4.81e-03	2
16 (8)	7	GO:0016043	cell organization and biogenesis	1.12e-05	7
C. Window scale *sc* = 0.25, GO level higher than 3					
1 (1)	5	GO:0000749	**response to pheromone during conjugation with cellular fusion**	**4.95e-06**	3
2 (2)	38	GO:0007049	cell cycle	**4.06e-12**	20
		GO:0007120	**axial budding**	3.68e-04	3
		GO:0007534	gene conversion at MAT locus	2.22e-03	2

Cluster	Size	GO Id	GO Term	p-Value	N
3 (3)	9	GO:0007050	cell cycle arrest	1.54e-02	1
		GO:0016043	cell organization and biogenesis	1.82e-02	5
4 (4)	12	GO:0008151	cell growth and/or maintenance	1.58e-03	10
		GO:0006139	nucleobase, nucleoside, nucleotide and nucleic acid metabolism	3.49e-02	6
5 (5)	20	GO:0006270	DNA replication initiation	**1.05e-06**	4
		GO:0006268	DNA unwinding	**7.58e-06**	3
		GO:0000074	regulation of cell cycle	7.88e-04	4
		GO:0006267	pre-replicative complex formation and ...	8.63e-04	2
		GO:0007067	mitosis	9.66e-04	4
		GO:0000114	G1-specific transcription in mitotic cell cycle	1.00e-03	2
		GO:0007010	cytoskeleton organization and biogenesis	1.82e-03	5
7 (6)	40	GO:0008283	cell proliferation	5.12e-03	10
		GO:0016043	cell organization and biogenesis	6.59e-03	15
8 (3)	11	GO:0016043	cell organization and biogenesis	**7.61e-07**	10
		GO:0006073	**glucan metabolism**	1.99e-03	2
		GO:0016051	carbohydrate biosynthesis	5.64e-03	2
D. Window scale *sc* = 0.3, GO level higher than 3					
1	7	GO:0000749	response to pheromone during conjugation with cellular fusion	1.71e-05	3
2	40	GO:0007049	cell cycle	**1.24e-12**	21
		GO:0007120	axial budding	4.28e-04	3
		GO:0007534	gene conversion at MAT locus	2.46e-03	2
		GO:0000902	cellular morphogenesis	6.16e-03	5
3	12	GO:0006333	**chromatin assembly/disassembly**	**1.40e-09**	5
		GO:0007047	cell wall organization and biogenesis	2.33e-03	3
		GO:0006486	protein amino acid glycosylation	7.61e-03	2
4	15	GO:0008151	cell growth and/or maintenance	1.28e-03	13
		GO:0009250	glucan biosynthesis	7.35e-03	2
		GO:0006970	response to osmotic stress	6.06e-03	2
5	20	GO:0006270	DNA replication initiation	**1.05e-06**	4
		GO:0006268	DNA unwinding	**7.58e-06**	3
		GO:0000074	regulation of cell cycle	7.88e-04	4
		GO:0006267	pre-replicative complex formation and ...	8.63e-04	2
		GO:0007067	mitosis	9.66e-04	4
		GO:0000114	G1-specific transcription in mitotic cell cycle	1.00e-03	2
		GO:0007010	cytoskeleton organization and biogenesis	1.82e-03	5
6	46	GO:0008283	cell organization and biogenesis	5.98e-04	19
		GO:0016043	glucan biosynthesis	6.89e-03	2

The number in the parenthesis after cluster index is the corresponding cluster index at the higher level as shown in Figure 7-2. The last column N represents the number of genes in the cluster which locate in the GO category. GO Terms shown in bold have the highest significance over all clusters. All p-values less than 1e-5 are shown in bold.

Table 7-2 GO Biological Process of the clusters with single gene (window scale *sc* = 0.15)

Cluster Index	Gene	Locus Id	Biological Process GO Term
1	TSL1	YML100W	response to stress; trehalose biosynthesis
3	STE6	YKL209C	peptide pheromone export
6	GAT3	YLR013W	transcription
8	KRE6	YPR159W	beta-1,6 glucan biosynthesis; cell wall organization and biogenesis
10	YFL064C	YFL064C	unkown
11	TEL2	YGR099W	telomerase-dependent telomere maintenance
13	HSL7	YBR133C	G2/M transition of mitotic cell cycle; regulation of progression through cell cycle
14	GIC1	YHR061C	Rho protein signal transduction; axial bud site selection, establishment of cell polarity (sensu Fungi); regulation of exit from mitosis
16	NDD1	YOR372C	G2/M-specific transcription in mitotic cell cycle
17	HOS3	YPL116W	histone deacetylation
18	ARP7	YPR034W	chromatin remodeling
19	STB1	YNL309W	G1/S transition of mitotic cell cycle
20	CLB4	YLR210W	G2/M transition of mitotic cell cycle; S phase of mitotic cell cycle; regulation of cyclin dependent protein kinase activity
23	SIM1	YIL123W	microtubule cytoskeleton organization and biogenesis
24	YER189W	YER189W	unknown
31	PDR16	YNL231C	phospholipid transport; response to drug; sterol biosynthesis
32	CHS1	YNL192W	cell budding; cytokinesis, completion of separation
34	OPY2	YPR075C	cell cycle arrest in response to pheromone
35	SWI6	YLR182W	G1/S-specific transcription in mitotic cell cycle
37	SKN7	YHR206W	response to osmotic stress; response to oxidative stress; transcription

7.3.3 Networks created at different detail levels

Figure 7-4 shows the inferred networks at different detail levels. Figure 7-4.a shows the network with window scale *sc* = 0.15. For better visualization, only the edges with p-value less than 0.001 are shown, and the time delay error during d-separation check τ_{error} is set as 10 min. Since there are so many links, it is difficult to capture the cluster relationships at this level. However, we can identify some highly connected nodes, interesting links and sub networks.

One interesting finding is that most of the highly connected nodes with outward edges involve regulation roles. For example, cluster 7, cluster 9, TEL2 (cluster 11), HSL7 (cluster 13), GIC1 (cluster 14), cluster 15, NDD1 (cluster 16), HOS3 (cluster 17), ARP7 (cluster 18), STB1 (cluster 19), CLB4 (cluster 20), cluster 21, SWI6 (cluster 35), cluster 36, SKN7 (cluster 37) all involve different kinds of regulation roles. Especially the clusters including cell cycle related transcription factors, like SWI6, NDD1, cluster 15 (MBP1, FKH1), cluster 7 (SWI4), cluster 21 (ACE2, MCM1), cluster 36 (SWI5) are all highly connected and have multiple outward edges. There are some other interesting nodes as described in the following.

Figure 7-4.a shows cluster 9 is highly negatively connected. Cluster 9 includes MSB2 and HHO1. MSB2 involves signal transduction, and HHO1 involves negative regulation of DNA recombination. Cluster 9 has a strong negative link with cluster 7, whose biological process includes "cell proliferation" and "cell organization and biogenesis". This matches the functions of HHO1.

The major biological process of HOS3 (Cluster 17) is histone deacetylation. Histone deacetylation is associated with repression of gene activity through controlling chromatin activity. Figure 7-4.a shows HOS3 represses cluster 9 and 12. The significant biological process of cluster 12 is "chromatin assembly or disassembly". The significant biological process of cluster 9 includes "regulation of DNA recombination". HOS3 activates cluster 22, whose significant biological process includes "DNA unwinding and replication initiation". All of these match the function of HOS3.

Figure 7-4.a shows cluster 21 and 22 are two of the most highly connected nodes. The genes in cluster 21 have regulation roles. The major Biological Processes of cluster 22 include "DNA unwinding, replication initiation; pre-replicative complex formation". Figure 7-4.a shows most nodes pointing to cluster 21 and 22 involve regulation roles. By checking the cluster relationship in Figure 7-2 and cluster cell cycle stage information in Figure 5-8, we find the genes in both cluster 21 and 22 are mainly in *M phase*. This means *M phase*, especially "DNA unwinding, replication initiation; pre-replicative complex formation", is highly regulated by transcriptional factors. Cluster 30 only includes inward edges. Its significant biological process is "cytokinesis" and "establishment of cell polarity". So cluster 30 has no regulation roles, and the network topology of cluster 30 matches its function.

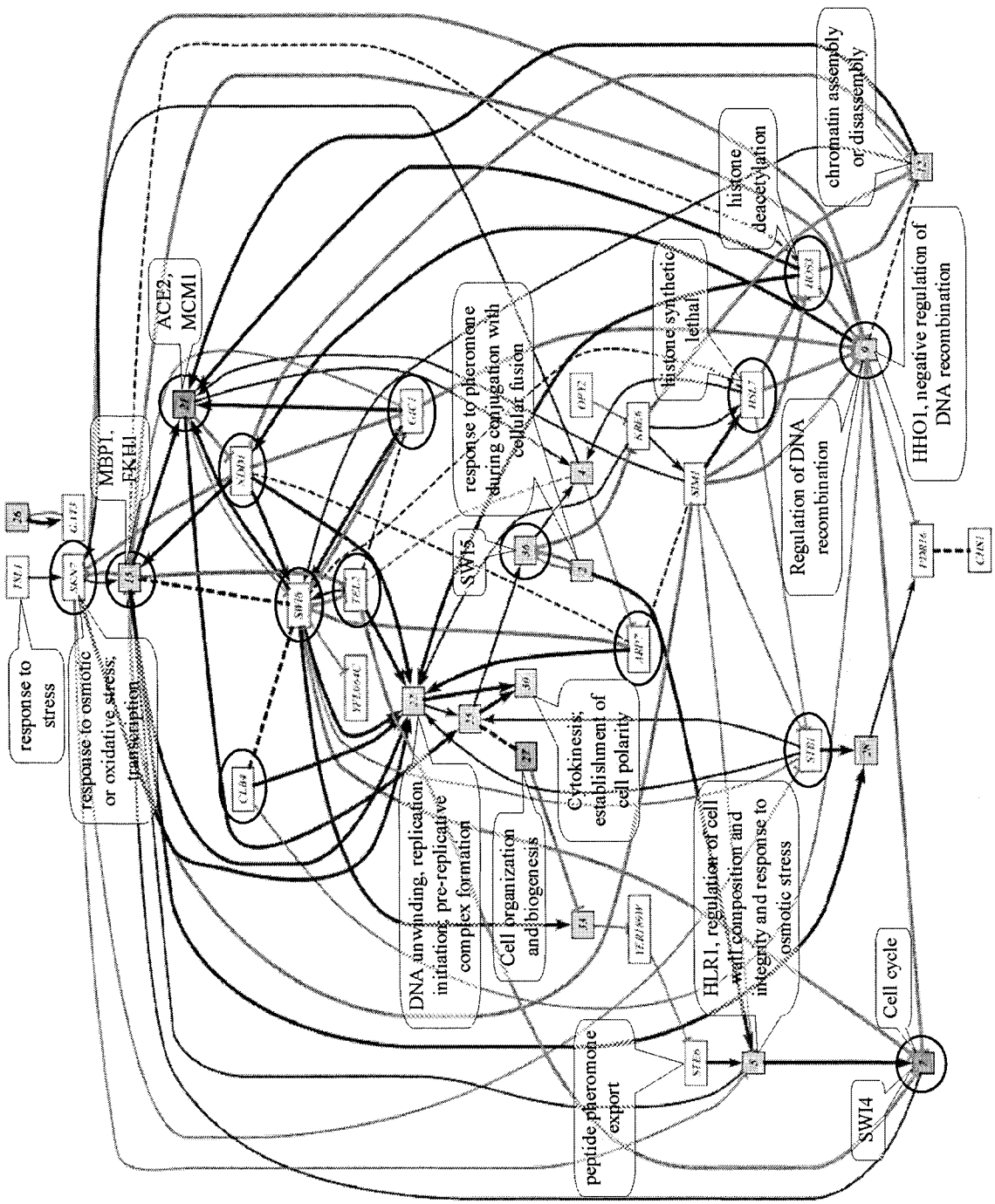
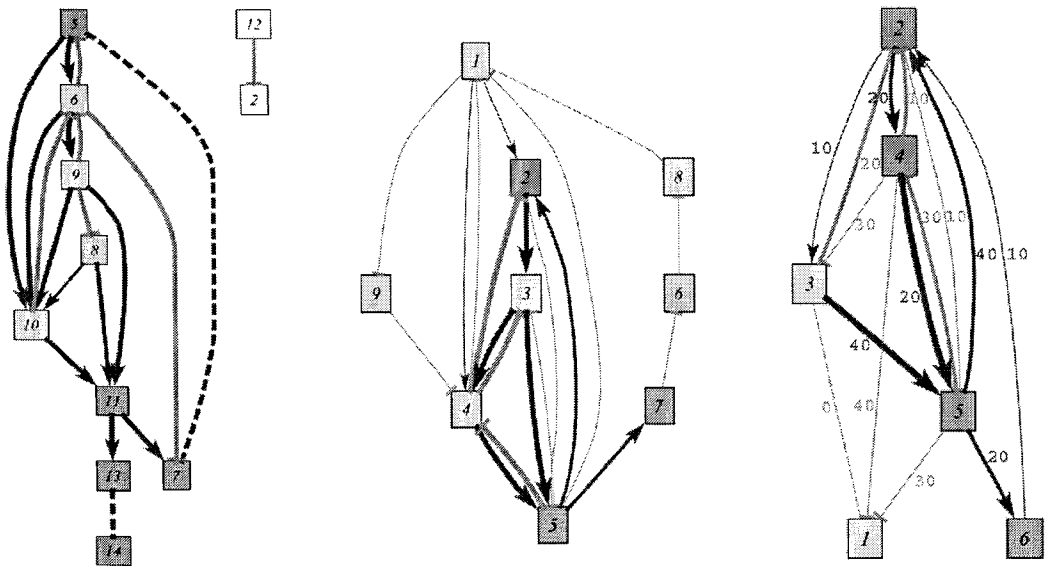


Figure 7-4.a Window scale *sc* = 0.15, τ_{error} = 10 min, T_p = 0.001. Slide window size M = 10. The circled nodes represent nodes having regulation roles. The annotations pointing to the nodes represent the genes located in the corresponding cluster or the major Biological Processes of the cluster.



b. $sc = 0.2$, $\tau_{error} = 0$, $T_p = 0.001$ c. $sc = 0.25$, $\tau_{error} = 0$, $T_p = 0.05$ d. $sc = 0.3$, $\tau_{error} = 0$, $T_p = 0.05$

Figure 7-4 Genetic networks at different levels. In the annotation, *sc* is clustering window scale, t_{Error} is the time delay error allowed during d-separation check, T_p is the p-value threshold, only the edges with p-values less than T_p are shown. Other graph annotation is same as Figure 7-2.

Apart from some interesting nodes, there are also some interesting interactions. Figure 7-4.a shows TSL1 (cluster 1) activates SKN7 (cluster 37). TSL1 participates in the biological process "response to stress". SKN7 participates in the biological process "response to osmotic or oxidative stress" and "transcription". Based on this information, we can hypothesize that SKN7 itself may not directly respond to the stress. Instead, SKN7 responds to the stress through TSL1 and plays a transcription regulation role.

Cluster 2 has the significant GO biological process "response to pheromone during conjugation with cellular fusion". STE6 (cluster 3) has the significant GO biological process "peptide pheromone export". Cluster 5 includes gene HLR1. HLR1is involved with the "regulation of cell wall composition and integrity and response to osmotic stress". Figure 7-4.a shows cluster 2 and STE6 → cluster 5 → cluster 7, i.e., "response to pheromone" and "peptide pheromone export" → "regulation of cell wall composition" → "cell cycle" related biological process. So the network topology matches the gene functions just described.

Figure 7-4.a shows many other interesting links. The explanation of these links requires more biological knowledge of the yeast cell cycle and is still under exploration. Since it is not the emphasis of this research work, we will not get into depth.

Figure 7-4.b, c and d show the networks with clustering window scale $sc = 0.2$, 0.25 and 0.3. We can see how networks are simplified as the window scale increases. Chapter 5 provides the detailed description of the network with window scale 0.3, as shown in Figure 7-4.d. Based on Figure 7-4.d, we can easily match the cell cycle development stage information, transcription factor relationships. However, this kind of information is difficult to identify in the detailed level as shown in Figure 7-4.a. Therefore, the network analyzing process is the combination of the network information at different levels. We can study the networks with a large window scale first, and then study the networks at a more detailed level by comparing the cluster relationships shown in Figure 7-2.

Network created with Short-time Correlation based algorithm

Figure 7-4.a shows the network at detailed level. By checking the distribution of the cell cycle related TFs, we cannot identify cycles linking these TFs together. The possible reason is that standard time correlation cannot identify the transient interactions, as described in Chapter 6. In order to catch these interactions, we adopted Short-time Correlation based algorithm at the detailed level. The network at each time frame will not be described in detail here. Figure 7-5 shows network created with Short-time Correlation based algorithm with window size M = 10. The network is composed of significant edges (p-value $\leq$ 0.0001) at all time frames.

Comparing with Figure 7-4.a, Figure 7-5 clearly shows the cycles linking the cell cycle related TFs. Multiple cycles can be identified: 22 → 7 → 9 → 21 → 22 with period 80min; 22 → 9 → 21 → 22 with period 80min; 22 → 7 → NDD1 → SWI6 → 22 with period 90min; 22 → 7 → NDD1 → 22 with period 90min; 22 → 7 → 15 → 36 → 22 with period 90min; 22 → 7 → 15 → 22 with period 90min; 22 → 7 → 15 → 21 → 22 with period 90min; 22 → 7 → NDD1 → TSL1 → 21 → 22 with period 90min. All of these cycles have the period approximately same as the real cell cycle period (about 80min). Among these nodes, we can see cluster 22 is the crucial node because all cycles will pass cluster 22. Also cluster 7, 9, 21 and NDD1 play important roles in cell cycle because multiple cycles passing these nodes.

The role of TSL1 in the cycle has never been reported before. This provides us hypothesis for further study.

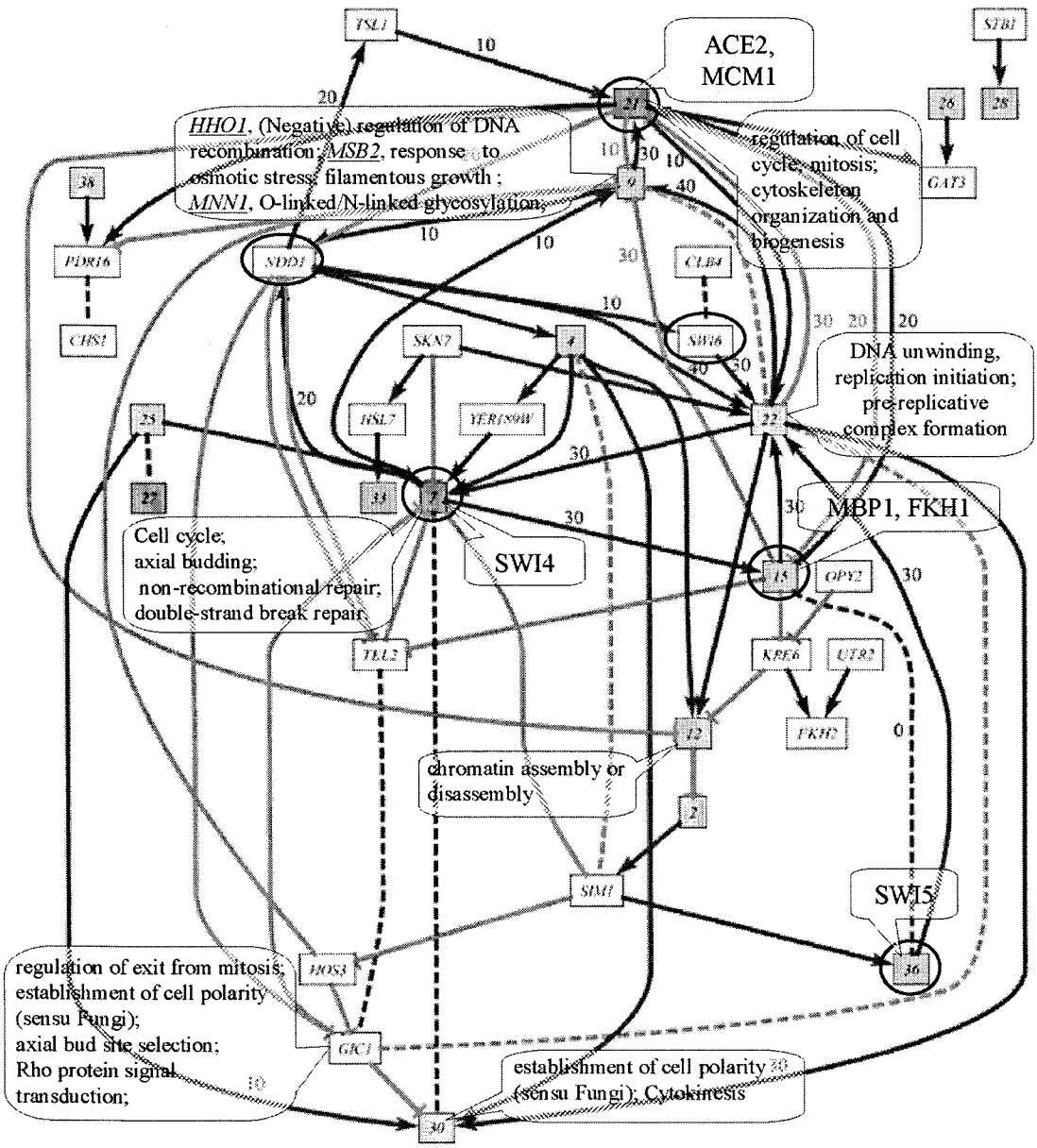


Figure 7-5 Genetic network created with Short-time Correlation based algorithm at detail level (sc=0.15). The network is composed of significant edges (p-value ≤ 0.0001) at all time frames. Slide window size M = 10. The label beside the edge represents time delay over the edge. The circled nodes include cell cycle related transcription factor. Other graph annotation is same as Figure 7-4.a.

Comparing with Figure 7-4.a, Figure 7-5 shows some new significant interactions. One major Biological Processes of cluster 22 is "DNA unwinding, replication initiation". The

major Biological Processes of cluster 22 is "chromatin assembly or disassembly". In Figure 7-5, a new edge from cluster 22 to cluster 12 matches the relationships of their Biological Process. Another interesting finding is the edges from gene GIC1. GIC1 involves the Biological Processes of "regulation of exit from mitosis", "establishment of cell polarity (sensu Fungi)" and "axial bud site selection". Figure 7-5 shows three negative edges from GIC1: GIC1 → cluster 22, GIC1 → cluster 30 and GIC1 → cluster 7. By checking the cluster relationship in Figure 7-2 and cluster cell cycle stage information in Figure 5-8, we know the genes in cluster 22 are in *M phase*. This matches the negative link from GIC1 to cluster 22. The major Biological Processes of cluster 30 and cluster 7 include "establishment of cell polarity (sensu Fungi)" and "axial budding" respectively. Both of them match the links GIC1 → cluster 30 and GIC1 → cluster 7. Figure 7-5 shows cluster 4 is one of the highly connected nodes. Multiple edges are out from cluster 4. Cluster 4 has two genes, PIR3 and YKL151C. The Biological Process of PIR3 is "cell wall organization and biogenesis". The function of gene YKL151C is uncharacterized. The Biological Process of PIR3 cannot explain the relationships with other clusters. One hypothesis is that gene YKL151C may play some regulation roles.

Need to mention, because Figure 7-5 only shows the edges with p-value ≤ 0.0001, some potential links and cycles are not shown. Also because short-time correlation only uses part of the profile length, the significant of the global interactions will be lower by using short-time correlation. As a result, some interactions shown in Figure 7-4.a are not shown in Figure 7-5. Therefore, in order to capture more information, we should use both standard time correlation and short-time correlation method.

7.3.4 Regulatory sequence analysis

We used online regulatory sequence analysis tools (http://rsat.ulb.ac.be/rsat/) (van Helden 2003) to conduct regulatory sequence analysis within 800 bp upstream of the ORFs of each cluster. The upstream sequences overlapping with other ORF were discarded. Large duplicated regions (>= 40 bp alignment with less than 3 mismatches) are filtered out before analysis. The results are organized in Table 7-3. For each significant motif, we indicate the most significant cluster id ("cluster index"_"clustering level") and significance score. Other

clusters having the same motif are also shown in the table. The promoter information is retrieved from the search database SCPD (The Promoter Database of *Saccharomyces cerevisiae*) (http://rulai.cshl.edu/SCPD/). From Table 7-3, we can see the most significant motifs locate at different clustering levels. A possible explanation is that the degrees of coregulation for different promoters are different. If we use common clustering algorithms, like K-means, it would be difficult to identify all of these motifs.

Table 7-3 Significant motifs and corresponding Transcription Factors

Motif	Cluster of most significant motif	Other clusters having the same motif	Promoter
ACGCGT	7_1 (14.14)	5_2 (12.08), 2_3 (11.50), 2_4 (12.92)	**MCB** (8/8)
CACGAA	7_1 (3.82)	30_1(1.43), 5_2(1.91), 7_2 (2.79), 2_3(3.78), 2_4(3.5)	**SCB,CCBF,SWI6,SWI4** (2/5); ABF1,BAF1; REB1; MAL63
ACGCGA	7_1 (5.55)	5_2(1.29), 2_3(4.91), 2_4(4.61)	**MCB** (2/2)
CGCGAA	2_3 (3.78)	7_2(2.79), 8_3(0.81), 2_4(3.50)	**SCB,CCBF,SWI6,SWI4** (2/5); MAL63; ABF,BAF; REB1
AACAAA	3_4 (1.68)		ABF1,BAF1; UASH; UASPHR
AAACAA	4_4 (2.30)	9_2(0.81), 4_3(2.18)	ROX1; ABF1,BAF1; **SFF**
TAGGAA	21_1 (0.49)		**MCM1** (14/19); GCR1; HSE,HSTF; PQBOX; UASH;
TAAACA	11_2 (1.30)	5_3 (0.78)	**SFF** (2/6); MAL63; ROX1; **MCM1**; MIG1
AGGAAA	5_3 (0.68)	5_4 (0.62)	**MCM1**(20/22); URSSGA; UASPHR
AGGGTA	27_1 (0.64)	14_2 (0.64)	**MCM1**; REB1; URSPOX1
CCAGCA	7_3 (2.96)	25_1 (1.12), 13_2 (2.58), 6_4 (2.91)	**SWI5** (5/8); CUP2; **ACE2**; PHO4;
CATCCA	6_4 (1.49)	14_2 (0.33), 7_3 (1.13)	GCR1; UASPHR

The cluster id is in the format "cluster index"_"level", e.g., "7_1" represents cluster 7 at level 1 (sc=0.15). The number after cluster id is the significance score of the motif in the cluster. The promoters related with cell cycle are shown in bold. The number after the promoter indicates the promoter record number over total record number of this motif in the database. E.g., MCM1 (14/19) represents there are 19 records having motif TAGGAA, and 14 of them correspond to promoter MCM1. If there is only one record of the promoter in the database, no number will be shown.

After identifying the significant motifs, we can search the upstream of ORFs to see how the motifs are distributed over the upstream of ORFs. Figure 7-6 shows an example of the

motif distribution at the upstream of genes in cluster 3 at level 4 ($sc = 0.3$). If the gene in the cluster has the identified regulatory sequence motif at its sequence upstream, we can assume that this gene may be regulated by the transcription factor corresponding to the motif. Coregulated genes may have different time delays, so some of them may locate in the nearby clusters. Therefore, we can also search the genes in the nearby clusters having direct links with the cluster under consideration. After we obtain the matching information between TF and regulated genes, we can create a more detailed network by combining this information with the genetic network of clusters. The first step is to identify the distribution of the TFs over the network and get the relationships of TFs, then extend the links from the TF to the regulated genes which are identified based on the motif information and produce a more detailed gene transcriptional regulatory network. Usually, there will be some genes for which we cannot identify TF motifs at the upstream of their ORFs. More sophisticate regulatory sequence analysis can be applied, and further information may be required to determine the regulatory relationships of these genes.

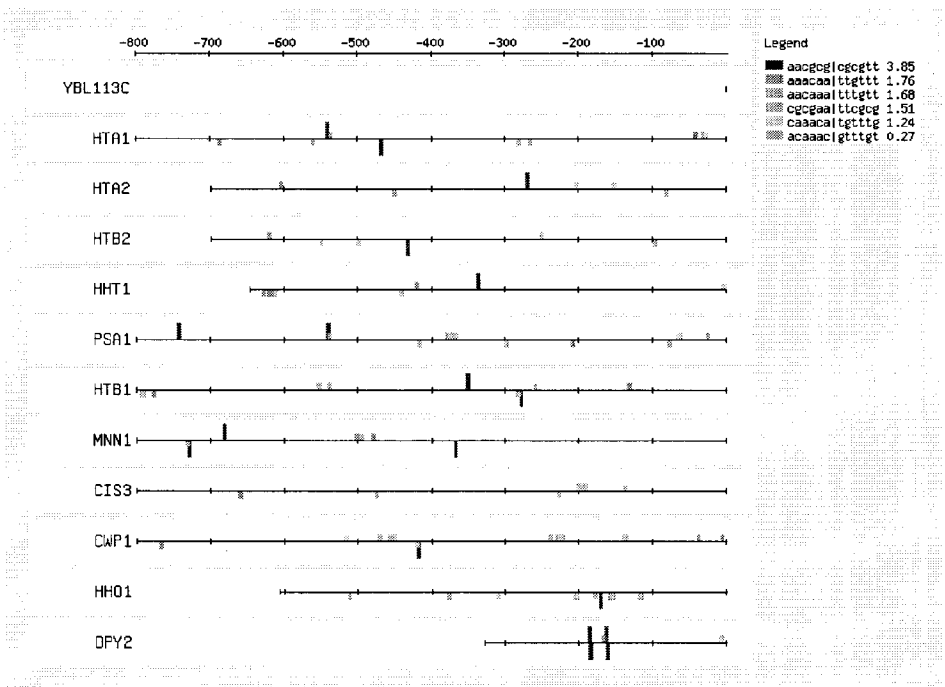


Figure 7-6 Motif distribution at the upstream of genes in cluster 3 at level 4 (sc=0.3)

7.3.5 Combine the regulatory sequence results in genetic network

Table 7-3 shows the significant motifs and their distribution over the clusters. Based on the cluster relationships at different levels, as shown in Figure 7-2, we can calculate the distribution of the motifs and corresponding promoters over the clusters at level 4 (cluster window size $sc = 0.3$). Here we just list the distribution of the cell cycle related promoters: cluster 2 (MCB, SCB), cluster 4 (SFF), cluster 5 (MCM1, SFF), cluster 6 (ACE2, MCM1, SWI5). By combining the cluster promoter information with the cluster relationships shown in Figure 6-7.b, we can produce the graph as shown in Figure 7-7.

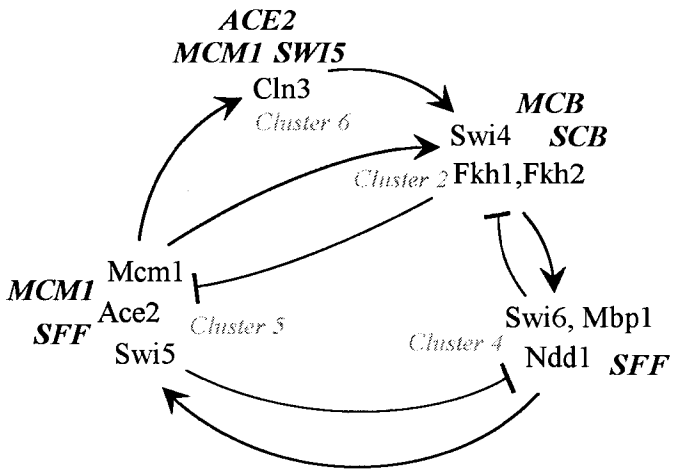


Figure 7-7 Inferred transcription factor relationships by integrating the sequence promoter information. The relationships of transcription factors are based on Figure 6-7.b. The names in italic bold represent the promoters identified in the cluster, as shown in Table 7-3.

As shown in (Wittenberg and Reed 2005), MCB (Mlu Cell cycle Box) is the optimal binding site for MBF (Mbp1). SCB is the Swi4 Cell cycle Box and is the optimal binding site for SBF (Swi4). Swi4 or Mbp1 and a common component Swi6 are two alternative heterodimeric transcription factors in the G1 gene cluster. The binding sites for Swi4 and Mbp1 have considerable overlap. This matches our results shown in Figure 7-7, i.e., both MCB and SCB were identified in cluster 2. SFF is the Swi5 factor. Previous studies show Fkh1 and Fkh2 are also capable of binding to SFF sites in vitro and in vivo (Kumar, Reynolds et al. 2000; Pic, Lim et al. 2000; Hollenhorst, Pietz et al. 2001). As shown in Figure 7-7, the distribution of SFF matches both of these claims, because Fkh1 and Fkh2 are

located in cluster 2 and the promoter SFF was identified in cluster 4; also, Swi5 is located in cluster 5, and SFF was also identified in cluster 5. Ndd1 in cluster 4 has no apparent DNA-binding domain. Ndd1 transactivation activity depends on binding to the forkhead associated (FHA) domain of Fkh2 (Koranda et al.,2000). The promoter SFF identified in cluster 4 and 5 seems matching this fact. Transcription factor Mcm1 regulates a large group of genes in *M-G1 phase*. This again matches Figure 7-7, because MCM1 was identified in both cluster 5 (mainly *M phase*) and cluster 6 (mainly *early G1 phase*). Figure 7-7 shows that the promoters ACE2, MCM1 and SWI5 were identified in cluster 6, and this matches the network topology as the corresponding transcription factor genes Ace2, Mcm1 and Swi5 are located in the direct parent cluster 5. Therefore, the network topology and identified promoter information match the literature very well.

7.4 Discussions

In this chapter, we combined the algorithms described in previous chapters with regulatory sequence analysis. We use the Multi-scale Fuzzy K-means clustering algorithm to cluster gene expression profiles at different detail levels. In the most detailed level, we can identify clusters with very similar expression profiles. Cluster Gene Ontology annotation results show very significant GO Biological Processes in these clusters. Detail interactions, highly connected genes or clusters can be identified for further study. At less detailed levels, other less significant GO Biological Processes can be identified. A similar situation occurs for the integration of regulatory sequence analysis. Significant regulatory sequence motifs can be identified in the clusters at different detailed levels. By combining this motif and promoter information with the genetic network of clusters, we get biological explanations matching previous literature results. Therefore, Multi-scale Fuzzy K-means clustering algorithm provides a powerful way to capture coregulated genes with different degree of coregulations and genes involving in similar biological processes or functions. By integrating genetic networks created based on the cluster centers and the cluster motif information, we can get more detailed gene regulatory relationships and provide reasonable biological explanations and hypotheses.

One potential problem of identifying regulatory sequence motifs based on the Multi-scale Fuzzy K-means clustering algorithm is false positive detection. This can be relieved by combining the information between clusters at different levels. For real motifs, we suppose they may also exist at adjacent detailed levels. Also, different regulatory sequence analysis algorithms can be applied. In this way, we can find more potential motifs and know which motifs are more conserved among all the results. Other prior knowledge and data analysis results can also be combined to resolve the real regulatory relationships.

ChIP-chip (ChIP is the abbreviation of Chromatin ImmunoPrecipitation) binding assays (Ren, Robert et al. 2000; Lee, Rinaldi et al. 2002) is one method of verifying the relationships between Transcription Factor (TF) and cis-regulatory elements. Just like microarray measures the mRNA accumulation levels on the genome scale, ChIP-chip data can monitor the protein-DNA interactions across the entire genome. But the ChIP-chip binding assay also tends to detect many false positive target genes, and the binding information alone is not enough to determine the regulatory roles of the TFs. Usually multiple TFs can be detected binding at the upstream, but that does not mean all of them will act together and participate in the same regulatory process. Therefore, integration of ChIP-chip data, regulatory sequence analysis and microarray data is necessary to better identify the relationships between TFs and cis-regulatory elements.

CHAPTER 8. CONCLUSIONS

8.1 Summary

In this work, we try to infer genetic networks based on time series gene expression data. Gene expression data are noisy, large scale and with groups of gene expression profiles coregulated. Clustering is used to find coregulated genes and serves as a preprocess step for genetic network inference. Genes can participate in different biological processes and be coregulated with different groups of genes, so clustering gene expression profiles is a fuzzy process. Also, the degree of coregulation can be quite different between among groups of coregulated genes. In order to capture this kind of information, we proposed Multi-scale Fuzzy K-means clustering algorithm. Gene Ontology cluster annotation and regulatory sequence analysis results show our clustering algorithms are effective. Very significant Biological Processes were identified in the highly coregulated genes. From analysis based on the clusters at different levels, all major cell cycle related promoters were identified and the results match those in the literature.

Time series expression profiles provide dynamic information for inferring gene regulatory relationships. However, the time profile length is very limited, complex models cannot be adopted. Instead, we use pair wise time correlation to capture the pairwise linear relationships among genes or clusters. One major problem of correlation is that there are many indirectly caused correlations. In order to differentiate the direct and indirect interactions, we propose the CBTC network inference algorithm which integrates d-separation and partial correlation theory with time correlation. Constraints are imposed during d-separation checks to decrease the false deletion rate. The results for simulation and yeast cell cycle data show that the CBTC algorithm identified most interactions, cell cycle loops, negative feed back loops and the distributions of TFs over the network matching literature results.

Time correlation is based on entire expression profiles, but gene interactions can happen within specific time and conditions instead of across the whole expression profile. In order to

capture these transient interactions, we propose short-time correlation based genetic network inference. With the variable-sized sliding window, we can capture the transient interactions with different length of duration. The networks at different time frames reflect the changes of network topologies over time. By visualizing the short-time correlation coefficients under different parameter dimensions, we can capture detailed interaction changing over parameter values. Results show some previously missing interactions and the most significant time interval for each interaction.

Finally, we integrate gene regulatory sequence information with genetic network inference. Based on the Multi-scale Fuzzy K-means clustering results, all major cell cycle related motifs were identified. By combining the genetic networks with the promoter information corresponding to the motifs, we can obtain a reasonable biological explanation and provide hypotheses for future biological studies.

Unlike other optimization-based heuristic methods, our genetic network inference algorithms are straightforward, efficient and open box. Users can know how the networks are created or the edges are deleted. As the cost of microarray chips decrease, an increasing number of time samples will be available. This will help us extract more detailed dynamic information, especially when we use the short-time correlation method. Based on the inferred genetic networks, we can select some interesting genes to create a sub-network with more detailed models. Thus, our genetic network inference algorithm can also be used as a preprocessing step for more complex models.

8.2 Limitations and future work

As we just described, correlation based genetic network inference aims to detect the pair wise linear relationships. It will have difficulties in dealing with complex relations. Short-time correlation based network inference algorithms can capture transient interactions and deal with more complex situations if more time samples can be provided.

Our network inference is based on cluster centers. By integrating regulatory sequence analysis results, we can resolve more detailed transcription regulatory relationships among genes. But there are still lots of genes for which we cannot resolve regulatory relationships either because we cannot identify significant motifs and corresponding promoters or there are

conflicts between results. In order to resolve their regulatory relationships, integration of more information is needed. Next, we will list some data or resources which can be integrated to achieve better performance.

Currently, there is a vast number of microarray data available, which includes both static data and time series data with different profile lengths. The integration of these data to infer more reliable networks is a challenging and practical problem.

As we briefly described in Chapter 7, ChIP-chip data provides direct TF binding information for the genes. If we can integrate this information with microarray data and regulatory sequence information, we can achieve better results.

Just like studying evolution based on genome sequence, we can also study genetic networks in an evolutionary way. By comparing the inferred genetic networks between evolutionarily related species, we can identify how genetic networks evolve. If some sub-networks are conserved between different species, we will have more confidence to say these inferred sub-networks are correct.

A more ambitious plan is to integrate genomics (genome sequence), transcriptomics (microarray), proteomics and metabolomics data and related prior knowledge. Actually, this is one of the major tasks of systems biology. It is definitely challenging and there is still a long way to go.

REFERENCES CITED

Affymetrix Inc. (2001). Statistical Algorithms Reference Guide. Santa Clara, CA, Affymetrix, Inc.

Akutsu, T., S. Miyano, et al. (1999). Identification of Genetic Networks from a Small Number of Gene Expression Patterns Under the Boolean Network Model. Pacific Symposium on Biocomputing 4, Hawaii.

Akutsu, T., S. Miyano, et al. (2000). Algorithms for Inferring Qualitative Models of Biological Networks. Pacific Symposium on Biocomputing 5, Hawaii.

Al-Shahrour, F., R. Diaz-Uriarte, et al. (2004). "FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes." Bioinformatics **20**(4): 578-80.

Arkin, A., J. Ross, et al. (1998). "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells." Genetics **149**(4): 1633-48.

Arkin, A., P. Shen, et al. (1997). "A test case of correlation metric construction of a reaction pathway from measurements." Science **277**(29): 1275-1279.

Ashburner, M. and S. Lewis (2002). "On ontologies for biologists: the Gene Ontology - uncoupling the web." In Silico Biology Novartis Found Symp **247**: 66-80; discussion 80-3, 84-90, 244-52.

Bailey, J. E. (1999). "Lessons from metabolic engineering for functional genomics and drug discovery." Nat Biotechnol **17**(7): 616-8.

Bailey, T. L. and C. Elkan (1994). "Fitting a mixture model by expectation maximization to discover motifs in biopolymers." Proc Int Conf Intell Syst Mol Biol **2**: 28-36.

Ball, G. H. (1965). "Data analysis in the social sciences: what about the details." AFIPS Proc. Cong. Fall Joint Comp. **27**(1): 533-559.

Ball, G. H. and D. J. Hall (1965). ISODATA, a novel method of data analysis and pattern classification, Stanford Research Institute.

Bar-Joseph, Z. (2004). "Analyzing time series gene expression data." Bioinformatics **20**(16): 2493-503.

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

Blake, J. and M. Harris (2003). The Gene Ontology Project: Structured vocabularies for molecular biology and their application to genome and expression analysis. Current Protocols in Bioinformatics. D. B. D. A.D. Baxevanis, R. Page, G. Stormo and L. Stein. New York, Wiley and Sons, Inc.

Bolouri, H. and E. H. Davidson (2002). "Modeling transcriptional regulatory networks." Bioessays **24**(12): 1118-29.

Chen, T., H. L. He, et al. (1999). "Modeling gene expression with differential equations." Pac Symp Biocomput: 29-40.

Cheng, Y. and G. M. Church (2000). "Biclustering of expression data." Proc Int Conf Intell Syst Mol Biol **8**: 93-103.

Cho, R. J., M. J. Campbell, et al. (1998). "A genome-wide transcriptional analysis of the mitotic cell cycle." Mol Cell **2**(1): 65-73.

Covert, M. W., C. H. Schilling, et al. (2001). "Metabolic modeling of microbial strains in silico." Trends Biochem Sci **26**(3): 179-86.

Cox, Z., A. Fulmer, et al. (2002). Interactive Graphs for Exploring Metabolic Pathways. ISMB, 2002, Edmonton, CA.

de Jong, H. (2002). "Modeling and simulation of genetic regulatory systems: a literature review." J Comput Biol **9**(1): 67-103.

de la Fuente, A., N. Bing, et al. (2004). "Discovery of meaningful associations in genomic data using partial correlation coefficients." Bioinformatics **20**(18): 3565-74.

de la Fuente, A., P. Brazhnik, et al. (2002). "Linking the genes: inferring quantitative gene networks from microarray data." Trends Genet **18**(8): 395-8.

D'Haeseleer, P. (2000). Reconstructing Gene Networks from Large Scale Gene Expression Data. Computer Science. Albuquerque, NM, The University of New Mexico**:** 207.

D'Haeseleer, P., S. Liang, et al. (1999). "Gene expression analysis and modeling." Pac Symp Biocomput(Tutorial).

D'Haeseleer, P., S. Liang, et al. (2000). "Genetic network inference: from co-expression clustering to reverse engineering." Bioinformatics **16**(8): 707-26.

Dickerson, J. A., D. Berleant, et al. (2001). Creating Metabolic Network Models using Text Mining and Expert Knowledge. Atlantic Symposium on Molecular Biology and Genome Information Systems and Technology (CBGIST 2001), Durham, North Carolina.

Dickerson, J. A., Z. Cox, et al. (2001). Creating Metabolic and Regulatory Network Models using Fuzzy Cognitive Maps. North American Fuzzy Information Processing Conference (NAFIPS), Vancouver, B.C.

Dickerson, J. A., Z. Cox, et al. (2001). Creating metabolic and regulatory network models using fuzzy cognitive maps. IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, Vancouver, BC , Canada.

Dickerson, J. A. and B. Kosko (1993). Virtual worlds as fuzzy cognitive maps. Virtual Reality Annual International Symposium, 1993., 1993 IEEE, Seattle, WA , USA.

Dickerson, J. A. and B. Kosko (1994). "Virtual Worlds as Fuzzy Cognitive Maps." Presence **3**(2, Spring): 173-189.

Du, P., J. Gong, et al. (2005). "Modeling Gene Expression Networks using Fuzzy Logic." IEEE Trans. on SMCB (Systems, Man and Cybernetics, Part B) **35**(6): (in press).

Du, P., E. S. Wurtele, et al. (2005). "Genetic Network Inference based on Time Series Expression Profiles." Bioinformatics **(submitted)**.

Eastmond, P. J. and I. A. Graham (2003). "Trehalose metabolism: a regulatory role for trehalose-6-phosphate?" Curr Opin Plant Biol **6**(3): 231-5.

Edwards, D. (2000). Introduction to graphical modelling. New York, Springer.

Edwards, J. S., R. U. Ibarra, et al. (2001). "In silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data." Nat Biotechnol **19**(2): 125-30.

Edwards, R., H. T. Siegelmann, et al. (2001). "Symbolic dynamics and computation in model gene networks." Chaos **11**(1): 160-169.

Eisen, M. B., P. T. Spellman, et al. (1998). "Cluster analysis and display of genome-wide expression patterns." Proc Natl Acad Sci U S A **95**(25): 14863-8.

Eisen, M. B., P. T. Spellman, et al. (1998). "Cluster analysis and display of genome-wide expression patterns." Proceedings National Academy of Science **95**: 14863-14868.

Famili, I., J. Forster, et al. (2003). "Saccharomyces cerevisiae phenotypes can be predicted by using constraint-based analysis of a genome-scale reconstructed metabolic network." Proc Natl Acad Sci U S A **100**(23): 13134-9.

Fatland, B. F., J. Ke, et al. (2002). "Molecular Characterization of a Novel Heteromeric ATP-Citrate Lyase that Generates Cytosolic Acetyl-CoA in *Arabidopsis*." Plant Physiology **130**: 740-756.

Foster, C. M., L. Ling, et al. (2004). "Expression of genes in the starch metabolic network of *Arabidopsis* during starch sythesis and degradation." In preparation.

Friedman, N. and D. Koller (2001). "Being Bayesian About Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks." Kluwer Academic Publishers.

Friedman, N., M. Linial, et al. (2000). "Using Bayesian networks to analyze expression data." RECOMB: 127-135.

Gasch, A. P. and M. B. Eisen (2002). "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering." Genome Biol **3**(11): RESEARCH0059.

Gautier, L., L. Cope, et al. (2004). "affy--analysis of Affymetrix GeneChip data at the probe level." Bioinformatics **20**(3): 307-315.

Gentleman, R. (2003). Hypothesis testing and GO.

Glass, L. (1975). "Classification of biological networks by their qualitative dynamics." J Theor Biol **54**(1): 85-107.

Goss, P. J. and J. Peccoud (1998). "Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets." Proc Natl Acad Sci U S A **95**(12): 6750-5.

Guet, C. C., M. B. Elowitz, et al. (2002). "Combinatorial synthesis of genetic networks." Science **296**(5572): 1466-70.

Hastie, T., R. Tibshirani, et al. (2000). "'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns." Genome Biology **1**(2): research0003.1-0003.21.

Hasty, J., F. Isaacs, et al. (2001). "Designer gene networks: Towards fundamental cellular control." Chaos **11**(1): 207-220.

Hatzimanikatis, V. and K. H. Lee (1999). "Dynamical Analysis of Gene Networks Requires Both mRNA and Protein Expression Information." Metabolic Engineering **1**: 275-281.

Hofestadt, R. (1995). "A rule based system for the detection of metabolic diseases." Medinfo **8 Pt 2**: 964-8.

Hollenhorst, P. C., G. Pietz, et al. (2001). "Mechanisms controlling differential promoter-occupancy by the yeast forkhead proteins Fkh1p and Fkh2p: implications for regulating the cell cycle and differentiation." Genes Dev **15**(18): 2445-56.

Hood, L., J. R. Heath, et al. (2004). "Systems biology and new technologies enable predictive and preventative medicine." Science **306**(5696): 640-3.

Hotelling, H. (1953). "New light on the correlation coefficient and its transforms." J. R. Statist. Soc. B **15**: 193-232.

Hughes, J. D., P. W. Estep, et al. (2000). "Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae." J Mol Biol **296**(5): 1205-14.

Irizarry, R. A., B. Hobbs, et al. (2003). "Exploration, normalization, and summaries of high density oligonucleotide array probe level data." Biostatistics **4**(2): 249-64.

Iyer, V. R., M. B. Eisen, et al. (1999). "The Transcriptional Program in the Response of Human Fibroblasts to Serum." Science **283**: 83-87.

Jensen, L. J. and S. Knudsen (2000). "Automatic discovery of regulatory patterns in promoter regions based on whole cell expression data and functional annotation." Bioinformatics **16**(4): 326-33.

Kaern, M., W. J. Blake, et al. (2003). "The engineering of gene regulatory networks." Annual Review of Biomedical Engineering **5**(1): 179-206.

Kamvysselis, M. (2003). Computational comparative genomics: genes, regulation, evolution. Department of Electrical Engineering and Computer Science. Cambridge, MA, Massachusetts Institute of Technology**:** 100.

Kato, M., T. Tsunoda, et al. (2001). "Lag analysis of genetic networks in the cell cycle of budding yeast." Genome Informatics **12**: 266-267.

Kauffman, S. A. (1969). "Metabolic stability and epigenesis in randomly constructed genetic nets." J Theor Biol **22**(3): 437-67.

Kim, S., S. Imoto, et al. (2004). "Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data." Biosystems **75**(1-3): 57-65.

Kishino, H. and P. J. Waddell (2000). "Correspondence analysis of genes and tissue types and finding genetic links from microarray data." Genome Inform Ser Workshop Genome Inform **11**: 83-95.

Kitano, H. (2002). "Systems biology: a brief overview." Science **295**(5560): 1662-4.

Kohonen, T. (1997). Self-organizing maps. Berlin ; New York, Springer.

Kosko, B. (1986). "Fuzzy Cognitive Maps." International Journal of Man Machine Studies **24**: 65-75.

Kosko, B. (1992). Neural Networks and Fuzzy Systems. Englewood Cliffs, Prentice Hall.

Kumar, R., D. M. Reynolds, et al. (2000). "Forkhead transcription factors, Fkh1p and Fkh2p, collaborate with Mcm1p to control transcription required for M-phase." Curr Biol **10**(15): 896-906.

Landahl, H. D. (1969). "Some conditions for sustained oscillations in biochemical chaines." B. Math. Biophys.(31): 775-787.

Lawrence, C. E., S. F. Altschul, et al. (1993). "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment." Science **262**(5131): 208-14.

Lee, T. I., N. J. Rinaldi, et al. (2002). "Transcriptional regulatory networks in Saccharomyces cerevisiae." Science **298**(5594): 799-804.

Liang, S., S. Fuhrman, et al. (1998). REVEAL, A general reverse engineering algorithm for inference of genetic network architectures. Pacific Symposium on Biocomputing 3, Hawaii.

Magwene, P. M. and J. Kim (2004). "Estimating genomic coexpression networks using first-order conditional independence." Genome Biol **5**(12): R100.

Marchal, K., G. Thijs, et al. (2003). "Genome-specific higher-order background models to improve motif detection." Trends Microbiol **11**(2): 61-6.

Martins, A. M., P. Mendes, et al. (2001). "In situ kinetic analysis of glyoxalase I and glyoxalase II in Saccharomyces cerevisiae." Eur J Biochem **268**(14): 3930-6.

Matsuno, H., A. Doi, et al. (2000). "Hybrid Petri net representation of gene regulatory network." Pac Symp Biocomput: 341-52.

Mendoza, L. and E. R. Alvarez-Buylla (1998). "Dynamics of the genetic regulatory network for Arabidopsis thaliana flower morphogenesis." J Theor Biol **193**(2): 307-19.

Mendoza, L., D. Thieffry, et al. (1999). "Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis." Bioinformatics **15**(7-8): 593-606.

Murphy, K., Mian, S. (1999). Modelling Gene Expression Data using Dynamic Bayesian Networks, Computer Science Division, University of California, Berkeley.

Murphy, K. P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. Computer Science, UNIVERSITY OF CALIFORNIA, BERKELEY**:** 255.

National Center for Biotechnology Information (NCBI) (2004). Microarrays: Chipping Away At The Mysteries Of Science And Medicine, NCBI. **2004**.

Pearl, J. (2000). Causality : models, reasoning, and inference. Cambridge, U.K. ; New York, Cambridge University Press.

Perrin, B. E., L. Ralaivola, et al. (2003). "Gene networks inference using dynamic Bayesian networks." Bioinformatics **19 Suppl 2**: II138-II148.

Pic, A., F. L. Lim, et al. (2000). "The forkhead protein Fkh2 is a component of the yeast cell cycle transcription factor SFF." Embo J **19**(14): 3750-61.

Reed, J. L. and B. O. Palsson (2003). "Thirteen years of building constraint-based in silico models of Escherichia coli." J Bacteriol **185**(9): 2692-9.

Ren, B., F. Robert, et al. (2000). "Genome-wide location and function of DNA binding proteins." Science **290**(5500): 2306-9.

Roth, F. P., J. D. Hughes, et al. (1998). "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation." Nat Biotechnol **16**(10): 939-45.

Savageau, M. A. (2001). "Design principles for elementary gene circuits: Elements, methods, and examples." Chaos **11**(1): 142-159.

Schafer, J. and K. Strimmer (2004). "An empirical bayes approach to inferring large-scale gene association networks." Bioinformatics.

Segal, E., M. Shapira, et al. (2003). "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data." Nat Genet **34**(2): 166-76.

Segal, E., R. Yelensky, et al. (2003). "Genome-wide discovery of transcriptional modules from DNA sequence and gene expression." Bioinformatics **19 Suppl 1**: I273-I282.

Shaw, O. J., C. Harwood, et al. (2004). "SARGE: a tool for creation of putative genetic networks." Bioinformatics **20**(18): 3638-40.

Shimada, T., M. Hagiya, et al. (1995). Knowledge-based simulation of regulatory action in lambda phage. Intelligence in Neural and Biological Systems, 1995 INBS'95, Proceedings., First International Symposium on, Herndon, VA , USA.

Shipley, B. (2002). Cause and correlation in biology : a user's guide to path analysis, structural equations and causal inference. Cambridge, U.K. ; New York, Cambridge University Press.

Shmulevich, I., E. R. Dougherty, et al. (2002). "Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks." Bioinformatics **18**(2): 261-74.

Simon, I., J. Barnett, et al. (2001). "Serial regulation of transcriptional regulators in the yeast cell cycle." Cell **106**(6): 697-708.

Smolen, P., D. A. Baxter, et al. (2000). "Modeling transcriptional control in gene networks--methods, recent results, and future directions." Bull Math Biol **62**(2): 247-92.

Spellman, P. T., G. Sherlock, et al. (1998). "Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization." Molecular Biology of the Cell **9**(December): 3273-3297.

Stuart, J. M., E. Segal, et al. (2003). "A gene-coexpression network for global discovery of conserved genetic modules." Science **302**(5643): 249-55.

Tegner, J., M. K. Yeung, et al. (2003). "Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling." Proc Natl Acad Sci U S A **100**(10): 5944-9.

Thijs, G., M. Lescot, et al. (2001). "A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling." Bioinformatics **17**(12): 1113-22.

Thomas, R., D. Thieffry, et al. (1995). "Dynamical behaviour of biological regulatory networks--I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state." Bull Math Biol **57**(2): 247-76.

Toh, H. and K. Horimoto (2002). "Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling." Bioinformatics **18**(2): 287-97.

Tompa, M., N. Li, et al. (2005). "Assessing computational tools for the discovery of transcription factor binding sites." Nat Biotechnol **23**(1): 137-44.

van Helden, J. (2003). "Regulatory sequence analysis tools." Nucleic Acids Res **31**(13): 3593-6.

van Helden, J. (2004). "Metrics for comparing regulatory sequences on the basis of pattern counts." Bioinformatics **20**(3): 399-406.

van Helden, J., B. Andre, et al. (1998). "Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies." J Mol Biol **281**(5): 827-42.

van Helden, J., B. Andre, et al. (2000). "A web site for the computational analysis of yeast regulatory sequences." Yeast **16**(2): 177-87.

van Helden, J., A. F. Rios, et al. (2000). "Discovering regulatory elements in non-coding sequences by analysis of spaced dyads." Nucleic Acids Res **28**(8): 1808-18.

van Someren, E. P., L. F. Wessels, et al. (2002). "Genetic network modeling." Pharmacogenomics **3**(4): 507-25.

van Someren, E. P., L. F. Wessels, et al. (2001). Genetic network models: A comparative study. Proc. of SPIE, Micro-arrays: Optical Technologies and Informatics (BIOS01).

Wagner, A. (2002). "Estimating coarse gene network structure from large-scale gene perturbation data." Genome Res **12**(2): 309-15.

Wahde, M. and J. Hertz (2000). "Coarse-grained reverse engineering of genetic regulatory networks." Biosystems **55**(1-3): 129-36.

Weaver, D. C., C. T. Workman, et al. (1999). Modeling Regulatory Networks with Weight Matrices. Pacific Symposium on Biocomputing 4, Hawaii.

Wittenberg, C. and S. I. Reed (2005). "Cell cycle-dependent transcription in yeast: promoters, transcription factors, and transcriptomes." Oncogene **24**(17): 2746-55.

Woolf, P. J. and Y. Wang (2000). "A fuzzy logic approach to analyzing gene expression data." Physiol Genomics **3**(1): 9-15.

Yang, Y. H., S. Dudoit, et al. (2002). "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation." Nucleic Acids Res **30**(4): e15.

Zou, M. and S. D. Conzen (2005). "A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data." Bioinformatics **21**(1): 71-9.

ACKNOWLEDGMENTS

I owe gratitude to many people for their help to my research, thinking, writing and personal life.

I am deeply indebted to my major professor, Dr. Julie Dickerson, for her constant support and guidance for my research through the years and for giving me freedom to satisfy my curiosity in research. I greatly appreciate her leading me into the fantastic research area of genetic network inference and systems biology. I am also indebted to my co-major professor, Dr. Eve Wurtele, for her help and advice in thesis research, especially in biology. Many thanks to my committee members Dr. Nicola Elia, Dr. Xun Gu, Dr. Yao Ma and former committee member Dr. Dan Ashlock. Their advice, time and patience are highly appreciated.

I would like to thank my friends, Michael Lawrence, Jie Li, Ling Li, Jian Gong, Lishuang Shen, Yuting Yang and many other people for the discussions and help during the years. Especially, I would like to thank Michael Lawrence for his careful checking of my thesis. I would also like to thank Josette Etzel, Adam Tomjack and other people developing FCModeler, which implements the network visualization in this research. Also, I want to thank all the members in MetNetDB research group for providing an environment to discuss and learn.

My graduate research has been made possible through the support of National Science Foundation and Information Technology Research Programs. Also, I would like to thank Virtual Reality Applications Center for providing me with a good work environment.

Finally, I owe my best gratitude to my beloved wife, Zhaomin Huang, my parents and other family members for their love that is always encouraging me.