ANALYSIS OF GENE EXPRESSION MICROARRAY TIME SERIES DATA

Ola ElBakry

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

April 2013

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Ola El Bakry**

Entitled: **Analysis of Gene Expression Microarray Time Series Data**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. P. Grogono

_____ External Examiner
Dr. E.R.M. Tillier

_____ External to Program
Dr. R. Ganesan

_____ Examiner
Dr. W.E. Lynch

_____ Examiner
Dr. W. Zhu

_____ Thesis Co-Supervisor
Dr. M.O. Ahmad

_____ Thesis Co-Supervisor
Dr. M.N.S. Swamy

Approved by _____
Dr. J.X. Zhang, Graduate Program Director

April 25, 2013

_____
Dr. Robin Drew, Dean
Faculty of Engineering & Computer Science

Abstract

ANALYSIS OF GENE EXPRESSION MICROARRAY

TIME SERIES DATA

Ola ElBakry, Ph. D.

Concordia University, 2013.

Regulatory interactions among genes and gene products are dynamic processes, and hence, modeling these processes is essential. In recent years, research efforts in the field of microarray data analysis have been constantly increasing due to the rapid growth of microarray technology, and due to the growing interest in the understanding of complex diseases. It is of vital importance to identify and characterize changes in gene expression over time. Since genes work in a cascade of networks, reconstruction of gene regulatory networks is a crucial process for a thorough understanding of the underlying biological interactions. Analysis of large scale microarray data is a challenging problem, where most of the microarray time series have only five to ten time points and the conventional time analysis techniques are not applicable.

The present study focuses on two important aspects of the microarray data analysis. The first part is concerned with the identification of the differentially expressed genes, whereas the second part with the reconstruction of the gene regulatory networks. New computational methods for time course microarray data that assist in analyzing and modeling the dynamics of the gene regulations are developed in this study.

The main challenges in the identification of differently expressed genes arise due to the availability of a very small number of replicated samples (usually two or three samples) in the face of a huge number of genes (thousands of genes). Further, most of the previous works, in this area have focused on static gene expressions, with only a limited number on methods for selecting the genes that exhibit changes with time. In the first part of this study, a general statistical method for detecting changes in microarray expression over time within a single or

multiple biological groups is presented. The method is based on repeated measures (RM) ANOVA, in which, unlike the classical F-statistic, statistical significance is determined by taking into account the time dependency of the microarray data. A correction factor for this RM F-statistic that leads to higher sensitivity as well as a high specificity is introduced. The two approaches for calculating the p-values that exist in the literature, that is, those resampling techniques of gene-wise p-values and pooled p-values, are investigated. It is shown that the pooled p-values method compared to the method of the gene-wise p-values is more powerful and computationally less expensive, and hence it is applied along with the correction factor introduced to various synthetic data sets and a real data set. The results from the synthetic data sets show that the proposed technique outperforms the state-of-the-art methods, whereas those from using the real data set are found to be consistent with the existing knowledge concerning the presence of the genes.

As for the reconstruction of gene regulatory networks, challenges, such as the relatively large number of genes compared to the small number of time points, result in an underdetermined problem. Additional constraints and information are needed to be able to capture the gene regulatory dynamics. Since gene regulatory interactions involve underlying biological processes, such as transcription and translation that take place at different time points, the consideration of different delays is a very crucial, yet a demanding problem. In the second part of this study, an approach based on pair-wise correlations and lasso that take into account the different time delays between various genes, is presented to infer gene regulatory networks. The proposed method is applied to both synthetic and real data sets. The results from the synthetic data show that the proposed approach outperforms the existing methods, and the results from the real data are found to be more consistent with the existing knowledge concerning the possible gene interactions.

The study on the identification of differentially expressed genes and the reconstruction of the gene regulatory networks, undertaken in this thesis, can be regarded to be directed towards a better understanding of the cellular dynamics.

Acknowledgement

I would like to express my sincere gratitude and appreciation to my advisors, Professor M. Omair Ahmad and Professor M.N.S. Swamy for their invaluable patience, support and encouragement. This work could not have been accomplished without their continuous guidance and support at every phase of the research. My profound gratitude goes to my parents for the most needed encouragement to finish my studies. Special thanks go to my husband for years of support. Without his sacrifice, it would have been impossible for me to complete this work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANOVA	Analysis of variance
cDNA	Complementary DNA
DD-LASSO	Delay Detection LASSO
DNA	Deoxyribonucleic acid
EA	Evolutionary algorithm
EM	Expectation-Maximization
FDR	False Discovery rate
GRN	Gene Regulatory Network
LARS	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
MOEA	Multi-objective Evolutionary Algorithm
MOP	Multi-objective Optimization Problem
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
RM ANOVA	Repeated Measures ANOVA
RNA	Ribonucleic acid
SVD	Singular Value Decomposition
TF	Transcription Factor
VSP method	Variable Shrinkage Parameter method

Chapter 1

Introduction

In DNA gene expression microarrays thousands of gene expression levels are measured simultaneously. Microarray data may provide insight into gene to gene interactions, gene function and pathway identification. Expression microarrays can be studied for static or temporal data. In a static experiment, the arrays are obtained at a single moment of gene expression. In a time series experiment the arrays are collected over a time course, allowing the study of the dynamic behavior of gene expression. Since the regulation of gene expression is a dynamic process, it is vital to identify and characterize changes in gene expression over time. In this work we are mainly interested in the time course data. The key challenges for the time series data are that the number of time points as well as the number of samples is small and the number of genes is very large.

The main microarray data analysis steps involves the identification of differentially expressed genes, gene clustering and gene regulatory network reconstruction. The identification of differentially expressed genes is to find genes whose expression changes in response to different biological conditions, which is a vital step of microarray data analysis. In this inference process, two essential steps are needed; the definition of the statistic measuring the differential expression, which enables us to rank the genes, and the assessment of the statistical significance of the results.

The aim of regulatory network reconstruction is to detect the most likely interactions by identifying sets of relevant model parameters that are required to obtain an appropriate correspondence between measured data and model output.

This work is concerned with the identification of differentially expressed genes and the network reconstruction. The gene selection is an essential primary step while the network inference gives more understanding of the underlying biological processes.

A microarray background and microarray data preprocessing background are found in the next two sections. Then, problem statement and research objectives are introduced in the following sections, followed by a brief description of the thesis organization in the last section.

1.1 Gene Expression Microarrays

The main nucleic genetic material of cells is represented by Deoxyribonucleic acid (DNA) molecules. It is a nucleic acid that contains the genetic information for the development and functioning of all living organisms. The DNA double helix molecules comprise two anti-parallel intertwined complementary strands. The genetic information in a living organism is the same in all cells. Nevertheless, according to the different types of cells and responses only some genes would be active (expressed). Expressed genes show how the cells function and the underlying biological processes. A gene is expressed when it makes a new protein. Transcriptional gene regulation is a process where the DNA of a certain gene is used as a template. This gene is translated later to a protein. The better understanding of these gene transcription activities lead to accurate understanding of the underlying cellular processes and responses. In the transcription process, hybridization occurs, where part of the DNA binds with the mRNA. The microarray technology repeats the hybridization process to know which genes are expressed.

DNA microarrays are used to measure changes in expression levels. Microarrays differ in fabrication, workings, accuracy, efficiency, and cost. A microarray is usually a slide containing large number of tiny spots consisting of probe sequences. They can be immobilized at micrometer distances, so it is possible to place many different probes on a small single surface of one square centimeter. The number of probes can reach 10,000 or more. Target RNA is generally extracted from samples of interest (e.g. cancer tumors), reverse transcribed into complementary DNA (cDNA), labeled with fluorescent dye and then hybridized to the array. There are one channel and two channel microarrays. The more common arrays are the two color arrays, where two different samples are labeled with different dyes (Cy3, green and Cy5, red), and then, hybridized simultaneously to the same slide. Two DNA strands hybridize if they are complementary to each other. One or both strands of the DNA hybrid can be replaced by RNA and hybridization will still occur as long as there is complementarity. The fluorescent intensity of a spot is equivalent to the amount of RNA expressed in the sample. The fluorescent dye can be detected by a light scanner that scans the surface of the chip for hybridized material. A summary of the microarray process for a two channel microarray is shown in Figure 1.1. Hybridization both labeled samples are mixed

purified and hybridized on the microarray base pair interactions between DNA samples(target) and DNA molecules on the microarray (probes). Usually green spots indicate only DNA from probe is fixed, while red spots mean only DNA from the experimental sample is fixed, whereas yellow spots show that DNA from both are fixed in equal amount, and grey spots appear when there is no hybridization. A sample of microarray image is shown in Figure 1.2.

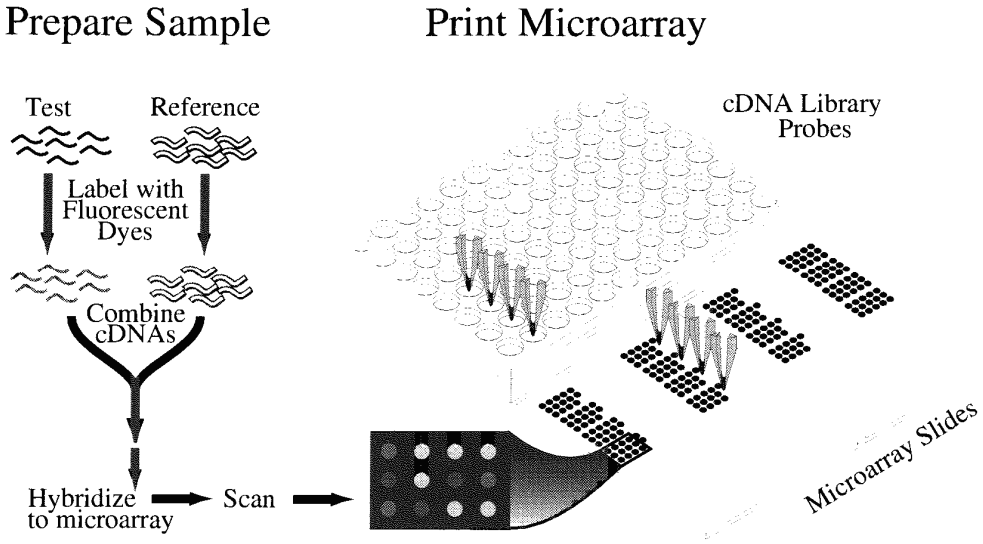

Figure 1.1 Two-channel Microarray formation process

In time series microarray data, the arrays are collected over a time course. Usually, microarray experiments are very noisy and there are lots of sources of error; hence, it is recommended to replicate the experiment several times to ensure the quality of the gene expression data. The microarray data measurements are repeated to form replicated samples. Then, the resulting microarray image is preprocessed, to get numerical values for each gene, known as gene expression data, and arranged in tables as shown in Table 1-1.

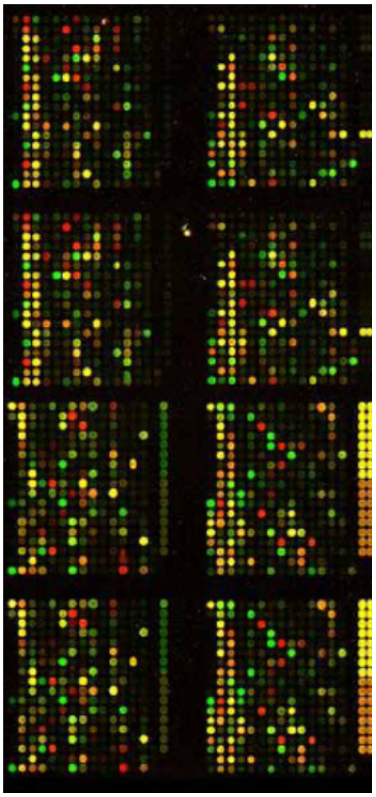

Figure 1.2 Output Microarray image

Table 1-1 Microarray data for each gene

	Time point 1	…….	Time point j	…….	Time point T
Sample 1					
…..					
Sample *i*			Yij		
…..					
Sample *n*					

1.2 Preprocessing of Microarray Data

First, the raw microarray data is preprocessed in order to minimize extraneous variations in the measured gene expression levels of hybridized mRNA samples, and the biological variations can be

more easily distinguished. An essential step of data preprocessing is to normalize the microarray data, where normalization is the process of removing systematic variation from the data. Systematic errors in DNA microarray experiments can result from unequal RNA quantities in the sample, differences in labeling and detection efficiencies. In addition to that, errors can be due to systematic biases in measured expression levels, scanner settings, laser saturation effects, print-tip variation and sample plate origin. Normalization adjusts individual intensities so that comparisons can be made both within an array and between arrays in the experiment. Adjustments are necessary to remove differences which are purely technical and do not represent true biological variation. The purpose of normalization is to adjust for effects which arise from variation in the microarray technology rather than from biological differences between the RNA samples or between the printed probes. These differences if left unadjusted will hinder the ability to identify true differentially expressed genes (i.e. detect the genes that are actually active and producing proteins) and may increase the number of false positives found. In order to remove the bias artifacts, sophisticated methods have to be applied. If the imbalance is more complicated than a simple scaling of one channel relative to the other, as it usually will be, then the dye bias is a function of intensity and normalization will need to be intensity dependent. The dye-bias will also generally vary with spatial position on the slide. Positions on a slide may differ because of differences between the 2 print-tips on the array printer, variation over the course of the print-run, non-uniformity in the hybridization or from artifacts on the surface of the array which affect one color more than the other. Finally, differences between arrays may arise from differences in print quality, from differences in ambient conditions when the plates were processed or simply from changes in the scanner settings. There are many other trends which could be estimated and adjusted for in the normalization step, although normally these are of less importance than the intensity and spatial trends already considered. For example, there can be differences between the purity of DNA from different amplification batches or from different clone libraries. This can mean that different spots on the microarray contain different effective quantities of DNA. Normalization not only corrects for different dye properties but also for concentration differences between the co-hybridized test and reference samples. The locally weighted linear regression (LOWESS) normalization and the

quantile normalization are able to correct intensity-dependent effects. After the normalization and scaling steps, microarray analysis can be applied to extract meaningful information from this microarray data.

1.3 More Recent Technologies for Gene Expression Data

In this thesis, we carry out the analysis of time-series gene expressions that are extracted from microarray data, since currently hybridization-based microarrays are the primary method for global gene expression analysis and microarray databases are easily available. The same analysis methods and techniques can be safely applied to time-series gene expression data that is generated using more recent technologies such as more advanced probe-based methods (e.g. Nanostring) [1] and RNA sequencing (RNA-Seq) [2] methods.

The Nanostring technology [1] is a variation of the DNA microarray where it uses molecular barcodes and microscopic imaging to detect and count up to several hundred unique transcripts in one hybridization reaction. Each color-coded barcode is attached to a single target-specific probe corresponding to a gene of interest. This technology employs two ~ 50 base probes per mRNA that hybridize in solution. The Reporter Probe carries the signal; the Capture Probe allows the complex to be immobilized for data collection. After hybridization, the excess probes are removed and the probe/target complexes aligned and immobilized in the nCounter Cartridge. Sample Cartridges are placed in the Digital Analyzer for data collection. Color codes on the surface of the cartridge are counted and tabulated for each target molecule.

RNA-Seq [2] has the potential to replace microarrays in transcriptome analysis due to its advantages in sensitivity, quantification, and replicability of experiments. RNA-Seq allows the sequencing of the entire transcriptome, and thus permits both transcript discovery and robust digital quantitative analysis of gene expression levels. This method relies on the generation of short reads of transcript sequence information which are then assembled into full-length transcripts and mapped to the genome. To generate this data, isolated RNA populations (e.g. small RNAs) are converted to cDNA for sequencing More recently developed methods involve direct sequencing of RNA to avoid artifacts generated by reverse transcription and subsequent modification steps The number of reads

for a given transcript (calculated in reads per kilobase of exon model per million reads, or RPKM) corresponds to the absolute expression level of that particular gene in the cell type or tissue in question, providing an absolute quantification method with large dynamic ranges in contrast to the relatively limited dynamic range of microarrays that depends on relative, rather than absolute, quantification of hybridization intensities.

1.4 Motivation

The understanding of the gene interactions that contribute to certain diseases provides a potential therapeutic strategy. A comprehensive investigation of the microarray data is a possible way to get such a detailed understanding. Identifying which genes are differentially expressed in treated samples followed by modeling these differentially expressed genes provide deep insight into the biological interactions and processes.

There are a limited number of methods that have been proposed for selecting the genes that exhibit changes with time. There are four main approaches that have been proposed to solve this problem. Peddada *et al.* [3] have identified genes by comparing each of the gene expressions with predefined candidate profiles. Hence, the larger the number of time points, the larger the set of predefined profiles that need to be used. Storey *et al.* [4] have determined significant genes by performing on each gene a hypothesis test to determine as to whether its population-average versus time curve is flat. Hence, any significant change at a single time point is missed, and only significant change at continuous time changes can be identified. Tai *et al.* [5] have used an empirical Bayes method to identify highly-expressed genes. However, this method does not provide explicit p-values or q-values for the genes, but it only ranks the genes according to their significance. Angelini *et al.* [6] have proposed a Bayesian approach in which each gene expression profile is estimated globally by expanding it over an orthogonal basis. Nevertheless, some model parameters need to be defined such as the degree of the polynomial and the maximal possible degree. The existing methods for identifying significant genes changing with time, still needs accurate study and improvements. Hence, there is a need to identify significant genes while avoiding previous drawbacks, such as setting prior model parameters, to identify significant genes

irrespective of the type of change with time.

Dependency networks such as that in [7-9] that are based solely on correlations have several shortcomings. The resulting networks are undirected graphs that do not provide sufficient information regarding the relationships between the various genes. In addition, whenever any two genes are correlated to a third gene, the first two genes will be falsely connected, thus resulting in triangular clusters of genes that do not represent the real topology of GRNs. This is a major drawback of such dependency networks. Further, such networks do not take into consideration the delays between various genes, which is an inherent property of GRNs. Since GRN is an abstract network, where there is underlying chemical reactions and biological processes, time delays between stimulus and response exist that should not be neglected. In fact, physical interactions between genes are mediated through other components such as DNA, RNA, proteins, and metabolites, and gene networks are system-level descriptions of cellular physiology. Hence, incorporating delays in the GRN model is an essential part for successful modeling. An approach for GRN reconstruction that takes into account the time delays is one where the delays are represented by a system of equations, such as in [10-15].

The previous works in [7-9] consider the relations between genes without any delay. On the other hand, approaches such as in [10, 11, 15] incorporate a fixed time delay in their model. Li *et al.*[12] have developed a GRN with variable time delays. They use a decision tree to discover the time-delayed regulations between the underlying genes. Hence, they need additional datasets for training before they can apply their method successfully. Lozano *et al.*[13] have used a group lasso penalty in order to obtain a Granger graphical model. The group lasso penalty considers all the different time lags and indicates X to be Granger-causal for Y, if it has a significant effect. Since the average effect of all time lags is studied as one feature, the actual time difference between the activation of X and its effect on Y is still unknown. In addition, due to the averaging effect, it is not possible to determine the actual effect of X on Y, as to whether it is positive or negative. Shojaie *et al.*[14] have proposed a truncating lasso penalty for the estimation of graphical Granger models. The truncating effect of the proposed penalty is motivated by the rationale that the number of effects (edges) in the graphical model decreases as the time lag increases. Consequently, if the number of

edges is less than a predefined number at time *t*, all the later estimates are forced to be zero. They apply a stopping condition to stop adding more delays in the model to provide an estimate of the order of the underlying model. However, in order to do so, they require a large number of samples, and in addition, they completely ignore all the samples of further time points.

A major drawback of all the above mentioned approaches is that they are not able to model the variable time lags between any two genes, without the need for a large number of data sets or samples. When a gene regulates another gene, there is a delay before the response of the second gene appears. This delay is attributed to the underlying biological processes, such as transcription and translation that are taking place. The main challenge in modeling such time delays arises from the fact that the amount of delay is unknown between the various genes.

1.5 Objectives of the thesis

The overall objective of this study is to have a better understanding of the various biological processes using microarray data. This is achieved by addressing the following two key problems. What are the genes whose gene expressions change with time? How do these genes interact? Answers to these questions are attempted in two parts. First, new techniques are developed to infer the differential gene expressions over time. The main challenge in this part is the large number of genes whereas the number of samples is small. Second, gene regulatory network model need to be reconstructed from the gens selected from the first part. Successful gene reconstruction will yield a dynamic model that would describe the biological interactions and dynamics for various conditions. The key limiting factor is the very limited number of time points and samples compared to the number of genes composing the network. Since the number of model parameters is large compared to the available measurement data, the system is usually underdetermined. In general, without constraints, there are multiple solutions and the system of equations is not uniquely identifiable from the microarray data. It is required to obtain an appropriate system despite the non-identifiable parameter values. Thus, the identification of model structure and model parameters requires constraints representing prior knowledge, simplifications or approximations. Expression level of genes in a given cell can be influenced by a pathological status, a pharmacological or medical

treatment. The response to a given stimulus is usually different for different genes and depends on time.

Thus, the main objectives of the present work are to detect differential gene expressions over time, and to infer a detailed GRN structure, using time-series microarray data, that detects the most likely gene interactions taking into account the possible delays between different genes and to distinguish between the direct and indirect relationships. Successful gene reconstruction will provide valuable information for the pharmaceutical and biotechnology industries to design new drugs for complex diseases.

1.6 Thesis Organization

The organization of the thesis is as follows. An overview of the statistical background and the current literature for the identification of differentially expressed genes, and the gene regulatory network reconstruction are provided in Chapter 2. In Chapter 3, a detailed description of the proposed methodologies for the identification of differentially expressed genes is presented. Experimental results concerning the performance of the proposed methodologies for the identification of differentially expressed genes are given in Chapter 4. Then, the proposed approach for network reconstruction is described in Chapter 5. The experimental results concerning the performance of the approach for network reconstruction are illustrated in Chapter 6. The conclusions are summarized in Chapter 7. Finally, the R code of the proposed methods are found in the appendix.

Chapter 2

Literature Review

First, statistical background is introduced in the first section, followed by the previous work for the identification of the differentially expressed genes and the network reconstruction are found in the following two sections. Then, a brief summary is presented in the last section.

2.1 Statistical Background

2.1.1 Hypothesis Testing

In order to identify differentially expressed genes, hypothesis testing is applied. In a hypothesis test, there is an initial research hypothesis of which the truth is unknown. Then, the first step is to state the relevant null, H_0, and alternative hypotheses. Afterwards, decide which test is appropriate, and state the relevant test statistic. Subsequently, the distribution of the test statistic under the null hypothesis is either derived from the assumptions, or the test statistic follows a standard distribution, such as the Student's t distribution or normal distribution. Then, from the observations the observed value t_{obs} of the test statistic T is computed. Select a significance level (α), a probability threshold below which the null hypothesis will be rejected, while common values of α are 5% and 1%. According to the distribution of the test statistic under the null hypothesis, a probability of the observation under the null hypothesis (the p-value) is calculated. The decision rule is to reject the null hypothesis if and only if the p-value is less than the significance level (the selected probability) threshold.

P-value is a measure of the evidence against the null hypothesis in a statistical test. It is the probability of the occurrence of a test statistic equal to, or more extreme than, the observed value under the assumption that the null hypothesis is true. As in any other statistical test, the decision is made by comparing the reference value of the test statistic (t) to the reference distribution obtained under H_0. If the reference value of t is typical of the values obtained under the null hypothesis, H_0 cannot be rejected; if it is unusual, being too extreme to be considered a likely result under H_0, H_0 is rejected and the alternative hypothesis is considered to be a more likely explanation of the data. The

distribution of the test statistic under the null hypothesis can be derived using resampling techniques.

2.1.2 Resampling Techniques

Resampling is a nonparametric method of statistical inference, that does not involve the utilization of the standard distribution tables (for example, normal distribution tables) in order to compute approximate probability values. It is used as a robust alternative to inference based on parametric assumptions when those assumptions are in doubt, or where parametric inference is impossible or requires very complicated formulas for the calculation of standard errors. These techniques include the bootstrapping as well as the permutation significance tests.

Bootstrapping is a statistical method for estimating the sampling distribution of an estimator by sampling with replacement from the original sample in such a manner that each number of the sample drawn has a number of cases that are similar to the original data sample. Due to replacement, the drawn number of samples that are used by the method of Resampling consists of repetitive cases. It can be used for constructing hypothesis tests. A permutation test is a statistical significance test in which a reference distribution is obtained by calculating all possible values of the test statistic under rearrangements of the samples. If the samples are exchangeable under the null hypothesis, then the resulting tests yield exact significance levels. Confidence intervals can then be derived from these tests. The leading assumption is that it is possible that all of the treatment groups are equivalent, and that every member of them is the same before sampling began. From this, one can calculate a statistic and then see to what extent this statistic is special by seeing how likely it would be if the treatment assignments had been rearranged. Permutation tests exist for any test statistic, regardless of whether or not its distribution is known. The argument invoked to construct a null distribution for the statistic is that, if the null hypothesis is true, all possible pairings of the two variables are equally likely to occur. The pairing found in the observed data is just one of the possible, equally likely pairings, so that the value of the test statistic for the unpermuted data should be typical, i.e. located in the central part of the permutation distribution.

The prime difference between the bootstrap and permutation tests is that Bootstrap is associated

with sampling with replacement while for permutation test the sampling is done without replacement. Moreover, Permutations test hypotheses is concerned with the distributions while bootstraps test hypotheses is concerned with the parameters.

2.2 Identification of Differentially Expressed Genes Literature Review

Early work for identifying differentially expressed gens was done by using a fixed threshold value, such as using a two-fold increase. However, this is statistically inadequate. There are large number of random biological variations that can occur during a microarray experiment, such as sample-to-sample differences and physiological variations. A more appropriate approach for the identification of differentially expressed genes includes calculation of a statistic based on replicate array data for ranking genes according to their possibilities of differential expression and selection of a cut-off value for rejecting the null-hypothesis that the gene is not differentially expressed.

2.2.1 Time Series Identification of Differentially Expressed Genes Review

Some of the previous work was originally done for static gene expressions (i.e. gene expressions are measured at a single time point) such as that of Tusher *et al.* [16] and was subsequently extended to time-course expressions in SAM (http://www-stat.stanford.edu/~tibs/SAM). However, since the main research work was carried out on static data there is no statistical validation for the genes identified. Some of the other research groups have focused on identifying differential genes for time series expressions among different classes. For instance, Park *et al.* [17] have proposed a statistical test procedure based on the ANOVA model where the effect of time is first removed and then the residuals are used.

There are a limited number of methods that have been proposed for selecting the genes that exhibit changes with time. There are four main approaches that have been proposed to solve this problem.

Peddada *et al.* [3] have identified genes by comparing each of the gene expressions with predefined candidate profiles. The candidate profiles are expressed in terms of the inequalities between the expected expression levels at different time points. Hence, the larger the number of

time points, the larger the set of predefined profiles that need to be used. The best fitting profile for a given gene is selected based on the goodness-of-fit criterion and the bootstrap test. A bootstrap test procedure is conducted for each gene independent of the other genes. This algorithm could be useful for classification purposes. If the differential genes are already known, this test can be used to easily identify different profiles. Storey *et al.* [4] have determined significant genes by performing on each gene a hypothesis test to determine as to whether its population-average versus time curve is flat. A statistic analogous to the t and F statistics has been defined. Two models, one based on the approximation of the population-average versus time curve by a polynomial and the other by a natural cubic spline have been proposed. The model fitting procedure for longitudinal sampling is much more complicated since it takes into account the dependency of the measurements for a given subject. A false discovery rate criterion is then applied and the q-values for the genes estimated. Any significant change at a single time point is missed, and only significant change at continuous time changes can be identified.

Tai *et al.* [5] have used an empirical Bayes method to identify highly-expressed genes. They have derived the corresponding statistics for both the one-sample and two-sample problems, where in the former, the null hypothesis is that the expected temporal profile is constant, while that in the latter, the two expected temporal profiles are the same. However, this method does not provide explicit p-values or q-values for the genes, but it only ranks the genes according to their significance.

Angelini *et al.* [6] have proposed a Bayesian approach in which each gene expression profile is estimated globally by expanding it over an orthogonal basis. Each gene expression profile is presented by a short vector of coefficients and Bayesian approach delivers the posterior distribution of this vector. The method can accommodate various types of error distributions such as the normal, Student T and double-exponential. Since all the computations are performed analytically, the application of resampling methods is avoided. Nevertheless, some model parameters need to be defined such as the degree of the polynomial and the maximal possible degree. Their model can be useful for generating simulation data and is available in their software BATS [18].

Generally, the statistical inference consists of two main parts; definition of the quantity measuring

differential expression namely, the statistic, and the assessment of the statistical significance of the results. In the microarray data, due to the small number of samples, the statistic may need moderation. Moderation is well-studied in the microarray literature as shown in the next subsection.

2.2.2 Variance Moderation Review

Baldi and Long [19] have implemented an empirical Bayes approach, where population variances were estimated by a weighted mixture of the sample variance and an overall factor selected using expression values from all the data. The moderated t-test replaces the usual variance estimate with a Bayesian estimator based on a hierarchical prior distribution. Efron *et al.* [20] have added a factor to the denominator of the statistic. This additional factor is the same for all the genes and is commonly chosen from the set of pooled standard deviations. They have chosen the factor as a quantile of the standard deviation values of all the genes. Tusher *et al.* [16] have implemented a procedure to choose the factor automatically. They have estimated the factor among the percentiles of the standard errors by minimizing a coefficient of variation. The coefficient of variation of the median absolute deviation of the test statistic is computed over a number of percentiles. Broberg [21] has proposed a computationally intensive method to determine the added factor by minimizing a combination of estimated false positive and false negative rates over a grid of significance levels and factors. Smyth [22] has used an empirical Bayesian technique, where small variances are raised and large variances shrunken towards a common value. Cui *et al.* [23] have proposed a shrinkage estimator for gene-specific variance components based on the James–Stein estimator and have used it to construct a test statistic. The shrinkage estimator makes a priori assumptions about the distribution of the variance components. Wright *et al.* [24] have proposed a model, where the within gene variances are drawn from an inverse gamma distribution, whose parameters are estimated across all genes. Most of the proposed correction algorithms are defined and applied for the t-test only. Smyth [22], Cui [23] and Wright *et al.* [24] have proposed extensions to their algorithm to the multi-groups testing and the F test. Nonetheless, they have distribution assumptions for the residual error.

To find the significance, there are two approaches for computing the p-values by considering

either the permutations of the whole set of genes (pooled) [4, 25] or the permutations of each gene independently (gene-wise) [3, 17, 26].

After identifying the significant genes, the next step is to understand the interactions between these genes through network reconstruction.

2.3 Network Reconstruction Literature Review

The observed changes in gene expression over time are either due to direct effects of the stimulus on specific genes or result from secondary gene to gene interactions. Genes are working in a cascade of networks; hence, there is growing interest in the use of expression data to construct biological networks. GRNs provide an understanding of the genetic architecture of complex diseases, and thus, assist in developing new therapeutic solutions. The goal of network inference is to detect the most likely interactions by identifying sets of relevant model parameters. Intensity values of samples are usually averaged to reduce the complexity of the data set. There are different network model architectures that can be employed to reconstruct the gene regulatory network (GRN). The model architecture is a parameterized mathematical function that describes the general behavior of a target component based on the activity of regulatory components. Once the model architecture has been defined, the network structure (i.e. the interactions between the components) and the model parameters (e.g. type/strengths of these interactions) need to be learned from the data. Over the last years, a number of different model architectures from gene expression data have been proposed. In general, the network nodes represent compounds of interest, e.g. genes or modules (sets of compounds). Model architectures can be distinguished by the representation of the activity level of the network components. Since both network structure and parameters are unknown, statistical approaches such as graphical models and linear systems are used to estimate the genetic networks. The concentration or activity of a compound can be represented by Boolean or other logic values, discrete, fuzzy or continuous values. Furthermore, network model architectures can be distinguished by the type of model (stochastic or deterministic, static or dynamic) and the type of relationships between the variables (directed or undirected; linear or non-linear function or relation table). The major network classes are the system of equations, Boolean

network, Bayesian network and the information theory architectures. The different network architectures are shown in Figure 2.1.

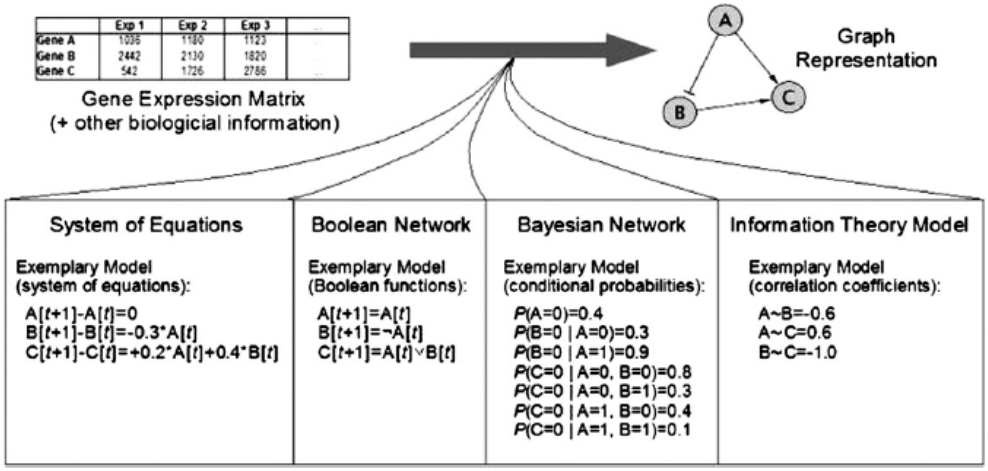

Figure 2.1 Network architectures

A drawback of the information theory models is that they are static, while that of the Boolean networks is that the gene expressions cannot be described adequately by only two states. The two models that best describe the dynamics are that the system of equation models and the Bayesian networks. The system of equations model has different aspects. The system of equation can be continuous (differential equations) or discrete (difference equations). Furthermore, the equations could be representing linear system or non-linear one. In addition to that, the system can be deterministic or stochastic taking into account the random variability of the gene expressions.

2.3.1 Information Theory Models and Measures of Association

If the network structure is unknown, statistical approaches such as graphical models are used to estimate genetic networks. It is mainly concerned with constructing Dependency Graphs between different genes. These graphs should reveal the distinction between direct and indirect interactions of various genes, thereby inferring the underlying network topology. Correlations are widely used to infer the structure of the GRN. In this regard, Opgen-rhein *et al.* [7] have used a functional approach to find the dynamic correlations between various genes. They have considered the

observed gene expressions over time as realizations of random curves, rather than considering the individual time points separately. They have approximated the temporal expression of the genes using linear splines. Their approach has been based on a dynamic pair-wise correlation estimator which provides a similarity score for pairs of groups of randomly sampled curves. They compute the partial dynamic correlations matrix directly from the inverse of the correlation matrix. De la Fuente *et al.* [8] have proposed a method to construct approximate dependency graphs from large-scale biochemical data using partial correlation coefficients. The partial correlations of first and second order are computed using iterative methods. The correlation between two variables is evaluated by conditioning on all possible pairs of other variables. If any of these pairs yields a zero partial correlation the corresponding edge is removed from the correlation network. This is executed over all possible edges results in a network of direct interactions. The conditioning on any common causal descendent introduces a correlation between two variables that are independent conditional on their causal ancestors. Therefore, conditioning on all variables simultaneously can introduce some dependencies, which are not due to direct causal effects or common ancestors. Wille *et al.* [9] have used only the first order partial correlations, where they have applied graphical modeling to sub-networks of three genes to study the dependence between two genes conditional on the third one. Then, the sub-networks have been combined for the inference of the complete network. Dependency networks that are based solely on correlations have several shortcomings. The resulting networks are undirected graphs that do not provide sufficient information regarding the relationships between the various genes. In addition, whenever any two genes are correlated to a third gene, the first two genes will be falsely connected, thus, resulting in triangular clusters of genes that do not represent the real topology of GRNs. This is a major drawback of such dependency networks. Further, such networks do not take into consideration the delays between various genes, which is an inherent property of GRNs. Since GRN is an abstract network, where there is underlying chemical reactions and biological processes, time delays between stimulus and response exist and should not be neglected. In fact, physical interactions between genes are mediated through other components such as DNA, RNA, proteins, and metabolites, and gene networks are system-level descriptions of cellular physiology. Hence, incorporating delays in the

GRN model is an essential part for successful modeling. An approach for GRN reconstruction that takes into account the time delays is one where the delays are represented by a system of equations.

2.3.2 System of Equations

There are two main approaches for modeling the network with linear System of Equations. One approach depends on reducing the problem of dimensionality while the other uses the global set of genes directly. In the first approach, the problem of large number of genes can be reduced first by applying gene clustering. The network is reconstructed between clusters, not each gene. Cluster-representative genes are used for the modeling. It could be assumed that genes which have similar expression patterns also have the same regulators. Another way for solving the underdetermined problem is to reduce the dimensionality of data using techniques such as the Principal Component Analysis (PCA) or the Singular Value Decomposition (SVD).

For instance, Guthke *et al.* [11] have used gene clustering combined with a heuristic search strategy for finding optimized network reconstruction. A modified fuzzy C-means algorithm has been employed for clustering. Afterwards, the network reconstruction has been composed of two parts the model structure and the model parameters. Prior knowledge concerning the connectivity between genes has been exploited to restrict the search space for the model structures. The model structure has been decomposed into smaller sub-models. The sub-model estimation starts with an initial sub-model that represents a first order lag element. The sub-model of each gene possesses two non-zero parameters; the parameter that realizes the self-regulation effect and the parameter that describes the influence of the external stimulus on the expression of the gene. Two directions of search have been applied; forward selection and backward elimination. For each model structure the model parameters have been fitted to the gene expression data using standard optimization techniques. Then, the mean square error between the model output and the data has been determined and used to assess the model structure. In order to find initial parameter values for the iterative optimization procedure, time derivatives have been used. They are calculated based on an interpolation between the data points. A drawback of the interaction networks between nodes of representative gene clusters is that the resulting network is an abstract network between gene

clusters. On the other hand, the gene regulatory networks of single genes give more insight into the various biological processes.

Bansal *et al.* [10] have inferred the local network of gene to gene interactions surrounding a gene, or genes, of interest by perturbing only one of the genes in the network and measuring the gene expression profiles at multiple time points. To solve the underdetermined system problem several steps have been applied. First, they have applied a cubic smoothing spline filter with an adjustable smoothing parameter. The purpose of the smoothing is to reduce the noise. Afterwards, they have used interpolation to increase the number of time points using piecewise cubic spline interpolation. Finally, PCA has been applied to the dataset in order to reduce its dimensionality and solve the equation in the reduced dimension space. It works on small size networks. Nonetheless, for very large number of genes its performance deteriorates.

Holter *et al.* [27] have used Singular value decomposition (SVD) to solve the linear equation system. The time evolution of gene expression levels has been described by using a time translational matrix to predict future expression levels of genes based on their expression levels at some initial time. The time translational matrix has been deduced by modeling them by using the characteristic modes obtained by singular value decomposition. The expression data for each gene is viewed as a unit vector in a hyperspace, each of whose axes represents the expression level at a measurement time of the experiment. The SVD construction ensures that the modes correspond to linearly independent basis vectors. A linear combination of these modes describes the expression pattern of each gene. The resulting time translation matrix has provided a measure of the relationships among the modes and governs their time evolution. They have showed that a truncated matrix linking just a few modes is a good approximation of the full time translation matrix. To solve the inverse problem and infer the nature of the gene network connectivity, the causal relationships among the characteristic modes obtained by SVD have been considered.

The second approach is to reverse-engineer the global genetic pathways without using data reduction techniques. Van Someren *et al.* [15] have developed an algorithm which is based on the Least Absolute Shrinkage and Selection Operator (lasso) technique. They have utilized the literature

enrichment scores to find parameter concerning the lasso technique. This enabled them to select a single network solution. Lasso [28] is an algorithm that shrinks the least absolute weights such that only a few weights remain non-zero. The linear model assumes that the gene expression level of each gene is the result of a weighted sum of all other gene expression levels at the previous time point. In order to obtain an estimate of the complete set of model parameters from data, usually the squared error between the predicted and measured gene expression levels is minimized. In lasso technique, the standard squared error with a penalty term that sums the absolute values of the weights is obtained. A parameter is multiplied by the penalty term. It provides a trade-off between data-fit term and the penalty term. The mathematical details of lasso and its implementation can be found in [28, 29]. The properties and performance of lasso have been studied extensively and some improvements have been introduced. One of the most commonly-used modifications is that due to Zou [30]. He proposed an adaptive lasso penalty term which is weighted according to initial estimates and he has shown that if suitable weights are used, the adaptive lasso can achieve variable selection consistency.

All the previous work of [7], [8], and [9] consider the relations between genes without any delay. On the other hand, previous approaches, such as [10], [11]and [15], incorporate time delay in their model, however, they assume that all the genes are affected by other genes with a fixed delay. Li *et al.* [12] have developed a GRN with variable time delays. They have used a decision tree to discover the time-delayed regulations between the underlying genes. Hence, they need additional datasets for training before applying their method successfully to the problem of interest. Lozano *et al.* [13] have used a group lasso penalty in order to obtain a Granger graphical model. The group lasso penalty considers all the different time lags and indicates X to be Granger-causal for Y if the average effect is significant. Since the average effect of all time lags is studied as one feature, the actual time difference between activation of X and its effect on Y is still unknown. In addition, due to the averaging effect, it is not possible to determine the actual effect of X on Y as to whether it is positive or negative. Shojaie *et al.*[14] have proposed a truncating lasso penalty for the estimation of graphical Granger models. The truncating effect of the proposed penalty is motivated by the rationale that the number of effects (edges) in the graphical model decreases as the time lag

increases. Consequently, if there are less than a predefined number of edges at time at *t*, all the later estimates are forced to zero. They apply a stopping condition upon which they stop adding more delays in the model to provide an estimate of the order of the underlying model. However, in order to do so, they require a large number of samples, and in addition, they completely ignore all the samples of further time points.

2.4 Summary

As can be seen from the above literature review, the knowledge and understanding of the biological pathways are far from being complete. Clearly, a comprehensive investigation of the microarray data is a possible way to get detailed understanding. Identifying which genes are differentially expressed in treated samples is the first step to improve the biological understanding. As shown from the first section of the literature review, the existing methods for identifying significant genes changing with time, still needs accurate study and improvements.

The identified genes are used to reconstruct gene regulatory network for further understanding of the underlying dynamic processes. The GRN reconstruction is one of the major challenges in systems biology. A review of relevant literature studies demonstrates that the existing algorithms are of limited accuracy. A main drawback of all the previous approaches is that they are not able to model the variable time lags between any two genes, without the need for a huge number of data sets or samples. When a gene regulates another gene, there is a delay before the response of the second gene appears. This delay is attributed to the underlying biological processes taking place such as transcription and translation. The main challenge in these time delays is that the amount of delay is unknown between the various genes. There is a necessity to develop new techniques in these relatively new areas where studies devoted to these topics remain insufficient.

Chapter 3

Identification of Differentially Expressed Genes

In this work we are interested in the time-series data analysis. The first question as to which genes change their expressions is solved by identifying the differentially expressed genes [31]. For the first part, RM ANOVA will be used to get the statistic. Permutations are applied to get the p-values, and hence, determine the significance of the statistic. A new moderated statistic is introduced. For multiple time course data, a mixed design ANOVA is employed to compute the statistic, followed by a procedure similar to that of single time series.

3.1 Identifying Differentially Expressed Genes for a Single Time-course Data

In this section, an algorithm applicable to longitudinal time-series with samples is proposed for selecting genes according to their time-course profiles using gene expression data. The statistical inference consists of two main parts; definition of the quantity measuring differential expression, and assessing statistical significance of the results. In the proposed algorithm RM ANOVA will be used to get the statistic. To find its significance, permutations [32] are used to get the p-values. The methods of both the pooled p-values and the gene-wise p-values are used to evaluate the RM F significance. For the gene-wise p-values a new coarse-to-fine strategy is introduced to reduce the number of the required permutations. A new moderation factor is introduced and applied to the RM F-statistic.

3.1.1 RM ANOVA

Generally, ANOVA tests the null hypothesis of no differences between population means. One of the assumptions of ANOVA is the independence of the groups being compared. This is not true for longitudinal time-series data. Using a standard ANOVA in this case is not appropriate since it fails to model the correlation between the repeated measures. RM ANOVA takes into account these dependencies. The difference between the RM and the independent-measures (IM) ANOVA is that the former removes the variance caused by individual differences. Hypotheses for both the IM ANOVA and RM ANOVA are the same and test for the equality of the means. The mathematical

details of RM ANOVA can be found in [33].

The RM ANOVA may be thought of as a model designed to assess treatment differences while controlling the between-sample variability, when each gene expression value is measured a few consecutive times. The model is simple to interpret and takes into account the various aspects of the repeated-measurements data. The RM ANOVA model for each gene is given by [33]

$$Y_{ij} = \eta + \mu_j + \alpha_i + (\alpha\mu)_{ij} + \varepsilon_{ij} \quad (3\text{-}1)$$

where Y_{ij} is the microarray value for the ith sample at the jth time point, η is the population grand mean under all fixed ratios, μ_j is the fixed effect of the time j, α_i is the effect of the ith sample, $\alpha\mu_{ij}$ is the interaction effect and ε_{ij} is the random error of the ith sample at the jth time point. The RM ANOVA model is similar to that of the mixed-effect where the replicated samples are the random variables and the time points are the fixed effect variable. The sample effects, α_i's, are assumed to be independent of one another and the errors, ε_{ij}'s, are assumed to be independent of the other effects and of each other. The interaction term $(\alpha\mu)_{ij}$ of each sample is assumed to be independent of the interaction terms of the other samples, but can be dependent for the same subject. The null hypothesis, H_0, states that the effect of time is constant across all time points, while the alternative hypothesis, H_1, states that there is a change across time. The hypothesis can be summarized as follows:

H_0: $\mu_1 = \mu_2 = \mu_3 = \ldots = \mu_T$

H_1: H_0 is false

In addition to the previous assumptions, in order to use the F-tables directly, it is assumed that the effect terms, the error term and the interaction term are normally distributed and that the variances of the differences for all time point combinations are homogeneous, which is known as sphericity. However, the normality assumption is not usually satisfied in the microarray data, and hence, instead of using the F-tables, permutation procedure is employed. In the null case, the correlation attributed to any changes with respect to time does not exist. Consequently, in addition to the mean, the variances and the pair-wise correlations between different measurements, for the same subject,

in the null case, are equal. This interpretation for the null case is sufficient for the validity of the application of the permutation procedure which is previously applied in [3, 17].

For each gene, data is arranged in a table with T-columns and *n* rows as shown in Table 3-1. The columns indicate the time points and rows the replicated samples. In this table, Y_{ij} is the microarray value for the *i*th sample at the *j*th time point.

Table 3-1 Data arrangment for each gene

	Time point1	……	Time point *j*	…….	Time point *T*	
Sample 1						
…..						
Sample *i*			Y_{ij}			$\overline{Y_i}$
…..						
Sample *n*						
			$\overline{Y_j}$			$\overline{Y_{Tot}}$

The quantities shown in Table 3-1 are defined as follows

$$\overline{Y_i}=\frac{1}{T}\sum_{j=1}^{T}Y_{ij}\ ,\ \overline{Y_j}=\frac{1}{n}\sum_{i=1}^{n}Y_{ij}\ ,\ \overline{Y_{Tot}}=\frac{1}{T}\sum_{j=1}^{T}\overline{Y}_j=\frac{1}{n}\sum_{i=1}^{n}\overline{Y}_i \tag{3-2}$$

The means $\overline{Y_i}$, $\overline{Y_j}$ and $\overline{Y_T}$ are used to compute the following quantities:

$$SS_{Bt}=n\sum_{j=1}^{T}(\overline{Y_j}-\overline{Y_{Tot}})^2\ ,\ SS_{BS}=T\sum_{i=1}^{n}(\overline{Y_i}-\overline{Y_{Tot}})^2\ ,\ SS_T=\sum_{i=1}^{n}\sum_{j=1}^{T}(Y_{ij}-\overline{Y_{Tot}})^2 \tag{3-3}$$

$$SS_{re}=SS_T-SS_{BS}-SS_{Bt} \tag{3-4}$$

where SS_{Bt} is the sum of squares of the treatment levels, SS_{BS} is the sum of squares of the subjects, SST is the total sum of squares and SS_{re} is the residual (error) sum of squares, which is a measure of the total discrepancy between a model and the observed data. These quantities are used to calculate the RM F-statistic using

$$F = (n-1)\frac{SS_{Bt}}{SS_{re}} \tag{3-5}$$

After computing the RM F-statistic, it is required to estimate the significance of these values for which the standard F-tables are generally used. Nonetheless, there are several assumptions required to use these F-tables. The main assumptions are that the observations among the individuals should be independent and should have normal distribution and variance homogeneity (sphericity). In case the sphericity assumption is violated, there are several methods to adjust the numbers of degrees of freedom, for example that of Geisser and Greenhouse [34]. However, in the present work the main assumption of normal distribution cannot be verified, and hence, the permutation based p-values are computed and used in the determination of the significance rather than using the standard F-tables. The permutation procedure is a valid approach for the time-series microarray data, where the null hypothesis tests if there is no effect of the time variable on any of the microarray data measurements, and that the measurements under the null hypothesis are assumed to be identically distributed.

3.2 F-statistic Moderation

As mentioned earlier, a problem with the F-test for microarray data is the small number of samples. A small population of size n = 2 or 3 is very common and may lead to substantial underestimation or overestimation of the variances. Thus, a variance correction is needed. The role of the correction factor is to prevent genes whose expression is near zero from having large scores. There are genes that have very small expression values and hence are insignificant. The numerator of the F-statistic of such a gene, given by (3-5), is very small, and so is its denominator; hence the resulting F-statistic could be large. These genes will be falsely identified as significant (false positives). To prevent genes with large F-statistics but small numerators from being selected, a factor is added to the denominator of each of the F-statistic. By adding a factor to the denominator, it is prevented from being too small. On the contrary, there are genes that are significant whose F-statistics have very large denominators, and hence the resulting F-statistics would be small. These genes will be falsely identified as insignificant (false negatives). To prevent such genes from being

missed, our moderation factor shrinks the denominator towards the median of the variances of all the genes. The correction term serves as a control for both the underestimation and overestimation of the variance, and can suppress both false positives and false negatives. This correction factor for each gene is estimated by carefully combining the information from the expression values of the other genes. In our proposed moderation scheme, unlike most of the previous correction methods that apply a single correction to all the genes, the correction factor depends on the denominator of each F-statistic for each gene.

In general, if a single correction is applied to all the genes, then the adjusted denominator for F is given by [16, 22]

$$\tilde{S}_i = (1-\lambda)S_i+\lambda t \tag{3-6}$$

where S_i is the denominator of the F-statistic for each gene i and t is the target to which the denominator is shrunk. However, since not all the genes are overestimated nor all underestimated, applying a single correction to all the genes is inadequate. A more precise correction is needed, where the correction factor is applied to each group of genes according to their degree of underestimation or overestimation. If we have J number of groups, and for each group of genes $j \in$ J, a single correction parameter is used, then the adjusted denominator would be

$$\tilde{S}_i = (1-\lambda_j)S_i+\lambda_j t \tag{3-7}$$

Genes can be divided into J groups according to their quantiles. For instance, splitting the genes into their 5% quantiles render 20 groups of genes, while splitting the genes into their 1% quantiles would generate 100 groups. For each group, a different shrinkage parameter λ_j is applied. Grouping different genes according to their percentiles can solve the underestimation and overestimation problems more accurately. It is more likely that within each group, the different residual sum of squares have approximately the same degree of underestimation or overestimation. However, a more local correction can be applied where each gene has its own shrinkage parameter λ_i. Hence, for each gene i, if S_i is the residual sum of squares and t is the target to which the residual sum of squares is shrunk, then the adjusted denominator can be expressed as

$$\widetilde{S}_i = (1-\lambda_i)S_i+\lambda_i t \tag{3-8}$$

An important consideration in variance moderation techniques is the choice of an appropriate value for the shrinkage parameter λ. It is required to choose the shrinkage parameter, $\lambda \in [0,1]$, so that an underestimated S_i value has a larger value of λ, and hence a larger proportion of the target t is added; for an overestimated S_i , λ is required to be small. In our proposed correction scheme, the median, m, of all the residual sum of squares is chosen to be the target to which the residual sum of squares of all the genes are shrunk. The median is preferred to the mean, since it is more robust. The two schemes given by (3-7) and (3-8), will now be examined.

For each group of genes $j \in$ J, a single correction parameter λ_j is employed in (4-8), and the adjusted denominator is given by

$$\widetilde{S}_i = (1 - \lambda_j)S_i + \lambda_j m \tag{3-9.a}$$

where the correction parameter λ_j is given by

$$\lambda_j = \frac{m}{\overline{S}_j + m} \tag{3-9.b}$$

and $\overline{S}_j$ is the mean of the denominators of the F-statistic in group j.

If $\overline{S}_j$ <<m, there is a greater probability of $\overline{S}_j$ being underestimated and then $\lambda_j \rightarrow 1$. Hence, a large proportion of the median value is added to the denominator of the F-statistic, thus correcting for the underestimation problem. On the other hand, if $\overline{S}_j$ >>m, then there exists a greater probability of $\overline{S}_j$ being overestimated and then $\lambda_j \rightarrow 0$. Consequently, a very small proportion of the correction factor would be added to the denominator of the F-statistic, and hence the overestimation problem is also corrected. For a balanced $\overline{S}_j$, an appropriate value of λ_i is applied for the correction factor.

For the second correction scheme, as given by (3-10), each gene itself can be considered a single group and the adjusted denominator is then given by

$$\widetilde{S}_i = (1-\lambda_i)S_i+\lambda_i m \tag{3-10.a}$$

where

$$\lambda_i = \frac{m}{S_i + m} \tag{3-10.b}$$

If the variance S_i is very small (underestimated) or very large (overestimated), it is shrunk towards the median. If $S_i << m$, then $\lambda_i \rightarrow 1$. Hence, a large proportion of the median value is added to the denominator of the F-statistic, thus correcting for the underestimation problem. On the other hand, if $S_i >> m$, then $\lambda_i \rightarrow 0$. Consequently, a very small proportion of the correction factor would be added to the denominator of the F-statistic. Thus, the overestimation problem is also corrected. For a balanced S_i an appropriate value of λ_i is applied for the correction factor. All the existing techniques apply a single moderation factor to the whole set of genes, and thus correspond to the case of a fixed shrinkage parameter, while in our proposed algorithm varying shrinkage parameters are applied to different groups of genes. Figure 3.1 shows the shrinkage parameter λ as a function of the residual sum of squares S_i. As seen from this figure, as the number of groups increases, the variation of λ becomes smoother and its range wider. Although we assume that small (large) values are more probably underestimated (overestimated), there is a probability that the residual sum of squares are truly small (large). Hence, a good moderation technique should provide correction for overestimation as well as underestimation of S_i , while maintaining the variations of the residual errors of the different genes. The wide range of values available for the shrinkage parameter, in the proposed method, preserves the diversity of the residual sum of squares while correcting for the underestimation or overestimation.

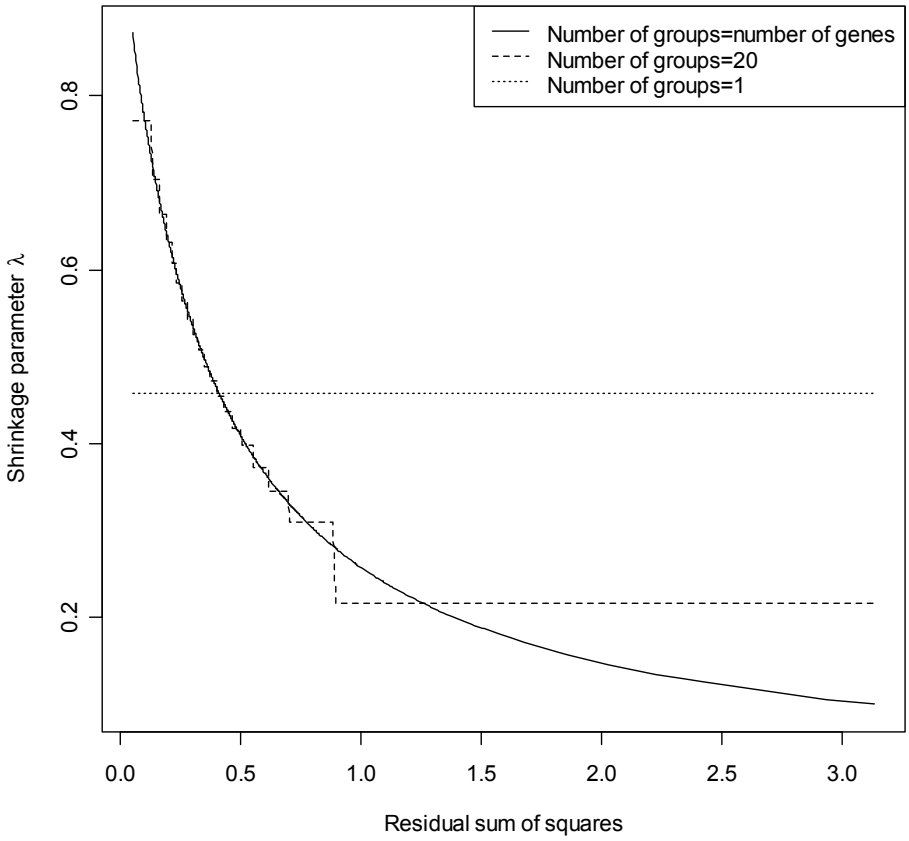

Figure 3.1 The shrinkage parameter for different number of groups.

Figure 3.2 shows the histograms of the residual errors when 1, 20 and p (p being the number of genes) shrinkage parameters are applied. The histograms show that by applying a large number of shrinkage parameters λ, the range of the residual errors can be kept large; this is useful in maintaining the true distribution of the residual errors and in smoothly shifting the small (large) values towards the larger (smaller) ones, thus correcting the underestimation (overestimation) of S_i.

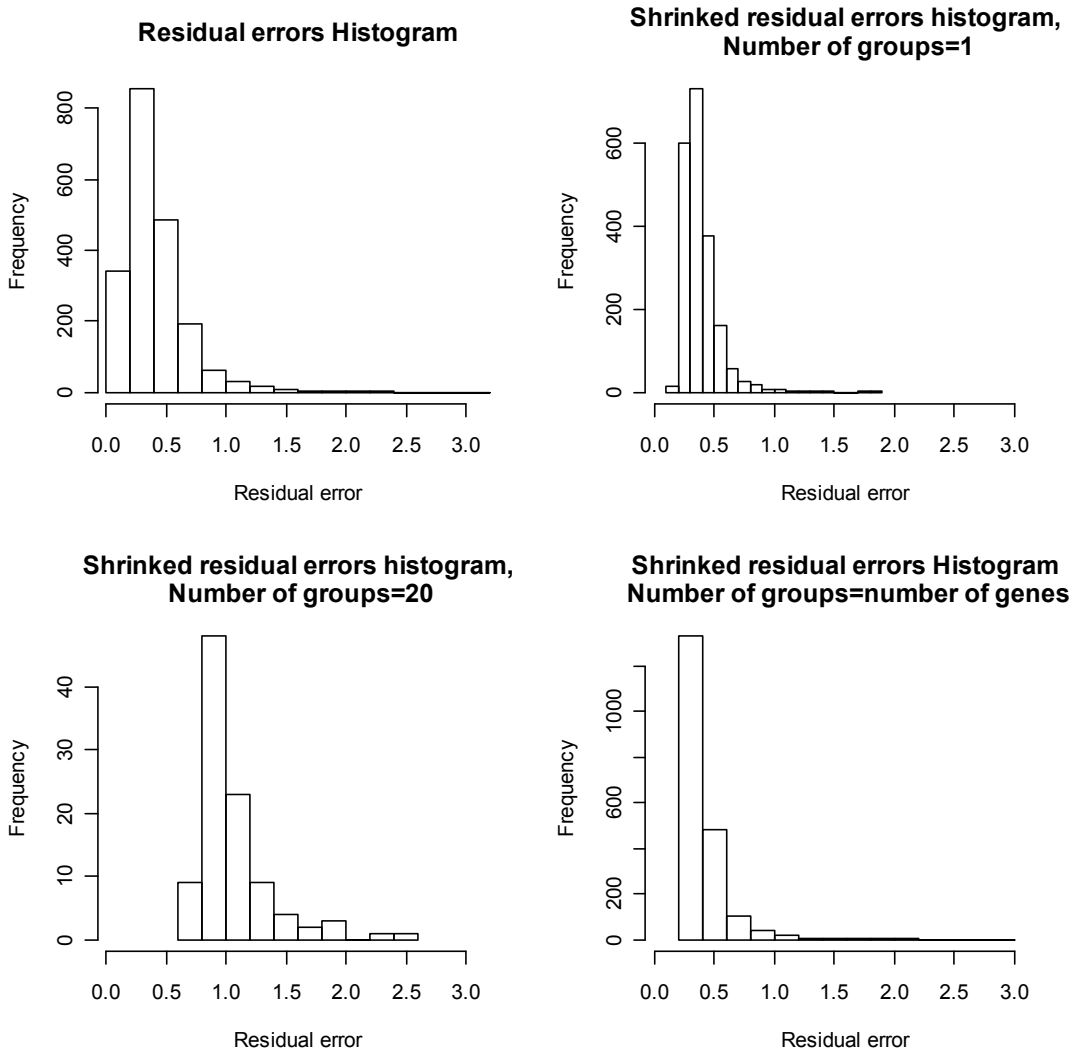

Figure 3.2 Histograms of the residual errors for different number of groups.

The different ranges for the residual errors are given in Table 3-2. It is seen from this table that the range of the residual sum of squares is the largest when the shrinkage parameter is chosen according to (3-10), that is, a parameter λ_i is chosen for each gene i. It is expected that this wide range for λ would yield results superior to those of the existing methods, where a single value for λ is chosen.

Table 3-2 Range of residual sum of squares

	Min	Max	Range
Without moderation	0.0508	3.1359	3.08
J = 1	0.1078	1.8567	1.748
J = 20	0.775	2.5319	1.761
J = *p*	0.287926	2.8577	2.569

3.3 Calculating p-values using Permutations

The permutation-based hypothesis testing methods have the advantage of not requiring the data to follow a particular distribution as compared to the parametric hypothesis testing methods. The permutation approach preserves the unknown correlation structure of the gene expression data and results in an empirical p-value. Resampling techniques, such as permutations and bootstrapping techniques, are the most widely used techniques, in microarray time-series data analysis, to evaluate the significance of a given statistic [3, 16, 17, 25]. They are generally preferable over using the standard F-tables, since these techniques do not assume any shape for the null distribution. It is shown in [35] that bootstrapping, as a resampling technique, is the most robust way for making corrections to the violation of the assumptions of normality and sphericity in repeated measures analysis, compared to other correction methods. Since we do not use the F-tables directly and use permutations as a resampling technique in our method, we are able to overcome the drawbacks of the violation of the above mentioned assumptions.

Permutation tests are exact only if the data points that are rearranged are *exchangeable* under the null hypothesis, that is, if the joint distribution of the observations remains unchanged under rearrangements of the data labels when the null hypothesis is true. This implies that the observations viewed individually must be identically distributed. Though permutation tests are often described as distribution free, again this is true only under the null hypothesis. The way of permuting the data depends on the null hypothesis to be tested. Problems involving complex relationships among variables may require permuting the residuals of some model instead of the raw data. Since equal means, variances and pair-wise correlations between time points, under the

null hypothesis, are sufficient conditions for the permutation procedure to be valid, permutation procedure is well-suited for time-series microarray data analysis. Moreover, in the null case, processes that run in time do not exist, and hence, there is no correlations function of time. Consequently, in addition to the mean, the variances and the pair-wise correlations between different measurements, for the same subject, in the null case, are equal. Since, the measurements under the null hypothesis are assumed to be identically distributed; the permutation procedure does not affect the correlation structure of the time-series microarray data, between different measurements. Whatever test statistic is employed in a randomization test, the null hypothesis is that of no effect of the treatment variable on any of the measurements, whether mean, variance or correlation.

3.3.1 Permutation Procedure

The concept of permutations relies on exchangeability. Permutation tests require relatively few assumptions and can be applied in a wide variety of settings. Design factors may be fixed or random, nested or crossed, and it is these features that determine which strategy should be used. To construct a permutation test, one must decide which units are to be permuted, whether the permutations should be restricted, and whether it is best to permute raw data or residuals. Since, the null hypothesis for repeated-measures is: the measurement for each sample, for each time point, is the same as the measurement that sample would have provided for any alternative assignment of the time points. Hence, a random number is generated for each sample independently of the other samples to determine which of the T measurements is to be assigned to the first treatment, then which one to the second treatment, and so on. For the bth permutation, the corresponding F-statistic, F^{*b}, is calculated.

3.3.2 Computation of the p-values

After computing the F^{*b}-statistic, the p-values are calculated using either the method of the gene-wise p-values or that of the pooled p-values. For the former method, it is assumed that the null distribution of each gene is different from that of the other genes. The p-value of gene i is only based on the null distribution generated from its permutation only. For B number of permutations,

the p-value of gene *i* is given by

$$p\text{-value}_i = \frac{1}{B}\sum_{b=1}^{B}\gamma_b \tag{3-11.a}$$

where

$$\gamma_b = \begin{cases} 1 & if\ F_i^{*b} \geq F_i \\ 0 & otherwise \end{cases} \tag{3-11.b}$$

In this method the equivalent statistic $\overline{Y_j}$ can be used instead of the RM F-statistic. For the different permutations, $\overline{Y_j^{*b}}$ is calculated instead of F^{*b}.

For the pooled p-values it is assumed that all genes follow the same null distribution of statistic. The p-value for each gene *i* will be

$$p\text{-value}_i = \frac{1}{p.B}\sum_{b=1}^{B}\upsilon_b \tag{3-12.a}$$

where

$$\upsilon_b = \begin{cases} 1 & if\ F_j^{*b} \geq F_i \quad , j = 1,\ldots,p \\ 0 & otherwise \end{cases} \tag{3-12.b}$$

and B is the number of permutations and *p* the total number of genes.

3.3.3 Gene-wise p-values

For RM ANOVA permutations, the maximum number of permutations is $(T!)^{n-1}$. The smallest obtainable p-value is the reciprocal of the number of permutations. Hence, if the number of time points or samples yields a very small number of permutations, bootstrapping [36] can be used instead.

a) *Equivalent statistic using permutations*

Instead of computing F, $\overline{Y_j}$ has been found to be an equivalent statistic [32]. For different permutations, $\overline{Y_i}$ and $\overline{Y_{total}}$ do not change. Also *n*, *T* and SS_T are the same for all permutations, and

the only parameter that actually changes with the different permutations is $\overline{Y_j}$. Any increase in the value of $\overline{Y_j}$ increases the numerator and decreases the denominator. Hence, in the permutation procedure, instead of computing the F-value an equivalent statistic $\overline{Y_j}$ is computed. This alternate statistic can be used if the permutations are done for each gene individually, and not otherwise.

b) *Proposed coarse-to-fine strategy*

If the required p-value is about 10^{-5}, the minimum number of permutations needed for each gene would be 10^5. Instead of performing such a very large number of permutations for thousands of genes, a coarse-to-fine strategy is proposed. A fine gradation is carried out for small p-values and a coarser one in the less interesting region where p-values are large. The genes of large p-values cannot be rejected, even if a large number of permutations is used.

Let *l* the number of permutations, and *h* the number of values F^{*b} that exceed F. If it is required to estimate *p*-value with a standard error which is some fraction *c* of *p*-value, then according to [37] number of permutations should be increased until the estimated standard error is within the same fraction c of the estimated *p*-value. This procedure stops when the estimated standard error is satisfactorily small. The procedure based on [37] and found in [38] as follows:

Step 1: Let the number of permutations be *l* (initially *l* is chosen as 100). For each permutation F^{*b} test statistic is calculated. Afterwards, *h* is computed for a given gene.

Step 2: If $h \geq \frac{1}{c^2 + l^{-1}}$, the gene of this *h* value is not further examined. The p-value is calculated as *h/l.*

Step 3: If $h < \frac{1}{c^2 + l^{-1}}$, then the number of permutations *l* is increased and used to examine the gene significance, and Steps 1 and 2 repeated.

Step 4: Continue with the Steps 1, 2 and 3 until the condition $h \geq \frac{1}{c^2 + l^{-1}}$ is satisfied or the number of iterations exceeds a pre-specified limit and its p-value is computed.

Step 5: Repeat Steps 1 to 4 for all the genes.

By this procedure instead of applying B permutations on each of the p genes, where B is very large, a small number of permutations is applied on all those genes that are likely to have large p-values and a higher number is applied only on those genes that are likely to have small p-values. Thus, it reduces the computational time and effort.

3.3.4 Pooled p-values

Instead of generating the null distribution for each gene, the null distribution can be computed for the whole set of genes. It can be assumed that at the null hypothesis of no differences, all the genes will be the same. In addition to determining the method used to evaluate the p-values, moderation is required to compensate for the underestimation or overestimation of the variances of the various genes.

3.4 Identifying Differentially Expressed Genes for Multiple Time-course Data Based on Mixed Design ANOVA

The proposed method can be easily extended to identify differentially expressed genes between several biological groups, such as treatment and control, for microarray time series data. In this case, the F-statistic results from a mixed design where the repeated measures, representing the time, is one factor and the various biological groups a second factor. The mixed design model for each gene is given by [33]

$$Y_{ijk} = \eta + \mu_j + \beta_k + \alpha_i + (\mu\beta)_{jk} + (\alpha\mu)_{ij} + \varepsilon_{ijk} \qquad (3\text{-}13)$$

where Y_{ijk} is the microarray value for the ith sample at the jth time point, and the kth group, η is the population mean under all fixed ratios, μ_j is the fixed effect of the time j, β_k is the fixed effect of the group k, α_i is the effect of the ith sample, $\mu\beta_{jk}$ and $\alpha\mu_{ij}$ are the interaction effects and ε_{ijk} is the random error of the ith sample at the jth time point, and the kth group.

In this case we would like to test if there is a difference between the various groups, such as treatment and control groups. Hence, the null hypothesis, H_0, states that the there is no effect of

groups, while the alternative hypothesis, H_1, states that there is a change between the different groups. The hypothesis can be summarized as follows:

H_0: $\beta_1 = \beta_2 = \beta_3 = \ldots = \beta_k$

H_1: H_0 is false

For each gene, data is arranged in a table as shown in Table 3-3. The columns indicate the time points and rows the samples, for each group.

The quantities shown in Table 3-3 are defined as follows:

$$\overline{Y_i} = \frac{1}{T}\sum_{j=1}^{T} Y_{ijk} \;,\; \overline{Y_{jk}} = \frac{1}{n_k}\sum_{i=1}^{n_k} Y_{ijk} \;,\; \overline{Y_k} = \frac{1}{T}\sum_{j=1}^{T} \overline{Y_{jk}} \;,\; \overline{Y_j} = \frac{1}{K}\sum_{k=1}^{K} \overline{Y_{jk}} \;,\; \overline{Y_T} = \frac{1}{K}\sum_{k=1}^{K} \overline{Y_k} \qquad (3\text{-}14)$$

The means $\overline{Y_i}$, $\overline{Y_k}$ and $\overline{Y_T}$ are used to compute the following quantities:

$$SS_B = nT\sum_{k=1}^{K} (\overline{Y_k} - \overline{Y_T})^2 \;,\; SS_{swg} = T\sum_{k=1}^{K}\sum_{i=1}^{n_k} (\overline{Y_i} - \overline{Y_k})^2 \qquad (3\text{-}15)$$

where SS_B is the sum of squares of the group levels, SS_{swg} is the sum of squares of the subjects-within-group variability. These quantities are used to calculate the group F-statistic using

$$F = \frac{K(n_k - 1)}{K - 1}\frac{SS_B}{SS_{swg}} \qquad (3\text{-}16)$$

After computing F-statistic, the same moderation scheme proposed in Section 3.2 is applied. In this case the denominator of the F-statistic, S_i, is the sum of squares of the subjects-within-group variability, SS_{swg}. In addition, the permutation procedure has been applied for the case of multi-biological groups, in a manner similar to that of the one-factor RM ANOVA as described in Section 3.3.

Table 3-3 Data arrangment for each gene.

Group 1		Time point 1	……	Time point j	……	Time point T	$\overline{Y_i}$
	Sample 1						
	…..						
	Sample i			Y_{ij1}			
	…..						
	Sample n_1						
	$\overline{Y_{jk}}$						$\overline{Y_k}$
Group 2		Time point 1	……	Time point j	……	Time point T	
	Sample 1						
	…..						
	Sample i			Y_{ij2}			
	…..						
	Sample n_2						
	$\overline{Y_{jk}}$						$\overline{Y_k}$
……	…	..		…..		….	
Group K		Time point 1	……	Time point j	……	Time point T	
	Sample 1						
	…..						
	Sample i			Y_{ijk}			
	…..						
	Sample n_k						
	$\overline{Y_{jk}}$						$\overline{Y_k}$
	$\overline{Y_j}$						$\overline{Y_T}$

3.5 Summary of the Proposed Method for Identifying Differentially Expressed Genes

In this chapter, we have proposed a method for identifying differentially expressed genes, not only for single but also for multiple biological groups. This proposed VSP method can be summarized as follows.

Step 1: If there is a single time-course, for each gene, the RM F-statistic is computed using (3-5). On the other hand if there are multiple time-courses, the mixed design F-statistic is computed using (3-13).

Step 2: The proposed moderation scheme given by (3-9) or (3-10) is applied to each gene.

Step 3: Permutations are used to compute the p-values.

- The number of permutations is initially determined.
- The moderation scheme applied to the F-statistic of each gene is applied to the permuted F^*-statistics.

Step 4: The p-value of each gene is calculated using the permuted F^*-statistics and the original RM (or mixed design) F-statistics.

Step 5: A threshold for the p-values is set in order to select the differentially-expressed genes.

The proposed algorithm is expected to surpass the existing algorithms for the following reasons. It takes into account the time dependency of the longitudinal data by applying the RM ANOVA (or mixed design) F-statistic. The moderation scheme, proposed and integrated with the F-statistic, is to overcome the inaccurate estimation of the variance due to limited number of samples. In addition, since the null hypothesis tested is that "all the means for all time points are equal", any significant change at time, whether it be a continuous change or a change at a single time point, can be identified easily. In the next chapter the proposed method is applied to both synthetic and real data sets to evaluate is performance.

Chapter 4

Experimental Results on the Identification of Differentially Expressed Genes

4.1 Synthetic and Real Datasets Description

In order to compare the performance of the proposed technique with that of the previous ones, we carry out a simulation study by generating synthetic data using the simulation utility of BATS [18]. The generated microarray profile $z_i(t)$, for each gene i is given by [18]

$$z_i(t)=s_i(t) + \xi_i(t), \tag{4-1}$$

where $s_i(t)$ is a function of time and $\xi_i(t)$ is an additive noise independent of $s_i(t)$.

If the gene i is not differentially expressed, $s_i(t) =0$ and $z_i(t)$ will consists only of the noise component $\xi_i(t)$. For the significant genes $s_i(t)\neq 0$, and can be represented by a polynomial function of time as

$$s_i(t)=c_{i0}P_0(t)+ c_{i1}P_1(t)+ c_{i2}P_2(t)+.....+ c_{il}P_l(t), \tag{4-2}$$

where $P_m(t)$ is a normalized Legendre polynomial of degree m. $P_m(t)$ is generated in Matlab and given by

$$P_m(t)= \sqrt{\frac{2m+1}{2}}\frac{1}{2^m m!}\left[\frac{d^m}{dt^m}(t^2-1)^m\right] \tag{4-3}$$

Although the error distributions are independent, there are correlation structures between the various time points attributed to the polynomial function of time $s_i(t)$. As shown in (4-2), , since for each gene the various time points share the common coefficients, $c_{i0}, c_{i1}, c_{i2}, ..., c_{il}$, the measurements generated for the same gene are correlated. We shall now further elaborate on the correlation between various time points for each gene. The autocorrelation between any two time points t and $t+\tau$, for a given gene, can be expressed as

$$R[z(t),z(t+\tau)] =R[s(t), s(t+\tau)]+R[s(t), \xi(t+ \tau)]+R[\xi(t),s(t+\tau)]+R[\xi(t), \xi(t+ \tau)]. \tag{4-4}$$

Since the error terms ξ's are independent, the autocorrelation $R[z(t),z(t+\tau)]$ becomes

$$R[z(t),z(t+\tau)] = R[s(t), s(t+\tau)] \tag{4-5}$$

where
$$R[s(t), s(t+\tau)] = \int_{-\infty}^{\infty} s(t)s(t+\tau)dt \tag{4-6}$$

However, since $s(t)$ is function of $P_m(t)$, which is a normalized Legendre polynomial of degree m, the time t is scaled to lie in the range $(-1,1)$, and hence $R[s(t) ,s(t+\tau)]$ is given by

$$R[s(t) ,s(t+\tau])= \int_{-l}^{l} s(t)s(t+\tau)dt \tag{4-7}$$

$$= \int_{-1}^{1} \sum_{m=1}^{l} c_m P_m(t) \sum_{m=1}^{l} c_m P_m(t+\tau)dt \tag{4-8}$$

$P_m(t+\tau)$ can be easily expressed in terms of $P_{m-1}(t)$, $P_{m-2}(t)$,.., and τ. For instance,

$$P_1(t+\tau)= P_1(t)+\tau, \tag{4-9}$$

$$P_2(t+\tau)= P_2(t)+ 3\tau P_1(t)+ \tau^2 \quad \text{... .. and so on.} \tag{4-10}$$

In addition,

$$\int_{-1}^{1} P_m(t)P_k(t)dt = \begin{cases} \dfrac{2}{2m+1} & k = m \\ 0 & k \neq m \end{cases} \tag{4-11}$$

When we substitute *(4-9),(4-10)* and *(4-11)* in *(4-8), R[s(t),s(t+τ)]* becomes a function of the coefficients $c_{i0},c_{i1}, c_{i2},....., c_{il}$, and the delay τ. Hence, $R[s(t),s(t+\tau)]\neq 0$, and consequently $R[z(t),z(t+\tau)] \neq 0$, that is, there is a correlation structure between the time points.

The degree of the polynomial $s_i(t)$ varies for the various genes. For each significant gene i, the simulation utility samples the degree of the polynomial, l, from a discrete uniform distribution in [1, L_{max}]. L_{max} is chosen to be 6. For each significant gene i, the vector of coefficients c_i is randomly sampled from a multivariate normal distribution $N(0, \sigma^2\tau_i^2 Q_i^{-1})$, where σ^2 is chosen as 0.2 and matrix $Q_i = diag\ (1^{2v_i}, 2^{2v_i}, ..., L_i^{2v_i})$, where $v_i \sim U([0,1])$, U denoting a uniform distribution. The gene specific variance τ_i^2 is randomly sampled from $U([2, 6])$. Two models are considered for $\xi_i(t)$. In the first model, $\xi_{ia}(t)$ is independent and identically distributed (i.i.d) noise which follows either

Gaussian $N(0, \sigma^2)$ or Student T distribution, where the variance is set to 0.2. In the second model, $\xi_{ib}(t)$ is given by

$$\xi_{ib}(t) = \xi_{i1}(t) + \xi_{i2}(t) \tag{4-12}$$

where $\xi_{i1}(t)$ is independent and identically distributed (i.i.d) noise, and follows either Gaussian $N(0, \sigma^2)$ or Student T distribution, where the variance is set to 0.1, and $\xi_{i2}(t)$ for each sample is sampled from a multivariate normal distribution $N(0,\sum)$, where $\sum$ is covariance matrix of equal variances of 0.1 and equal covariances of 0.06 . $\xi_{i2}(t)$'s for each sample are independent of the other samples.

The data-sets are created with randomly generated sets of profiles, different values of replicates and three different noise realizations. Two examples for the randomly generated polynomials $s_i(t)$ are shown in Figure 4.1. Since for each gene the various time points share the common coefficients, c_{i0}, c_{i1}, c_{i2},….., c_{il}, the measurements generated for the same gene are correlated.

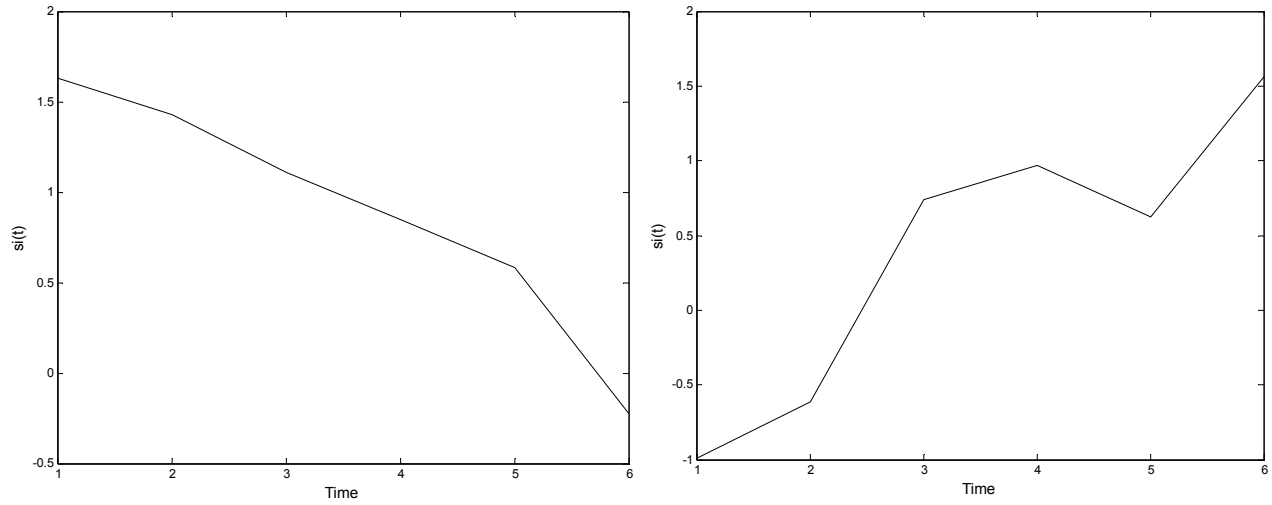

Figure 4.1 Examples of the generated time series, $s_i(t)$.

In our study, 300 datasets are generated to mimic the different possible structures of real data. The synthetic data parameters are set to meet common values found in real data sets. For instance, in the real data set [39] that we use later, the number of genes is 1900 and the number of time points is 6; in the synthetic data, we have generated the datasets where the total number of genes is 2000, and the number of time points is 6. In real datasets there is usually a small number of replicates (2 or 3

replicates), whereas Lee *et al.* [40] have recommended that for real experiments the number of replicates should be at least 3; hence, in our simulation study we have generated synthetic data with the number of replicates equal to 3.The generated datasets have 250 genes randomly chosen to be differentially expressed, corresponding to 12.5 % of the total number of genes. The signal to noise ratio (SNR) is chosen in the interval 2 to 6 and L_{max} as 6. Since the noise in the microarray data can have a heavier tail than in Gaussian noise, simulations are performed under three scenarios of i.i.d. noise, Gaussian of zero mean and variance 0.2, and Student T with 5 and 8 degrees of freedom. For each of the three scenarios of i.i.d. noise, 100 datasets are generated.

Next, in order to compare the performance of the gene-wise and pooled p-values methods, another 300 heterogeneous datasets are additionally generated, where two different types of noise are present in each dataset. For the first 100 datasets, Gaussian noise is added to some of the genes in each dataset, and Student T with 5 degrees of freedom added to the rest. Similarly, in the second 100 datasets, Gaussian noise and Student T with 8 degrees of freedom are added to each dataset, while in the remaining 100 datasets, Student T with 5 and 8 degrees of freedom are added to each dataset. Then, in order to assess the performance of our proposed algorithm, additional 300 datasets of 6 replicates and 300 datasets of 10 replicates are generated.

Finally, the performance of the proposed method is compared with that of the existing time-series methods. In addition to the synthetic data sets generated using BATS as explained in (4-1)-(4-3), in order to evaluate the sensitivity of the various time series methods including the proposed one, in the identification of genes that are not changing continuously in time, but rather have a significant change at only one or two time points, we assume the generated microarray profile $z_i(t)^*$, for each gene i, to be of the form

$$z_i(t)^* = r_i(t) + \xi_i(t) \tag{4-13}$$

where $r_i(t)$ is a function of time and $\xi_i(t)$ is an additive noise independent of $r_i(t)$, where the two aforementioned models for $\xi_i(t)$ are employed.

For the significant genes $r_i(t) \neq 0$, $r_i(t)$ is represented by a signal that is zero everywhere and only differentially expressed at only one or two time points as

$$r_i(t) = \begin{cases} a & t = t_a \\ 0 & otherwise \end{cases} \quad \text{(4-14.a)}$$

or

$$r_i(t) = \begin{cases} a & t = t_a \\ b & t = t_b \\ 0 & otherwise \end{cases} \quad \text{(4-14.b)}$$

The function $r_i(t)$ is chosen randomly to follow either (4-14.a) or (4-14.b). The time points t_a and t_b are chosen from a uniform distribution representing the range of the existing time points, and the magnitudes a and b determined so that the range of the SNR lies between 2 and 6. Two examples for the randomly generated functions $r_i(t)$ are shown in Figure 4.2.

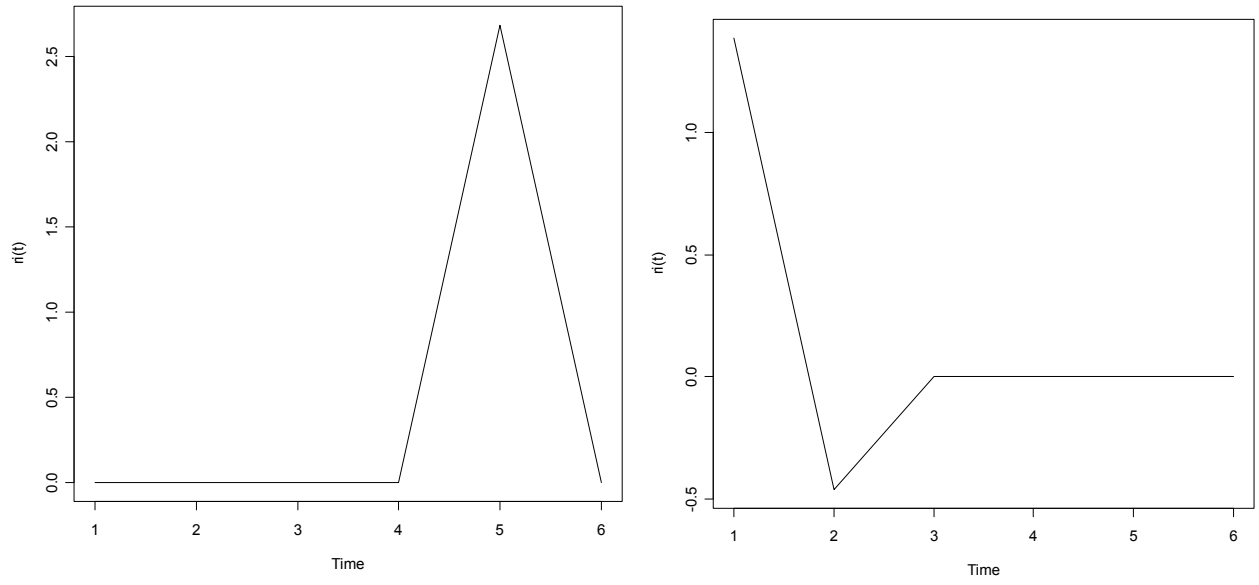

Figure 4.2 Examples of the generated time series, $r_i(t)$.

A total of 300 datasets are generated, where the synthetic data parameters are set to match the synthetic data generated using BATS. The total number of genes is 2000, the number of time points is 6, the number of replicates is 3 and 250 genes are randomly chosen to be differentially expressed. Simulations are performed under three scenarios of i.i.d. noise, Gaussian of zero mean and variance

0.2, and Student T with 5 and 8 degrees of freedom. For each of the three scenarios of i.i.d. noise, 100 datasets are generated.

In addition to this, real microarray data set found in [39], which consists of 1900 genes measured at 6 time points with 8 observations per time point, is used. Using this data set, Lobenhofer *et al*. [39] have studied the effect of Estrogen on inducing cell cycle progression in hormone-responsive breast cancer cells. Estrogen-treated cells were harvested after 1, 4, 12, 24, 36, and 48 h of treatment.

Furthermore, for multiple time course data, we carry out a simulation study by generating synthetic data, where the generated microarray profile z_i, for each gene *i,* for each group, is generated using

$$z_i^{(g)} = s_i(t)^{(g)} + \xi_i \qquad (4\text{-}15)$$

If the gene *i* is not differentially expressed, $s_i(t)^{(g)} = 0$ and z_i will consists only of the noise component ξ_i. For the significant genes $s_i(t) \neq 0$, and $s_i(t)^{(1)} \neq s_i(t)^{(2)}$, i.e. the significant gene *i* has two different expression profiles for the two groups. The function $s_i(t)$ is represented by a polynomial function of time similar to that found in (4-2).

Moreover, a real dataset that investigates the transcriptional response to three different abiotic stressors (Salt, Cold and Heat) in potato [41] is used. The dataset has 4 series (1 Control and 3 types of stress: Heat, Salt and Cold), and the control series is a reference microarray data that is not subjected to a stress condition. Each series consists of 3 time points, harvested at 3, 9 and 27 hours and 3 replicated samples at each time point.

4.2 Results of Single Time-series Data

In order to compare the performance of the proposed technique with that of the previous ones, the following metrics are used.

1. False positive (FP), defined as the number of false positives among the significant genes
2. False negative (FN), defined as the number of false negatives among the non-significant genes.

3. True positive (TP), defined as the number of genes correctly identified significant.

4. True negative (TN), defined as the number of genes correctly identified insignificant.

5. Sensitivity, defined as the ratio of the TP to (TP+FN).

6. Specificity, defined as the ratio of the TN to (TN+FP).

4.2.1 Coarse-to-fine Gene-wise p-values Versus the Ordinary Gene-wise p-values

The coarse-to-fine gene-wise method proposed in section 3.3.3 is now compared to the existing gene-wise method. The RM ANOVA is used to calculate the F-statistics for different genes. In order to calculate the individual p-values of the genes, the coarse-to-fine strategy is employed, where the standard error c is chosen as 0.1 and the initial number of permutations l as 100. The number of permutations is incremented by 100 every time the condition $h \geq \frac{1}{c^2 + l^{-1}}$ is not satisfied, or the number of permutations is still less than 100,000. In the existing gene-wise method, the same number of permutations is applied.

Figure 4.3 shows the true positives (TP) and false positives (FP) for both the methods for various p-values. Furthermore, the sensitivity and the specificity of both the methods are compared at a significance level of p-value = 0.001, which is equivalent to maximum false positives of 2 genes. The average results taken over 300 data sets are given in Table 4-1.

Table 4-1 Sensitivity and Specificity for the Existing and Proposed gene-wise p-values methods

Permutation Method	Sensitivity	Specificity
Existing	34.27%	99.9%
Proposed	33.72%	99.9%

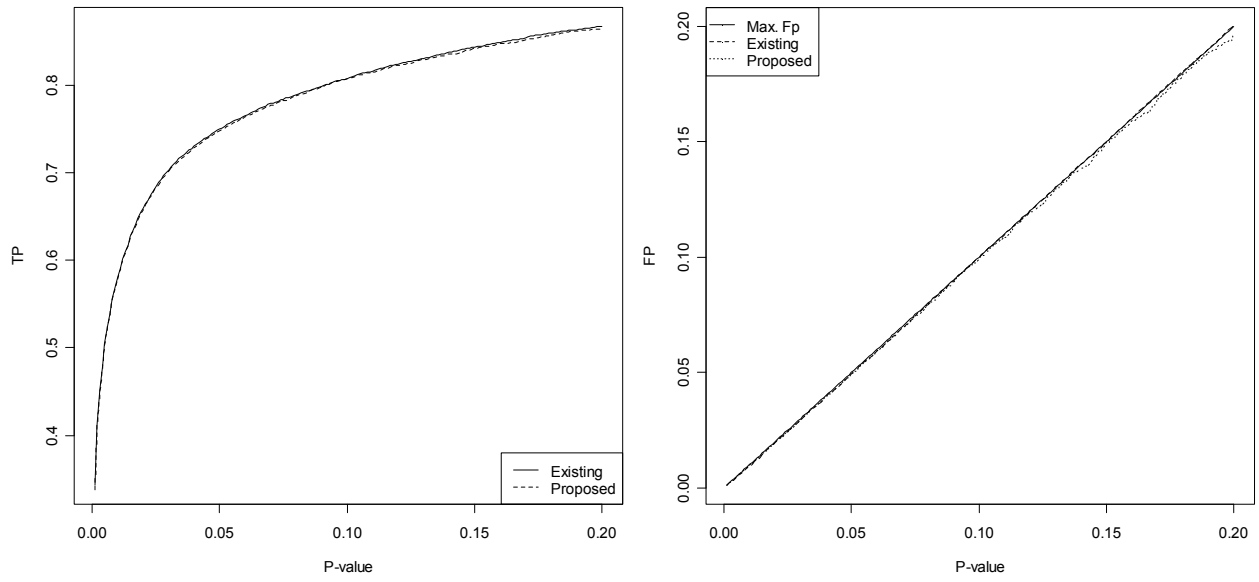

Figure 4.3 TP and FP for the existing and proposed gene-wise p-values methods.

As seen from Figure 4.3 and Table 4-1, the proposed and existing methods give almost the same sensitivity and specificity. However, the existing gene-wise method takes on an average 420 minutes of computation, while our proposed coarse-to-fine gene-wise p-values method requires only 42 minutes, thus providing a 10-fold reduction in the computational time. Since, the coarse-to-fine gene-wise p-values method takes less computational time and effort and its performance is compared to that of the pooled p-values, our proposed technique for gene-wise p-values is preferred.

4.2.2 Gene-wise p-values Versus Pooled p-values

We now compare the performance of the gene-wise p-values method to that of the pooled p-values method.

a) *Homogeneous Datasets*

In this case, the added noise in each dataset is of a single type. The gene-wise p-values have larger computational load than that of the pooled p-values. First, the two methods are compared without the use of any correction factors, in order to examine whether the computational burden of the gene-

wise p-values gives more accurate results. The RM ANOVA is used to calculate the F-statistics for different genes. In order to calculate the individual p-values of the genes, the coarse-to-fine strategy is applied, where the standard error c is chosen as 0.1 and the initial number of permutations l as 100. The number of permutations is incremented by 100 every time the condition $h \geq \frac{1}{c^2 + l^{-1}}$ is not satisfied, or the number of permutation is still less than 100,000.

In the pooled p-values algorithm, one null distribution is generated from all the genes at all the time points, and only 100 permutations are performed. The TP and FP values are plotted as functions of the p-value for both the methods and shown in Figure 4.4. Since the q-values are proportional to the p-values [25], the TP and FP values can be plotted alternatively as functions of the q-value. Moreover, the sensitivity and specificity of both the methods are compared at a significance level of p-value = 0.001, which is equivalent to maximum false positives of 2 genes. The average results over 300 datasets are shown in Table 4-2.

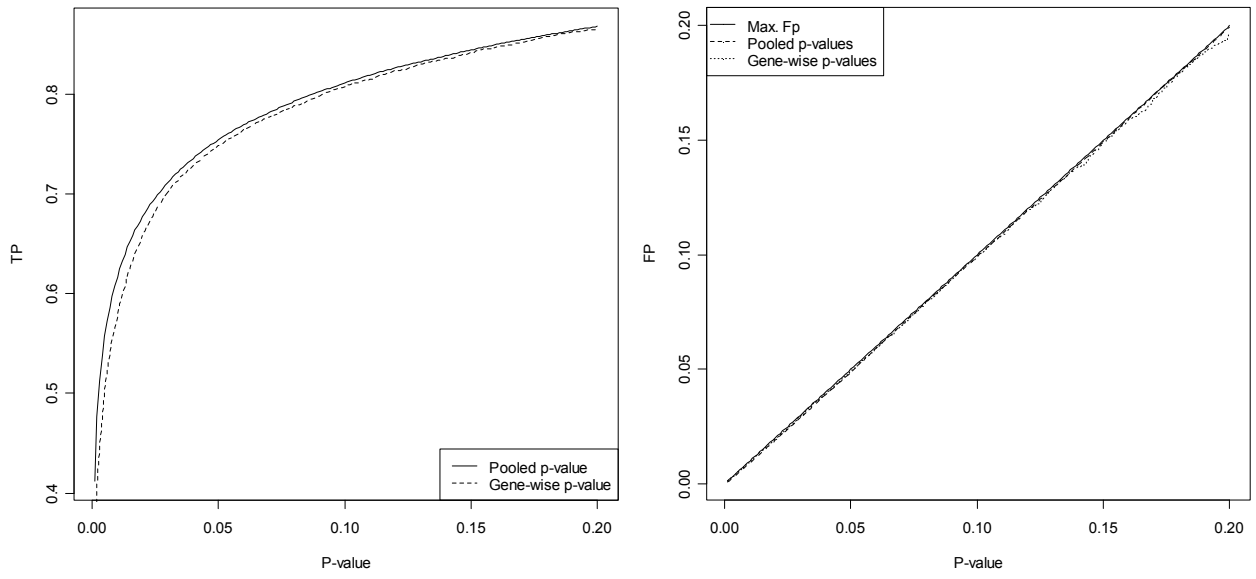

Figure 4.4 TP and FP for the gene-wise and pooled p-values methods.

Table 4-2 Sensitivity and Specificity for the gene-wise and pooled p-values methods

Permutation Method	Sensitivity	Specificity
Coarse-to-fine gene-wise p-values	33.72%	99.9%
Pooled p-values	41.15%	99.93%

Although the pooled p-values method outperforms the gene-wise p-values method, as seen from Figure 4.4 and Table 4-2, additional simulation is carried out using heterogeneous datasets.

b) *Heterogeneous Datasets*

In order to further examine the performance of the pooled p-values, heterogeneous datasets are employed, where two types of noise exist in each dataset. The TP and FP values are plotted as functions of the p-value for both the methods and shown in Figure 4.5. Furthermore, the sensitivity and specificity of both the methods are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-3.

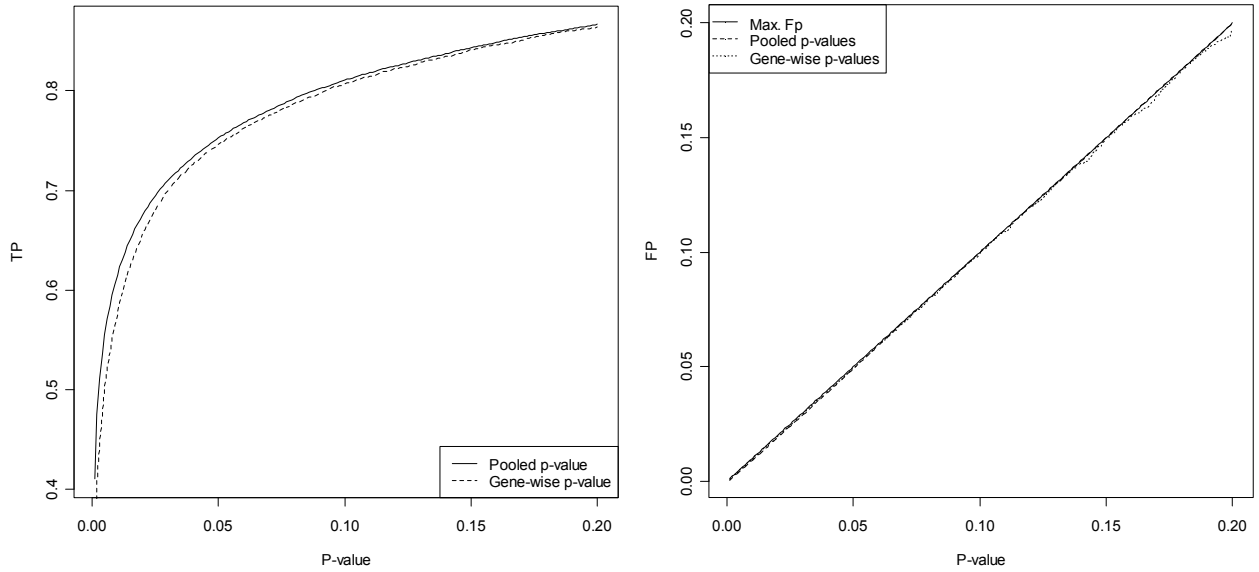

Figure 4.5 TP and FP for the gene-wise and pooled p-values methods in the heterogeneous case.

Table 4-3 Sensitivity and Specificity for the gene-wise and pooled p-values methods

Permutation Method	Sensitivity	Specificity
Coarse-to-fine gene-wise p-values	33.64%	99.9%
Pooled p-values	41.02%	99.94%

As seen from Figure 4.5 and Table 4-3, the performance of the pooled p-values method is still better than that of the gene-wise p-values approach even for heterogeneous datasets, and hence the pooled p-values method is preferred and recommended as it needs a much less computational effort. Since the pooled p-values method can be further improved by variance moderation, we now compare our proposed moderation technique to that of the existing moderation techniques.

4.2.3 Proposed Moderation for Different Quantiles

The RM ANOVA is used to calculate the F-statistics for different genes. Genes are divided into J groups according to their quantiles. For instance, splitting the genes into their 5% quantiles render 20 groups of genes, while splitting the genes into their 1% quantiles, generate 100 groups. For each group, a different shrinkage parameter λ_j is applied as in (3-9) and (3-10). The TP and FP values for different values of J are plotted as functions of the p-value and shown in Figure 4.6. The TP and FP values can be plotted alternatively as functions of the q-value [25]. The q-values being proportional to the p-values, they result in similar plots. In addition, the sensitivity and specificity for various values of J are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-4.

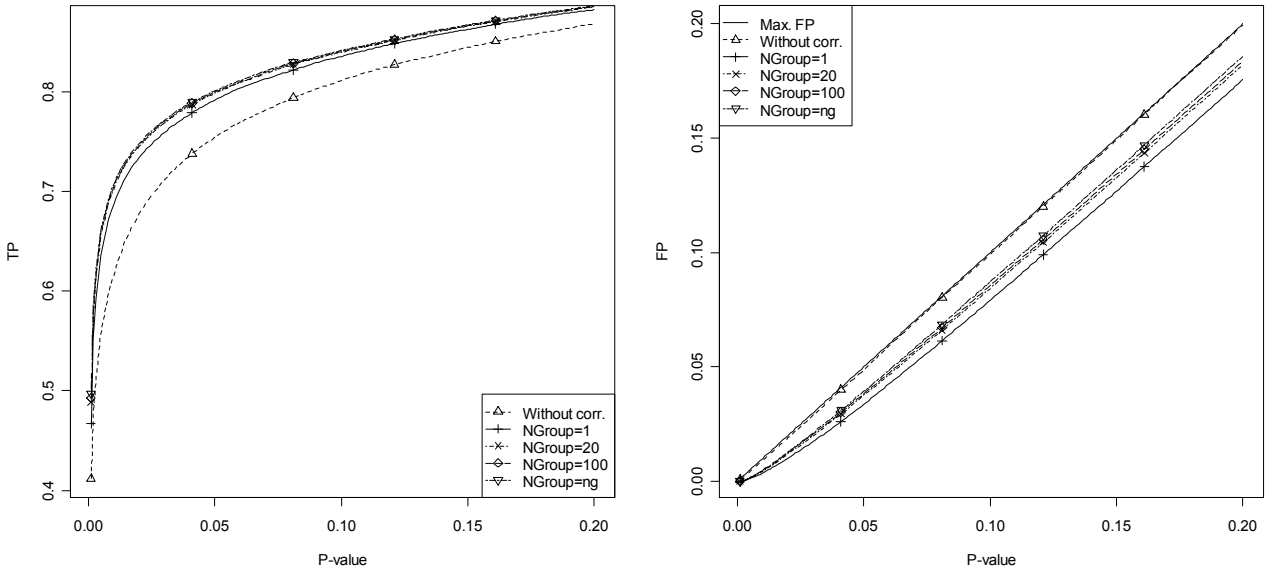

Figure 4.6 TP and FP for the proposed moderation technique.

Table 4-4 Sensitivity and Specificity for the proposed moderation technique

Number of groups (quantiles)	Sensitivity	Specificity
No correction	41.15%	99.93%
1(100%)	46.67%	99.99%
20(5%)	48.87%	99.99%
100(1%)	49.27%	99.99%
p (0.05%)	49.66%	99.99%

As seen from both Figure 4.6 and Table 4-4, the proposed moderation technique for any number of groups outperforms the pooled p-vales method without any correction, in terms of both the sensitivity and the specificity. For the same significance level, the moderation that considers each gene as a single group, i.e., choosing a shrinkage parameter λ for each gene, yields the best performance in terms of detecting the true positives. Hence, the algorithm, where each gene by itself is considered a single group, is further investigated; henceforth this algorithm is referred to as the variable shrinkage parameter (VSP) method.

4.2.4 The Proposed VSP Method using Different Number of Replicates

In this section, the performance of the VSP method is further examined using 3, 6 and 10 replicates. The TP and FP values for different number of replicates are plotted as functions of the p-value and shown in Figure 4.7. The sensitivity and specificity for 3, 6 and 10 replicates are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-5.

Table 4-5 Sensitivity and Specificity for the proposed moderation technique

Number of replicates	Method	Sensitivity	Specificity
3	No correction	41.15%	99.93%
	Proposed	49.66%	99.99%
6	No correction	74.60%	99.91%
	Proposed	77.47%	99.96%
10	No correction	85.09%	99.90%
	Proposed	85.99%	99.94%

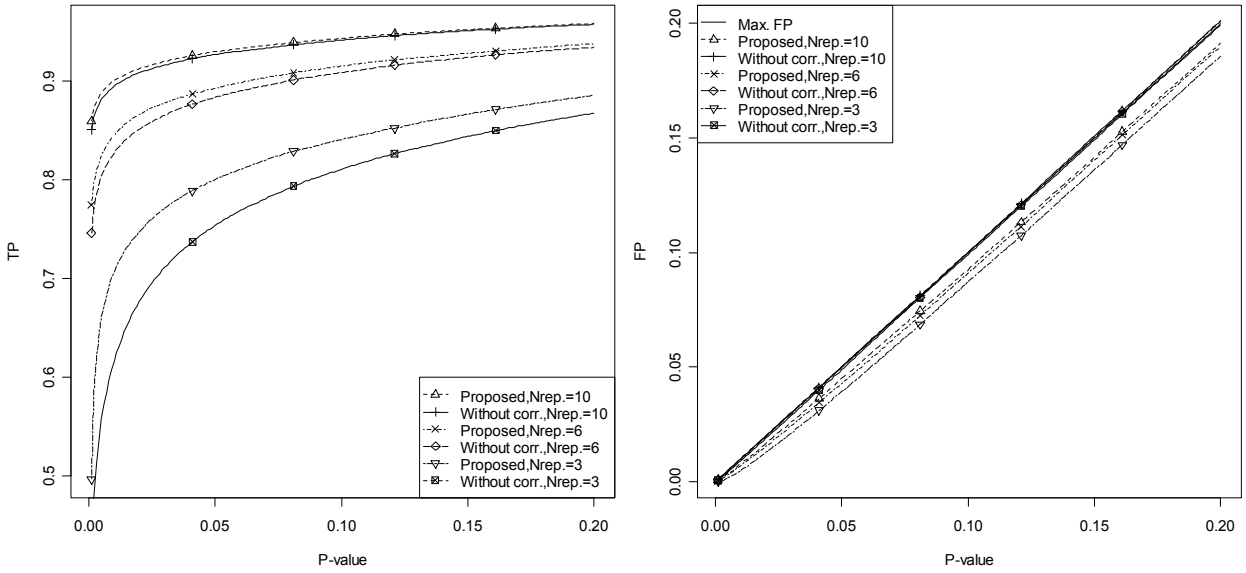

Figure 4.7 TP and FP for the proposed moderation for different number of replicates.

As seen from Figure 4.7 and Table 4-5, the sensitivity increases with the number of replicates. The proposed VSP method still outperforms the method without correction, in terms of both the sensitivity and specificity. Since moderation essentially corrects for underestimation and overestimation of S_i due to the limited number of replicates, the proposed VSP method yields a statistic that approaches the ordinary F-statistic as the number of replicates increases, as it should with any moderation technique.

4.2.5 Performance Comparison of the Proposed Method with Existing Moderation Techniques

The performance of the proposed VSP method is now compared to that of the 90th quantile of Efron *et al.* [20], the empirical Bayes correction factor method of Smyth [22] and the variance shrinkage method of Cui *et al.* [23]. Figures 4.8 and 4.9 show the TP and FP values as functions of the p-value for the various correction techniques, as well as for the original statistic without any correction, for the first and the second error models respectively. In order to make a fair comparison, the proposed correction method as well as the techniques of Efron *et al.*, Smyth and Cui *et al*. are applied to the RM F-statistic. Furthermore, the sensitivity and specificity of these

methods are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-6.

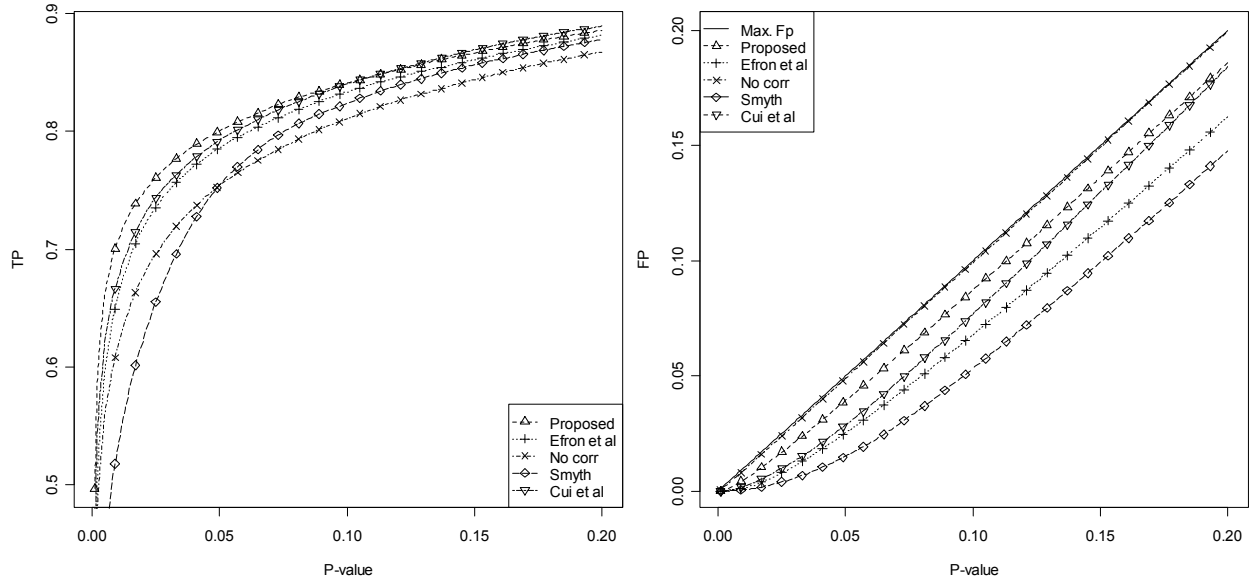

Figure 4.8 TP and FP for the different moderation techniques for the first error model.

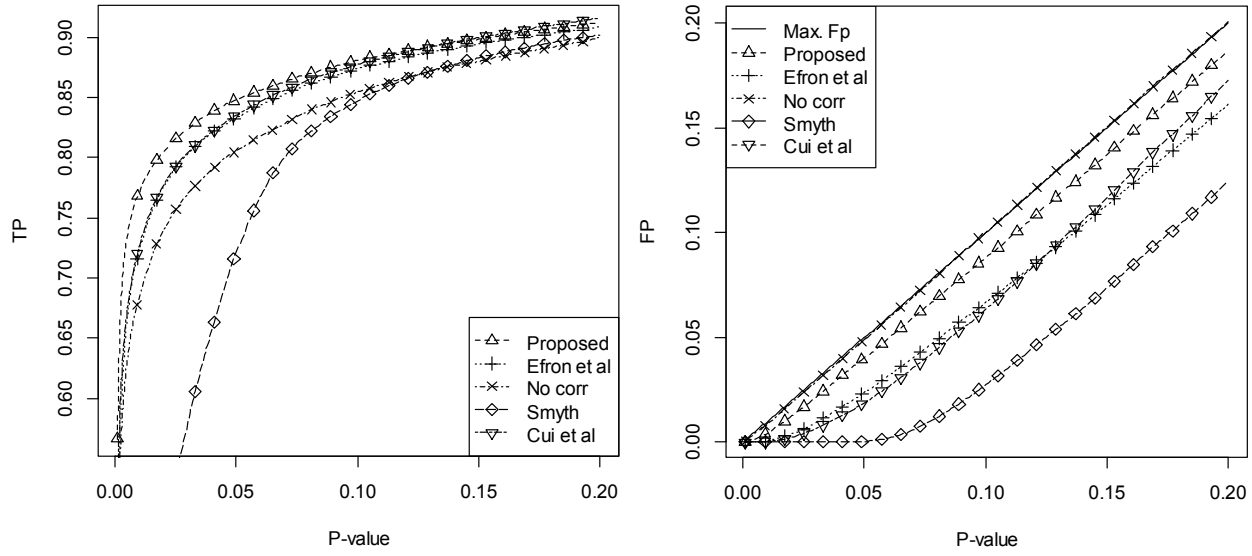

Figure 4.9 TP and FP for the different moderation techniques for the second error model.

Table 4-6 Sensitivity and Specificity for the moderation methods

Moderation Method	First Error Model $\xi_{ia}(t)$		Second Error Model $\xi_{ib}(t)$	
	Sensitivity	Specificity	Sensitivity	Specificity
No correction	41.15%	99.93%	48.15%	99.94%
Proposed VSP method	49.66%	99.99%	56.61%	100%
Efron *et al*. [20]	41.46%	99.99%	47.56%	100%
Smyth [22]	28.88%	99.99%	10.2%	100%
Cui *et al.* [23]	46.34%	99.99%	50.84%	100%

As seen from Figures 4.8 and 4.9, the specificity of any of the techniques is better than that of the ordinary statistic. Smyth's technique has the lowest FP rate, resulting in very good specificity; in spite of this, this technique has the lowest sensitivity, as seen from Table 4-6. It could be useful for ranking genes, but its ability to detect TP is very low, even when compared to the F-statistic without correction. Efron's method as well as that of Cui *et al.* has better sensitivity than the original RM F-statistic without correction and a small FP rate. From Figures 4.8 and 4.9 and Table 4-6, it is seen that the proposed VSP method is the most powerful, in the sense that it has the best results in terms of the TP rate, while maintaining a satisfactory FP rate. Although all the correction methods improve the specificity, the proposed correction technique has the best sensitivity, for the two error models.

4.2.6 Performance Comparison of the Proposed Method with Existing Time-series Methods

Our proposed algorithm is now compared to the existing techniques for time-series microarray data. The genes identified as significant using our VSP method are compared to those identified by EDGE [4], SAM [16] and Oriogen [3] methods. In SAM, there are two options for time-series data analysis: the slope and the area methods. In the slope method, each time-course is characterized by a slope, while in the area method, the signed area under the time-course curve is computed. Then, static data statistic is applied either on the slope or on the area. First the results of the synthetic data used in previous sections are introduced. Then, the results of the synthetic data generated using (4-13)-(4-14) are shown.

a) *Datasets Changing Continuously in Time*

The TP and FP values as functions of the p-value are shown in Figures 4.10 and 4.11. Also, the

sensitivity and specificity of these methods are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-7.

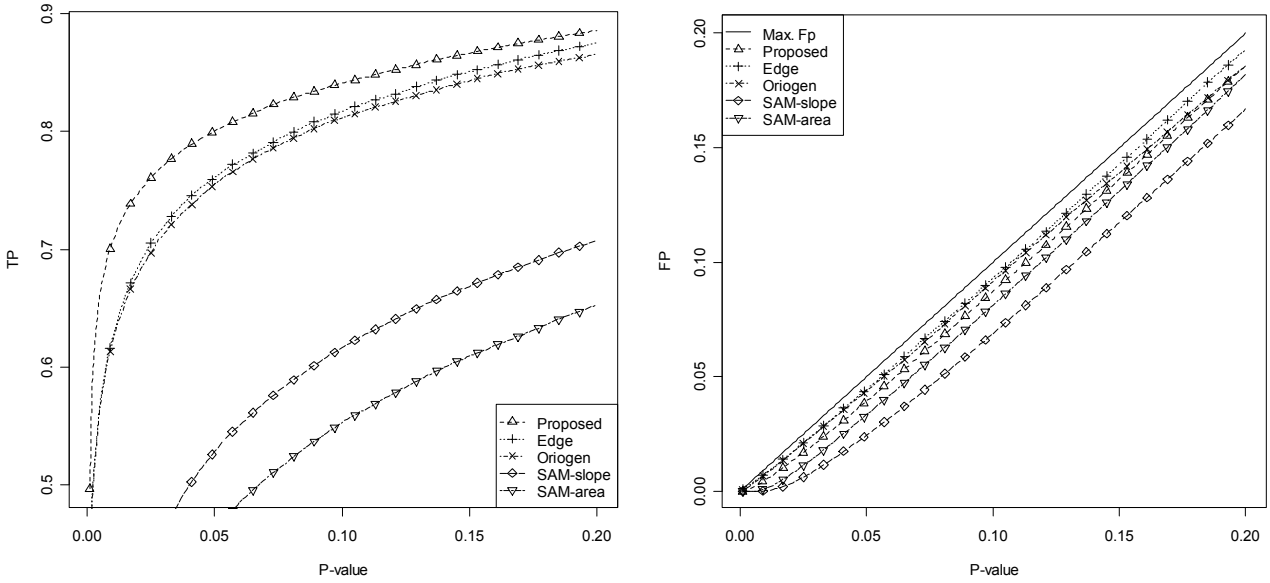

Figure 4.10 TP and FP for several time-series methods for the first error model (a).

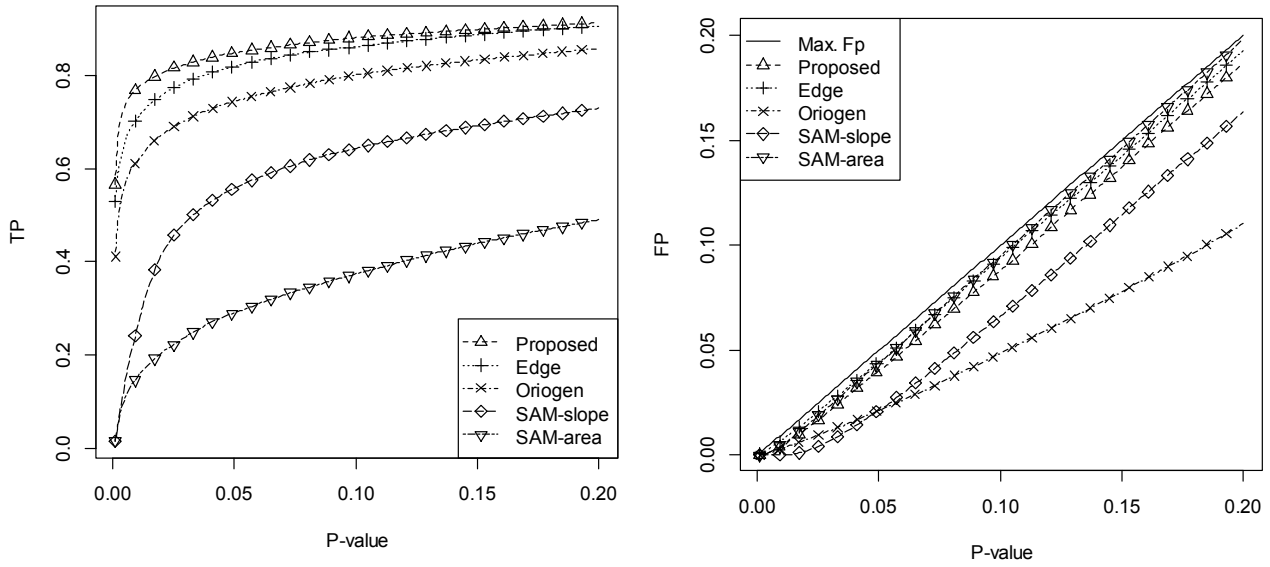

Figure 4.11 TP and FP for several time-series methods for the second error model (a).

Table 4-7 Sensitivity and Specificity for the Time-series Methods (a)

Time-series Method	First Error Model $\xi_{ia}(t)$		Second Error Model $\xi_{ib}(t)$	
	Sensitivity	Specificity	Sensitivity	Specificity
Proposed VSP method	49.66%	99.99%	56.61%	100%
SAM(slope) [16]	1.59%	100%	1.6%	100%
SAM(area) [16]	1.6%	100%	1.59%	100%
EDGE [4]	43.05%	99.89%	52.9%	99.94%
Oriogen [3]	41.44%	99.94%	41.13%	99.98%

As seen from Figures 4.10 and 4.11 and Table 4-7, all the techniques have FP values that are below the maximum allowable FP value; however, the proposed method outperforms the EDGE and Oriogen methods in terms of the specificity and surpasses all the methods in terms of the sensitivity, for the two error models.

b) *Datasets Changing at a Few Time Points*

The TP and FP values as functions of the p-value are shown in Figures 4.12 and 4.13. Also, the sensitivity and specificity of these methods are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-8.

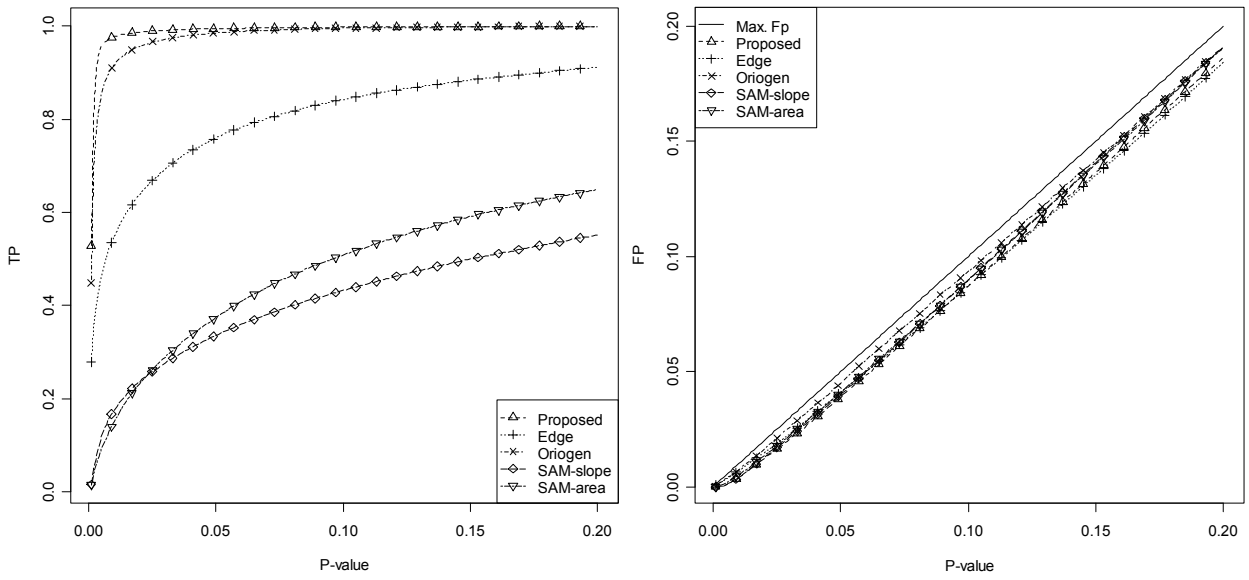

Figure 4.12 TP and FP for several time-series methods for the first error model (b).

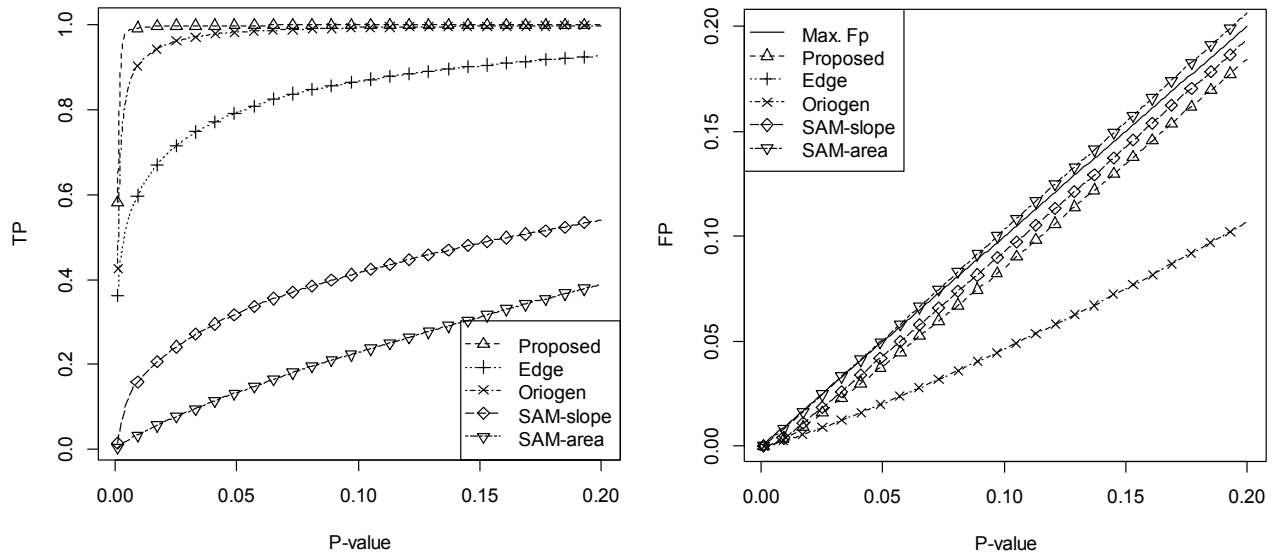

Figure 4.13 TP and FP for several time-series methods for the second error model (b).

Table 4-8 Sensitivity and Specificity for the Time-series Methods (b)

Time-series Method	First Error Model $\xi_{ia}(t)$		Second Error Model $\xi_{ib}(t)$	
	Sensitivity	Specificity	Sensitivity	Specificity
Proposed VSP method	52.83%	100%	58.19%	100%
SAM(slope) [16]	1.58%	99.99%	1.57%	99.99%
SAM(area) [16]	1.44%	99.99%	0.46%	99.96%
EDGE [4]	27.9%	99.9%	36.28%	99.96%
Oriogen [3]	44.83%	99.96%	42.58%	99.99%

As seen from Figures 4.12 and 4.13 and Table 4-8, the proposed method outperforms all the methods in terms of the sensitivity and the specificity, for the two error models.

The superior performance of our VSP method compared to the existing time series methods, regardless of the nature of the differentially expressed gene, can be attributed to several reasons. Since SAM applies a statistic that is not derived for time-series data, it has the lowest ability to detect differentially expressed genes. Both Oriogen and EDGE, unlike the proposed VSP method, do not account for the inaccuracy of the variance estimation in small samples, and the proposed moderation scheme cannot be integrated directly to these techniques. In addition, the statistic used in EDGE is based on fitting the data to a curve; hence, it has a lower ability to detect true positives when the genes are not continuously changing in time, but only significant at few time points. Since

our VSP method takes into consideration the possibility of correlated errors, its superior performance is not affected by the type of noise added.

4.2.7 Results Using Real Dataset

The proposed VSP method is now applied on the real dataset that has been used to study the estrogen effect on breast cancer in [3] and [39]. Our method yields 55 significant genes with a threshold p-value fixed at 10^{-5}, which corresponds to a maximum FP number of 0.019 (1900×10^{-5}) genes. The most significant groups from the 55 genes are that of the cell cycle (13 genes), transcription and chromatin structure (9 genes), DNA replication and repair (6 genes) and cellular signaling (6 genes). The cell cycle genes are CCNA2, MAD2L1, DTYMK, AURKA, CKS2, CDKN3, TUBG1, DTYMK, CDKN1A, CSE1L, CTSD, CDC20 and AURKB, the transcription and chromatin structure genes are ELF3, MYBL2, HMGB1, MYC, STAT1, TRIP13, SNRPB, CSNK1A1 and NR1D2, the cellular signaling genes are PRKCD, NPY1R, HSPB8, ERBB2, AKAP1, and PRKAR1A, and the DNA replication and repair genes are FEN1, DHFR, POLD1, POLE, LIG1 and RFC3.

Among the 55 genes, 50 genes have been previously identified as significant. There are 28 genes identified in both [3] and [39]. There are 44 genes identified significant in [39]. Moreover, 34 genes are found within the first 50 genes identified by the Oriogen method [3]. Our VSP method has thus identified 5 new genes that have not been identified by other methods. These are IRF1, CSNK1A1 (identified from two independent clones, clone ids: 381589, 510319), CYP27A1, NR1D2 and RBM3.

IRF1 serves as an activator of interferons alpha and beta transcription. It functions as a transcription activator of genes induced by interferons alpha, beta, and gamma. Furthermore, IRF1 has been shown to play a role in regulating apoptosis and tumor-suppression [42, 43]; it is shown therein that it is directly related to apoptosis in breast cancer cells and affected by the 17ß-Estradiol. CSNK1A1 interacts with HMGB1 and plays a role in transcription and chromatin structure [44, 45]. In addition, NR1D2 encodes the NR1 subfamily of hormone receptors and also is involved in transcription activity [46]. CYP27A1 encodes a member of the cytochrome P450 superfamily of

enzymes. It is directly affected by applying estrogen to the sample cells as found in [47]. RBM3 gene is a member of the glycine-rich RNA-binding protein family and encodes a protein with one RNA recognition motif (RRM) domain. It plays a role in apoptosis [48] in breast cancer cells.

EDGE [4] is also applied to this real dataset. At threshold p-value fixed at 10^{-5}, 2×10^{-5} and 3×10^{-5}, EDGE yields 33, 51 and 60 genes, respectively. A summary of the genes identified by EDGE and the proposed VSP method, compared to other methods is given in Table 4-9.

Table 4-9 Summary of the genes identified by the proposed VSP method and EDGE method

P-value threshold=10^{-5}	
Proposed VSP Method	**EDGE Method**
Number of genes identified =**55**	Number of genes identified =**33**
• 50 of the 55 genes identified by VSP have also been identified as significant in [3] and/or [39].	• 26 of the 33 genes identified by EDGE have also been identified by [3] and/or [39] and are among the top 55 genes identified by VSP.
• The VSP method has identified 5 new genes, not identified by [3] or [39]. These are IRF1, CSNK1A1 (identified from two independent clones), CYP27A1, NR1D2 and RBM3 and are consistent with biological literature.	• The gene CSNK1A1 identified by the VSP method has also been identified by EDGE.
	• EDGE has identified 5 genes that are not among the top genes identified by any of the methods [3], [39] or the proposed VSP method. These are PRDX4, LMNB2, ITGA6, SFRS3 and ATF4.
	• The gene RAD51 identified by EDGE is one of the top 63 genes identified by VSP method.
P-value threshold=2×10^{-5}	
Number of genes identified =**63**	Number of genes identified =**51**
• In addition to the above mentioned 55 genes, there are 8 genes that have been identified.	• In addition to the above mentioned 33 genes, there are 18 genes that have been identified.
• 3 of these 8 genes have been identified as significant in [3] and/or [39].	• 5 of these 18 genes have also been identified by [3] and/or [39] and are among the top 55 genes identified by VSP method.
• The gene RAD51, identified by EDGE, is also identified by the VSP method.	• The new genes CYP27A1, NR1D2 and RBM3, identified by the VSP method, are also identified by EDGE.
	• The gene MCM3 identified by EDGE has also been identified by [3] and [39] and is one of the top 71 genes identified by VSP method.

	• The gene CD55 identified by EDGE, is one of the top 71 genes identified by VSP method.
• The remaining 4 new genes out of the 8 genes are ULK3, RAP1, VEGFA and XIST.	• EDGE identified 8 genes that are not among the top genes identified by any of the methods of [3], [39] or VSP. These are ILF2, GFM2, CDK2, MXD4, CCNI, RARP1, RELA and RAB2B
P-value threshold=3×10^{-5}	
Number of genes identified =**71**	Number of genes identified =**60**
• In addition to the above mentioned 63 genes, there are 8 genes that have been identified.	• In addition to the above mentioned 51 genes, there are 9 genes that have been identified.
• 4 of these 8 genes has been identified as significant in [3] and/or [39].	• 2 of these 9 genes have been identified by [3] and/or [39] and are as among the top 55 genes in the proposed method.
	• The second clone of the gene CSNK1A1, identified by EDGE is one of the top 55 genes identified by the VSP method.
• The other 4 genes identified are PTPRF, STAT1, CD55 and DDIT3.	• The gene PTPRF is also one of the 9 genes identified by EDGE.
	• EDGE identified 5 genes that are not among the top genes in any of the methods [3], [39] or the proposed one. These are CYP4Z1, IGF2R, CDC25B, EFNA5 and TOP2A.

In conclusion, all the genes identified by EDGE and at the same time identified by [3] and [39] are also identified by the proposed VSP method. There are other common genes between EDGE and the VSP method that were not identified by [3] and [39]. In addition to these genes, EDGE as well as the proposed VSP method identified new genes that are not identified by previous methods.

To further explore the genes identified by our VSP method, hierarchical clustering is applied where correlation is used as the distance measure between the expression profiles. In order to determine the number of clusters, the average silhouette [49] is used, where the number of clusters k is selected by maximizing the average silhouette over a range of possible values for k. For the real dataset considered here, the maximum value of the average silhouette index is 0.85, when the number of clusters equals 2. The 2 clusters simply divide the expression profiles into the upregulated (36 genes) and downregulated (19 genes) genes, as shown in Figure 4.14. Although clustering the genes into a large number of clusters provides us with less optimum average

silhouettes, it gives more details about the gene expressions. For instance, Figure 4.15 and Figure 4.16 show the gene expressions for 3 and 6 clusters, respectively.

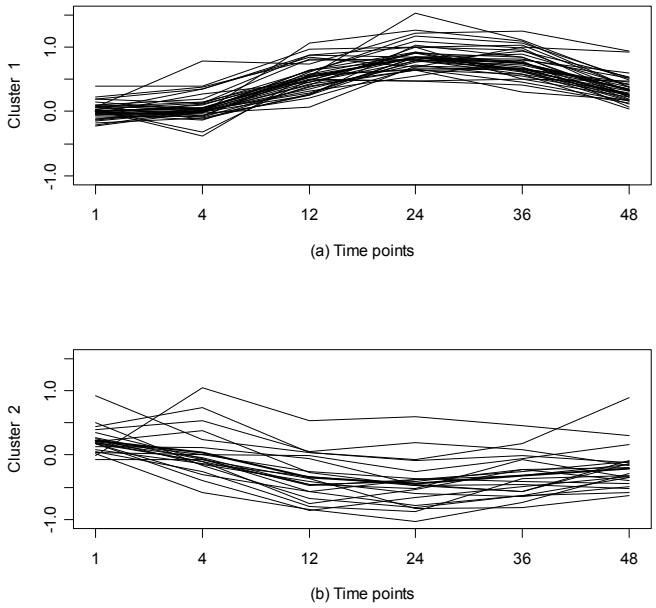

Figure 4.14 (a) Upregulated genes. (b) Downregulated genes.

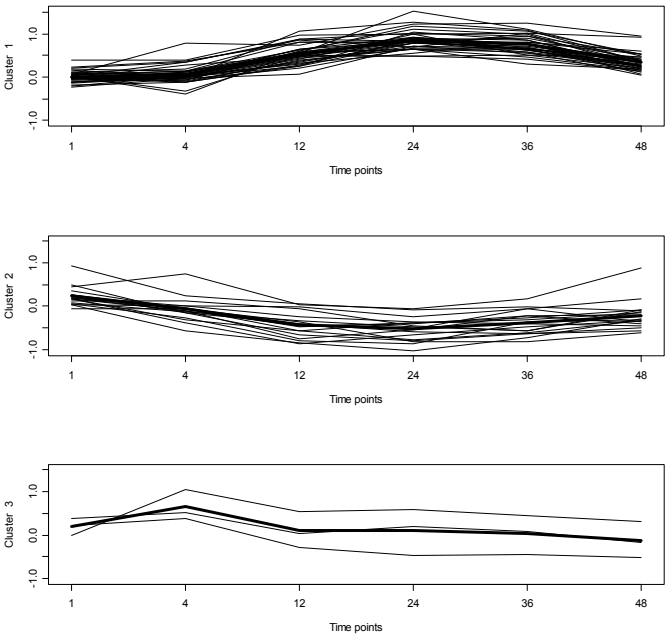

Figure 4.15 Gene Expressions for 3 clusters.

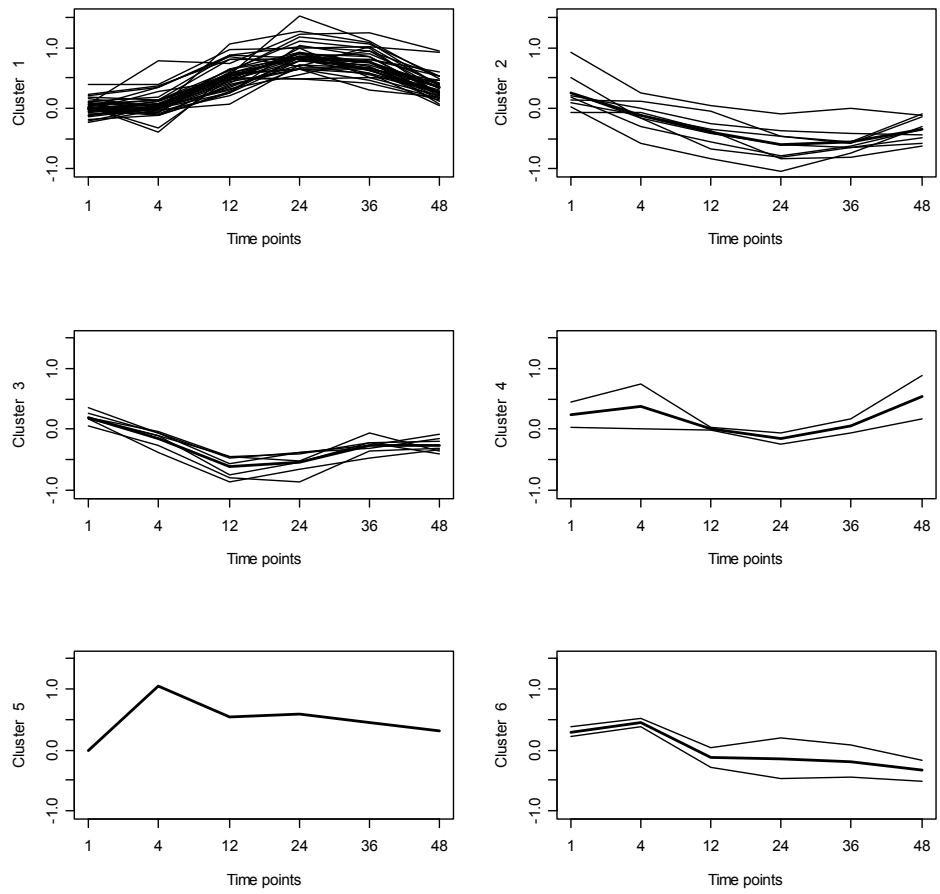

Figure 4.16 Gene Expressions for 6 clusters

As seen from Figure 4.14 to Figure 4.16, our proposed VSP technique is able to detect differentially-expressed genes. Thus, the investigation on the real data set clearly shows that our proposed technique is able to identify some significant genes that have been missed by other techniques.

In addition, in order to show that the proposed method along with the model used is a valid model for real microarray data, the residual errors of some of the genes found in [39] are plotted against the fitted values and shown in Figure 4.17 and Figure 4.18. The scatter plot should be symmetric vertically around 0. If nonlinearity occurs then the model could be inappropriate. However as shown in Figure 4.17 and Figure 4.18, the residual errors are distributed symmetrically around the

zero axis which indicates that the model employed in our method is a suitable model for the microarray data.

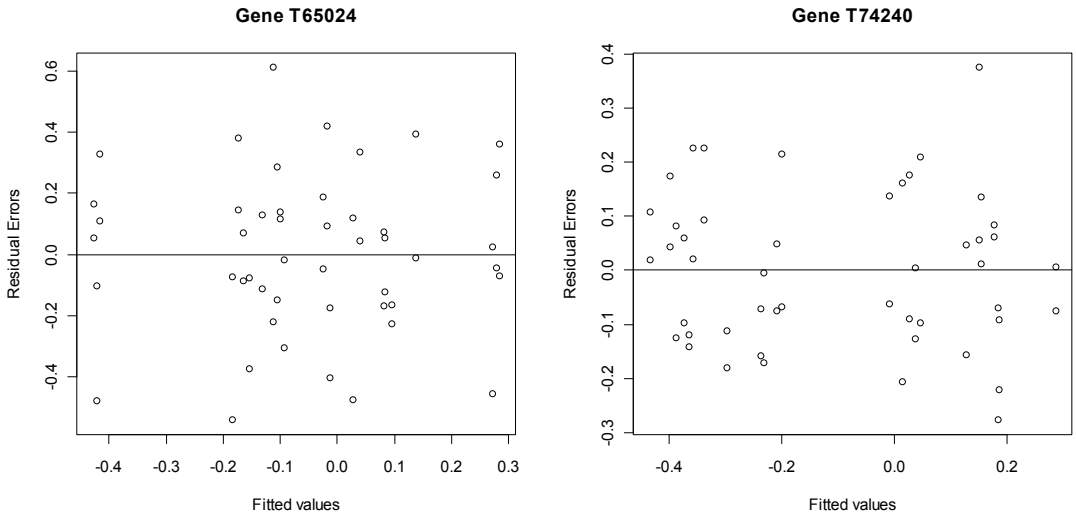

Figure 4.17 Scatter plot of residual errors for non-differeentially expressed genes

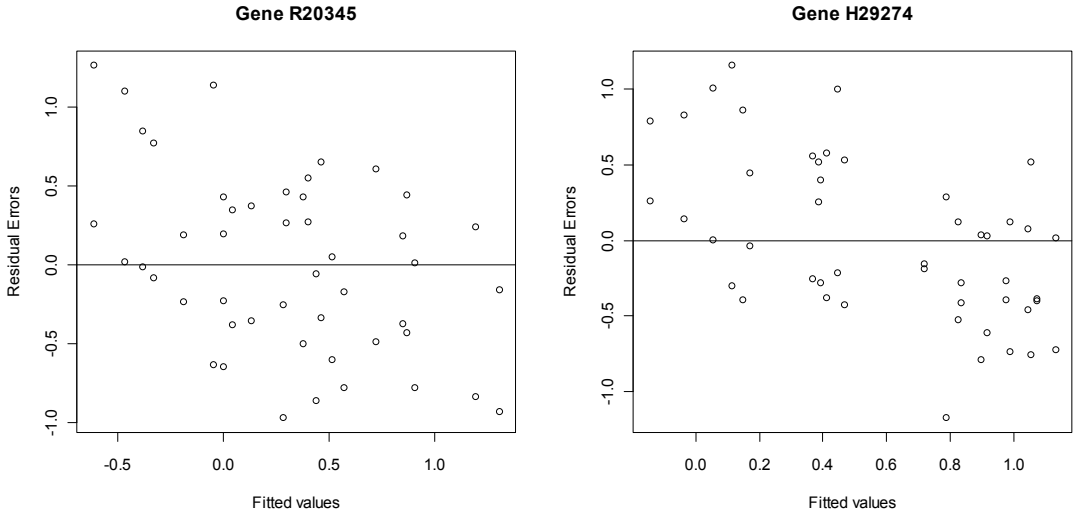

Figure 4.18 Scatter plot of residual errors for differeentially expressed genes

4.3 Results for Multiple Time-course Data

We now apply our VSP method to multiple time course-data, and compare our results with those of the previous methods.

4.3.1 Performance Comparison of the Proposed Method with Existing Moderation Techniques

The performance of the proposed VSP method is compared to that of the 90th quantile of Efron *et al.* [20], the empirical Bayes correction factor method of Smyth [22] and the variance shrinkage method of Cui *et al.* [23]. Figure 4.19 shows the TP and FP values as functions of the p-value for the various correction techniques, as well as for the original statistic without any correction. In order to make a fair comparison, the proposed correction method as well as the techniques of Efron *et al.*, Smyth and Cui *et al.* are applied to the same F-statistic. Furthermore, the sensitivity and specificity of these methods are compared at a significance level of p-value = 0.001. The average results over 300 datasets are shown in Table 4-10.

Table 4-10 Sensitivity and Specificity for the moderation methods

Moderation Method	Sensitivity	Specificity
No correction	25.94%	99.91%
Proposed VSP method	44.408%	99.99%
Efron *et al.* [20]	41.246%	99.99%
Smyth [22]	43.052%	99.99%
Cui *et al.* [23]	41.28%	100%

As seen from Figure 4-19, the specificity of any of the techniques is better than that of the ordinary statistic. As seen from Table 4-10, Cui's technique has the lowest FP rate, resulting in very good specificity; in spite of this, this technique has low sensitivity. Efron's method as well as that of Smyth has better sensitivity than the original RM F-statistic without correction and a small FP rate. As seen from Figure 4.19 and Table 4-10, the proposed VSP method is the most powerful, in the sense that it has the best results in terms of the TP rate, while maintaining a satisfactory FP rate. Although all the correction methods improve the specificity, the proposed correction technique has the best sensitivity.

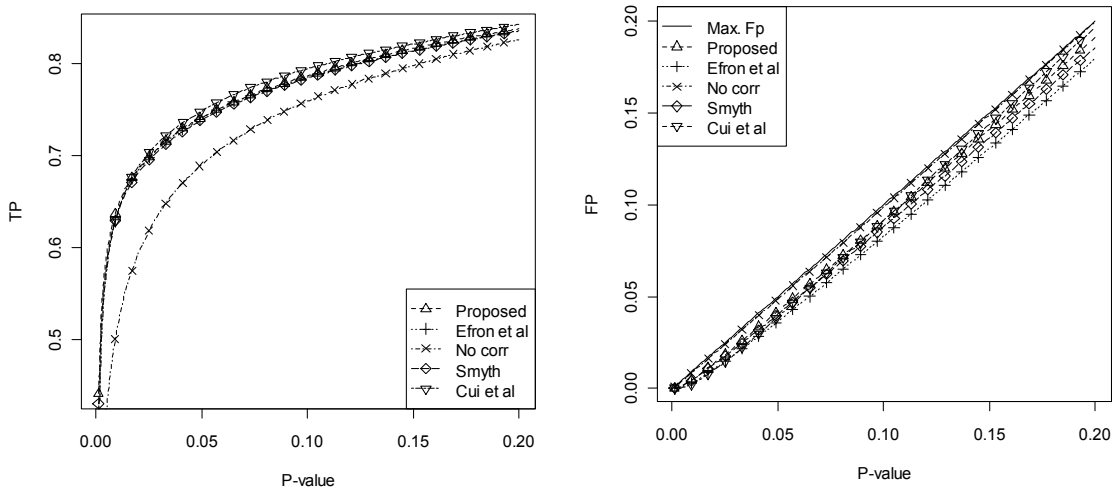

Figure 4.19 TP and FP for the different moderation techniques

4.3.2 Performance Comparison of the Proposed Method with the Existing Time-series Methods

Our proposed algorithm is compared to the existing techniques for time-series microarray data. The genes identified as significant using our VSP method are compared to those identified by EDGE [4] and SAM [16] methods.

The TP and FP values as functions of the p-value are shown in Figure 4.20. The sensitivity and specificity of these methods are compared at a significance level of p-value = 0.001, and the average results over 300 datasets are shown in Table 4-11.

Table 4-11 Sensitivity and Specificity for the Time-series Methods

Time-series Method	Sensitivity	Specificity
Proposed VSP method	44.408%	99.99%
EDGE [4]	42.12%	99.87%
SAM(slope) [16]	4.27%	100%
SAM(area) [16]	4.41%	100%

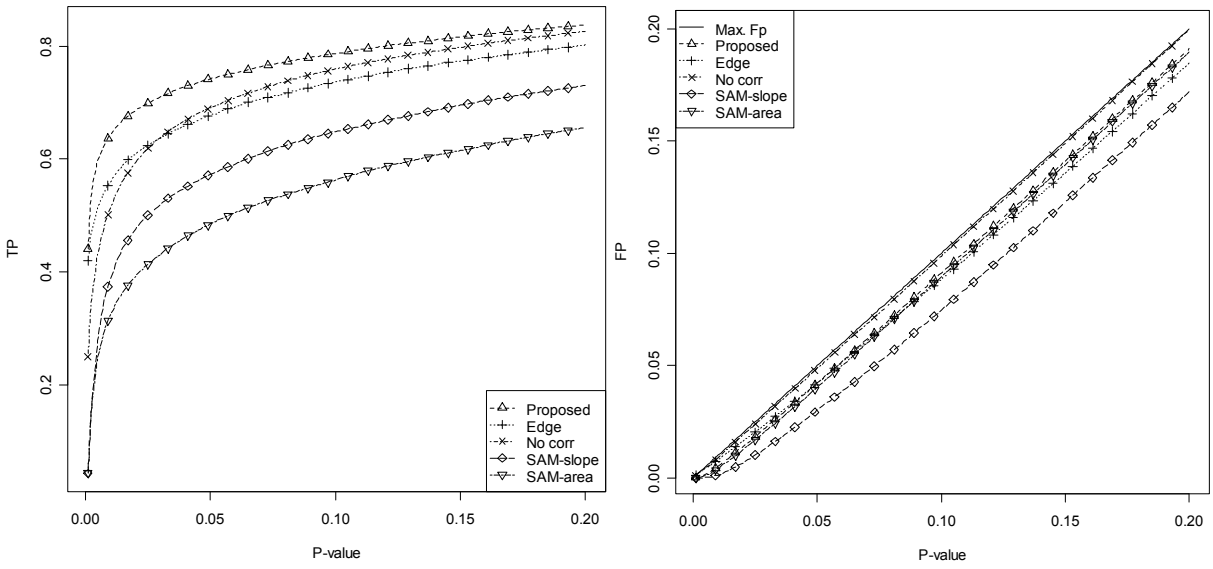

Figure 4.20 TP and FP for several time-series methods

As seen from Figure 4.20 and Table 4-11, all the techniques have FP values that are below the maximum allowable FP value; however, the proposed method outperforms the EDGE method in terms of the specificity and surpasses all the methods in terms of the sensitivity.

4.3.3 Results Using Real Dataset

The proposed VSP method is now applied on a real dataset that has been used to study the transcriptional response to Cold in potato [41]. The dataset used has 2 series, control series, and a cold stress series, where the control series is a reference microarray data that is not subjected to a stress condition. Each series consists of 3 time points, harvested at 3, 9 and 27 hours and 3 replicated samples at each time point.

Our method yields 92 significant genes with a threshold p-value fixed at 10^{-5}, which corresponds to a maximum FP number of 0.118 (11874×10^{-5}) genes. The most significant groups from the 92 genes are that of the Transcriptional regulation STMFB31, STMIJ20, STMIJ20, Enzymatic activity STMIR14, STMHE15, STMCN30, STMEJ12, STMEJ12, Signal transduction STMHS17, STMGX18, Hormone related STMIY82 and Transport STMCB83. There is only limited genomic information available for potato, which complicates cross-species comparisons. A large number of

clones represent genes with unknown function; these genes could provide a basis for the discovery of novel stress related proteins. Among the 92 genes, 90 genes have been previously identified as significant in [41].

In order to further explore the genes identified by our VSP method, the two genes identified by our approach as significant and missed by [41] are shown in Figure 4.21. Some of the genes identified as significant are shown in Figure 4.22, where blue is the control and black is when the genes are affected by cold stress. As seen from these figures, our proposed VSP technique is able to detect differentially-expressed genes, and the investigation on the real data set clearly shows that our proposed technique is able to identify some significant genes that have been missed by other techniques.

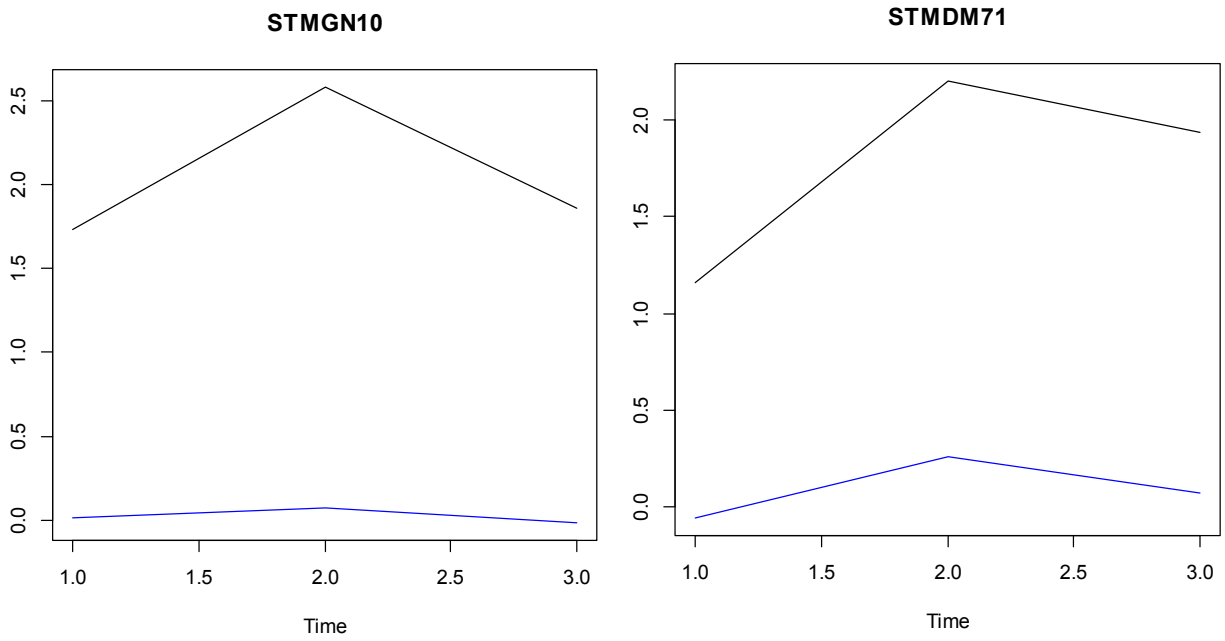

Figure 4.21 Genes expressions of the the two siginficant genes missed by other techniques

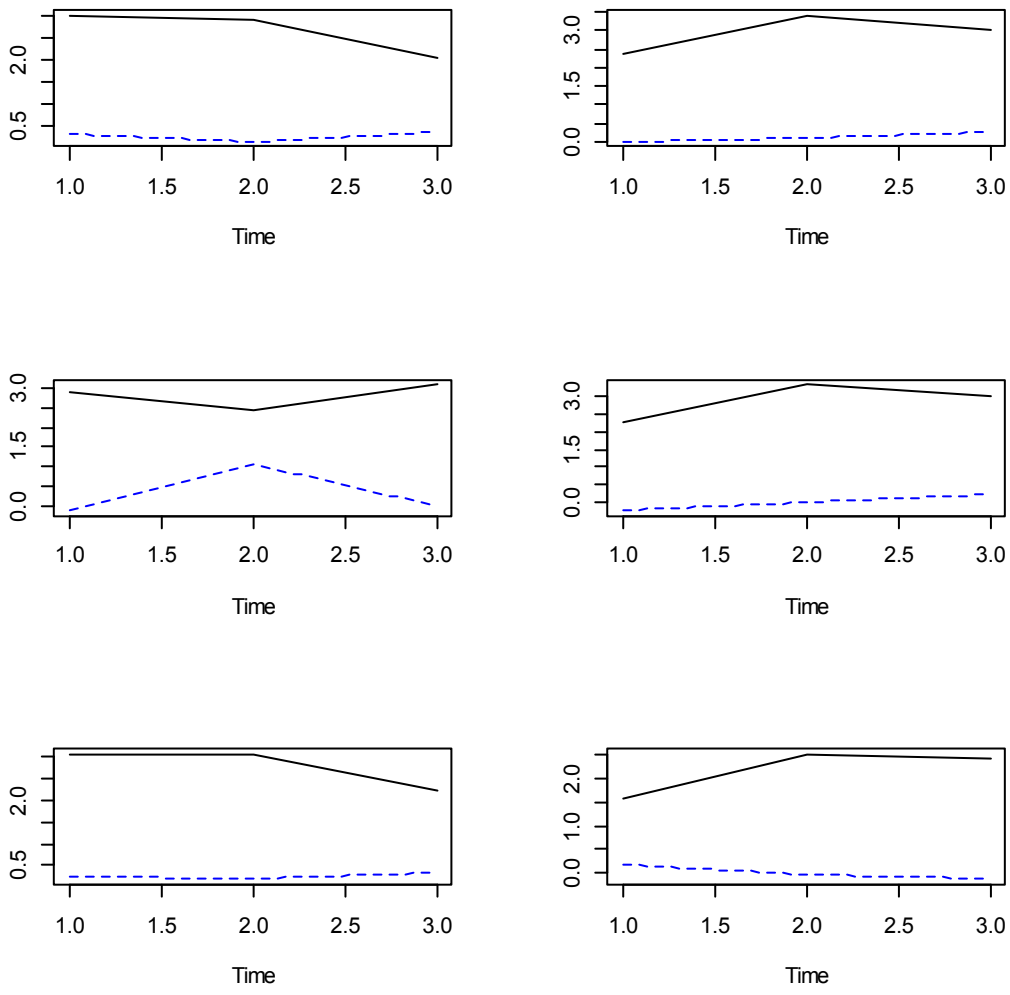

Figure 4.22 Gene Expressions of significant genes where cold stress are solid lines, while control are dashed lines.

4.4 Summary

In this chapter, we have applied the proposed VSP method on synthetic and real data for both the single and multiple biological groups. The experimental results on the synthetic data have shown that the pooled p-values procedure is able to detect more true positives than the gene-wise p-values method does. The proposed moderation factor has been shown to outperform the other moderation techniques in terms of the sensitivity. Furthermore, the proposed algorithm outperforms the existing

time-series analysis techniques in terms of both the sensitivity and the specificity. In addition, the proposed algorithm when applied to real data has been shown to detect those genes identified as significant by previous techniques, and to identify other significant genes consistent with existing biological knowledge.

Chapter 5

Reconstruction of Gene Regulatory Network

The second question of how these genes interact is answered with the help of gene network reconstruction [50]. In the network reconstruction, we are mainly concerned with the dynamic system of equation models as it describes the network dynamics for a time series data successfully. In this work, GRN networks are represented by linear system of equations, where the lag of each gene is distinguished from that of the other genes, and gene dependency networks are reconstructed. Moreover, an approach based on pair-wise correlations and lasso to infer the variably delayed GRN is presented. Our proposed technique takes into account the variable time delays between various genes.

The goal of network inference is to detect the most likely interactions by identifying sets of relevant model parameters to obtain a suitable correspondence between measured data and model output. A genetic network is inferred by learning a mathematical model to predict future gene expression values based on past gene expression values. Intensity values of samples are usually averaged to reduce the complexity of the data set.

System modeling includes two key parts: the network structure (i.e. the interactions between the components) and the model parameters (e.g. type/strengths of these interactions). Several algorithms determine both the network structure and the parameters simultaneously [28] while others determine either the network structure [51, 52] or the parameters [53].

The first subsection is concerned with the reconstruction of network structure only, and then the next subsections address the first approach where both the network structure and the parameters are determined.

5.1 Gene Dependency Networks

If the network structure is unknown, statistical approaches such as graphical models are used to estimate genetic networks. A graphical model is a representation of stochastic conditional dependencies between the investigated variables. The basic idea to infer a network from the

correlation is to refer to the genes as the nodes and to the correlations as the connectivity strengths assigned to the edges of the network. However, the correlation coefficients cannot be used directly, because they represent only marginal dependencies and also include indirect interactions between two variables. Instead, we need to rely on the concept of partial correlation which describes the correlation between any two variables *i* and *j* conditioned on all the other variables.

5.1.1 Partial Correlation

The purpose of partial correlation is to measure the degree of association between two random variables, with the effect of a set of controlling random variables removed. The control variables in partial correlation are the variables which extract the variance which is obtained from the initial correlated variables. Formally, the partial correlation between X_i and X_j given a set of n controlling variables $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_n\}$, written $\rho_{X_iX_j \cdot \mathbf{Z}}$, is the correlation between the residuals R_{Xi} and R_{Xj} resulting from the linear regression of X_i with $\mathbf{Z}$ and of X_j with $\mathbf{Z}$, respectively. The order of correlation in partial correlation refers to the correlation with control variables. For example, first order partial correlation is the one which has a single control variable. The zeroth-order partial correlation $\rho_{XY \cdot \emptyset}$ is defined to be the regular correlation coefficient $\rho_{X_iX_j}$. The first-order partial correlation is the difference between a correlation and the product of the removable correlations divided by the product of the coefficients of alienation of the removable correlations. When Z is a single variable, the first order partial correlation is given by

$$\rho_{X_iX_j \cdot Z} = \frac{\rho_{X_iX_j} - \rho_{X_iZ}\rho_{X_jZ}}{\sqrt{1-\rho_{X_iZ}^2}\sqrt{1-\rho_{X_jZ}^2}} \tag{5-1}$$

The *n*th-order partial correlation (i.e., with $|\mathbf{Z}| = n$) can be computed from the $(n - 1)$th-order partial correlations. Naïvely implementing this computation as a recursive algorithm yields an exponential time complexity. However, the overlapping sub-problems improves the computational time when using techniques such as dynamic programming.

Instead of using an iterative approach, all partial correlations between any two variables X_i and X_j of a set $\mathbf{V}$ of cardinality n given all others can be computed using the inverse of the correlation

matrix Ω. Given the correlation matrix Ω = (ω_{ij}), where $\omega_{ij} = \rho_{XiXj}$, is invertible and P = Ω^{-1}, the partial correlations are given by

$$\rho_{X_i X_j . V \setminus \{X_i, X_j\}} = \frac{-p_{ij}}{\sqrt{p_{ii} p_{jj}}} \tag{5-2}$$

Correlation and partial correlation should not be confused with causality, since many different causal relationships can correlate the same pair of variables. Although the correlation networks are not the same as the underlying causal networks, correlation is still informative about the underlying system. Although partial correlation analysis does not infer causal relationships, it excludes many of the possibilities, and thus is a step in the direction of causal inference. The strength of these coefficients indicates the presence or absence of a direct association between each pair of genes.

It is typical to conduct partial correlation when the third variable has shown a relationship to one or both of the primary variables. At first correlational analysis on all variables is conducted in order to determine whether there are significant relationships amongst the variables, including any "third variables" that may have a significant relationship to the variables under investigation. The spurious correlation in partial correlation refers to that type of correlation that is false or the correlation that actually does not exist. Partial correlation is generally helpful in detecting false relationships. Partial Correlation is used in models that assume a linear relationship.

5.1.2 The Graphical Model for Gene Dependency Networks

Let graph G = (V, E) be an undirected graph with vertices V = {1, 2… *p*}, where *p* is the number of genes, vertex V_i represent gene *i,* and the edges represent the association between the different genes. A high value of partial correlation between two genes will correspond to an edge between the two genes, while values approaching zero will correspond to conditional independence and no edge is found between the two genes. In the graphical model, only the direct interactions are drawn corresponding to edges between genes.

5.2 Gene Regulatory Network Model

Successful GRN reconstruction includes solving three problems: estimating the different time

delays between various genes, which genes are actually related to each other, and finally the parameter values for the related genes. Usually one or two of these problems is ignored to simplify the calculations. If the three problems are solved simultaneously, the number of unknown parameters grows rapidly, and hence, the results show poor performance. Herein, we introduce a two-step approach that can solve the three problems with an improved efficiency. In this work, pair-wise correlations along with lasso are applied to estimate variably delayed GRN. Although correlations have been previously used in the estimation of dependency networks between genes [7, 8], it has not been used to evaluate the correct time delays between different genes. After estimating the time delays for the candidate genes, lasso is used to differentiate between direct and indirect relations between various genes and the final GRN is inferred.

One of the simplest models is a linear genetic network model. The linear model assumes that the gene expression level of each gene is the result of a weighted sum of all other gene expression levels at the previous time point. The proposed model is linear and can be represented by a vector autoregressive (VAR) model. For given gene expression profiles of p genes, T time points and n number of samples, they can be represented by a VAR model of order d .The vector form is given by

$$X_t = B_{t-1} X_{t-1} + \cdots + B_{t-d} X_{t-d} + \varepsilon, \tag{5-3}$$

where $X_t = (x_{1t}, x_{2t}, \ldots, x_{pt})^T$, B_{t-j} is a $p \times p$ coefficient matrix at time difference j and ε is a zero mean, white noise process. The components $\beta_{ik}^{(t-j)}$ are of the interaction matrix B_{t-j} that describes the gene expression kinetics. In this model, a gene at time t is potentially regulated by the genes at previous time points $t-1$, $t-2$, .., $t-d$. It is cumbersome to solve this model directly, due to the limited number of samples n and time points T, while the number of covariates to be solved will be $p \times d$. Instead, we first evaluate the potential delay between every two genes regardless of their actual interactions. Thus, the delay estimation will reduce the number of covariates to p covariates. Then, the direct interactions among genes are evaluated by a lasso procedure.

5.2.1 The Graphical Model

Let graph G = (V, E) be a directed graph with vertices V = {1, 2… p}, where p is the number of genes, vertex V_i represent gene *i,* and the directed edges represent the direct association between the different genes. The direction of the edge represent the causal effect between the two genes, for instance, the edge E_{ij} means that gene i affects gene j. Correlation does not imply causality, but instead the proposed model can be interpreted in terms of Granger causality [54]. A time series x_i is considered a granger cause of x_j if the knowledge of past values of x_i improves the prediction of x_j, compared to only using the past values of x_j.

5.2.2 Time Delay Estimation

Since the gene expression values in a time series experiment can be considered as a short time-series signal, discrete signal processing techniques such as cross-correlation can be applied. It is assumed that the data are observed at integer time points and the lag between two observations *x(t)* and *x(t-τ)* is given by τ. In order to estimate the time delays between various genes, pair-wise correlations are employed between each two genes, for various time delays. The time delay at which the correlation is maximum, is the estimated delay between these two genes. The concept of having the delay estimate at the maximum correlation between two genes, is widely established in signal processing field, as it is shown in [55] that this estimate is the maximum likelihood (ML) estimate of the delay. Thus, the problem of estimating the delay between two genes can be approached using the maximum likelihood theorem [55]. In [56] the effect of noise and other various factors on the delay estimate are studied.

First, we define the autocorrelation function $R_{ss}(\tau)$ as the set of correlation coefficients between the time series *s(t)* and lags of itself over time *s(t-τ)*.

$$R_{ss}(\tau) = E[s(t)\, s(t+\tau)] \tag{5-4}$$

The maximum correlation between the signal and itself occurs when there is no delay between them. The more the signal *s(t-τ)* is shifted away from the original signal *s(t)* the less the autocorrelation value will be, as shown in Figure 5.1.

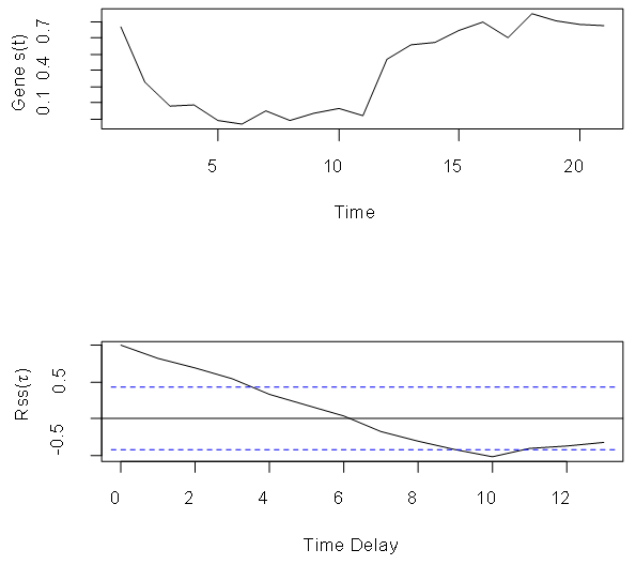

Figure 5.1 Autocorrelation of a signal *s(t)*

Similarly, when we have a shifted signal $s(t-\tau)$, and compare it to the signal *s(t)*, the time delay τ at which the correlation is maximum, is considered to be the delay between the two signals as shown in Figure 5.2, where the signal is shifted by 3 and the maximum $R_{ss}(\tau)$ is at 3.

If two genes are correlated at a certain delay, we assume they will take the simple form [57]:

$$x_1(t)=s(t)+n_1(t), \tag{5-5}$$

$$x_2(t)= a\times s(t-D)+n_2(t), \tag{5-6}$$

It is assumed that $s(t)$, $n_1(t)$, and $n_2(t)$ are independent and stationary processes , and a is a nonzero constant. The pairwise correlation between $x_1(t)$ and $x_2(t)$ is defined as

$$R_{x_1x_2}(\tau)=E[x_1(t)x_2(t+\tau)]= E[(s(t)+n_1(t))(a\times s\ (t+\tau-D)+n(t+\tau))]$$

$$=a\times E[s(t)s(t+\tau-D)]= a\times R_{ss}(\tau - D), \tag{5-7}$$

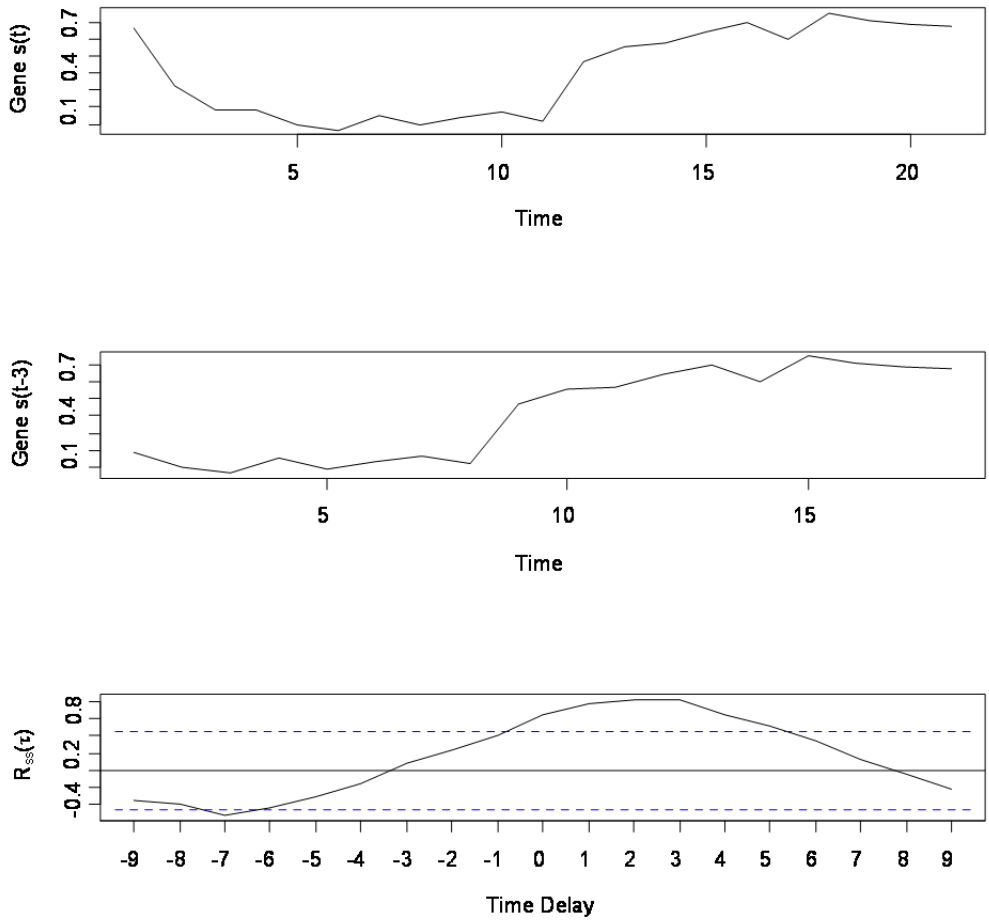

Figure 5.2 Cross-correlation between the two signals *s(t)* and *s(t-3)*

Since the maximum R_{ss} is at 0, the peak value occurs at τ=*D*. For continuous data $x_1(t)$ and $x_2(t)$, which exist over interval *T*, the sample cross correlation estimate is given by[57]:

$$\hat{R}_{x_1x_2}(\tau) = \frac{1}{T}\int_0^T x_1(t)x_2(t+\tau)dt \ , \quad 0 \leq \tau < T \tag{5-8}$$

The expected value of the estimate $\hat{R}_{x_1x_2}(\tau)$ is given by

$$E[\hat{R}_{x_1x_2}(\tau)] = \frac{1}{T}\int_0^T E[x_1(t)x_2(t+\tau)]dt = \frac{1}{T}\int_0^T R_{x_1x_2}(\tau)dt = R_{x_1x_2}(\tau) \tag{5-9}$$

Hence, $\hat{R}_{x_1x_2}(\tau)$ is an unbiased estimate of $R_{x_1x_2}(\tau)$. In practice, only the observations of *s(t)* and *s(t-D)*, the discrete signals $x_1(k)$ and $x_2(k)$, are available. The gene expressions $x_1(k)$ and $x_2(k)$ are

noisy and sample length limited, hence an approximate value of $R_{x_1x_2}(\tau)$ is computed. The normalized correlation coefficient, $r_{x_1x_2}(\tau)$ is given by

$$r_{x_1x_2}(\tau)=\frac{\sum_{k=1}^{T}(x_1(k+\tau)-\bar{x}_1)(x_2(k)-\bar{x}_2)}{\sqrt{\sum_{k=1}^{T}(x_1(k)-\bar{x}_1)^2}\sqrt{\sum_{k=1}^{T}(x_2(k)-\bar{x}_2)^2}}, \tag{5-10}$$

where $\bar{x}_i=\frac{1}{T}\sum_{k=1}^{T}x_i(k)$, T is the total number of time points, τ is the time lag between the two genes x_1 and x_2 and $1 \le r \le 1$. Since the estimates of the cross-correlation are being made using discrete information; there are errors in the estimates. The peaks are more sensitive to errors introduced by the finite observation time, especially in cases of low SNR. In microarray data, there are usually replicated samples, hence, the average values for each gene expression are first calculated, and then, the pairwise correlations are evaluated. In literature, there are other approaches that use replicates to estimate the pair-wise correlations, such as, the standard deviation (SD)-weighted correlation coefficient [58], and is defined by

$$r_{x_1x_2}(\tau)_{SD}=\frac{\sum_{k=1}^{T}\left(\frac{x_1(k+\tau)-\bar{x}_1}{S_1(k)}\right)\left(\frac{x_2(k)-\bar{x}_2}{S_2(k)}\right)}{\sqrt{\sum_{k=1}^{T}\left(\frac{x_1(k)-\bar{x}_1}{S_1(k)}\right)^2}\sqrt{\sum_{k=1}^{T}\left(\frac{x_2(k)-\bar{x}_2}{S_2(k)}\right)^2}} \tag{5-11}$$

where

$$x_1(k)=\frac{1}{n}\sum_{i=1}^{n}x_{1i}(k)\ ,\ x_2(k)=\frac{1}{n}\sum_{i=1}^{n}x_{2i}(k) \tag{5-12}$$

$$S_1^2(k)=\frac{1}{n-1}\sum_{i=1}^{n}(x_{1i}(k)-x_1(k))^2\ ,\ S_2^2(k)=\frac{1}{n-1}\sum_{i=1}^{n}(x_{2i}(k)-x_2(k))^2 \tag{5-13}$$

However, this type of coefficient will slightly improve Pearson correlation when the number of samples is very large which is not common in microarray data. In addition, there is a multivariate correlation estimator [59] that estimates correlation from replicated samples and is implemented in

a library package called "CORREP", where it assumes equal number of replicates. However, Pearson correlation has the best results and thus, it is used in our proposed method. Correlation here is used solely to determine the optimum delay between any two genes regardless of whether they have direct or indirect relationship. In order to determine the model structure, and which genes are regulators, lasso is employed.

5.2.3 Model Structure and Parameter Reconstruction

Linear genetic network model is one of the common models in GRN reconstruction. The linear model assumes that the gene expression level of each gene is the result of a weighted sum of all other gene expression levels at previous time points. The current model is given by

$$X_t = \sum_{j=1}^{d} B_{t-j} X_{t-j} + \varepsilon, \qquad (5\text{-}14)$$

where B_{t-j} is the adjacency matrix describing the gene expression kinetics at time difference j, whose elements are $\beta_{ik}^{(t-j)}$. The element $\beta_{ik}^{(t-j)}$ represents the existence ($\beta_{ik}^{(t-j)} \neq 0$) or non-existence ($\beta_{ik}^{(t-j)}=0$) of an action of the regulator gene k on the target gene i. Further, ($\beta_{ik}^{(t-j)}> 0$) or ($\beta_{ik}^{(t-j)}< 0$) indicates as to whether the gene is activating or inhibiting.

Since in microarray data, there could be replicate experiments and hence replicated time series, the replicated time series are arranged to form a single time series that can be used to fit the linear model as in (5-14). The time series data is arranged such that for each gene the replicates at each time point are ordered followed by the replicates at the second time point and so on. Hence, the final number of samples is $N=n\times T$, instead of n only. Since the total number of data points is larger, more accurate results are expected. This type of data arrangement is similar to that found in [60]. Bowden *et al.* [60] have shown that this type of arrangement, which they called as interleaved time series, is successful for modeling replicate time series in a single AR model with the only constraint that missing data have to be imputed as all the time series must be of the same length.

If there are enough time points, (5-14) can be solved using the ordinary least squares where an error term is to be minimized. The standard error term would be the squared error between the predicted and measured gene expression levels, and the ordinary least square (OLS) estimate for the

coefficients is given by

$$\hat{\beta}_{OLS} = \arg_{\beta} min \left\| \mathrm{X}_t - \sum_{j=1}^{d} B_{t-\mathrm{j}} \, \mathrm{X}_{t-j} \right\|^2 \qquad (5\text{-}15)$$

However, the resulting network will not be sparse, whereas it is well known in literature that GRNs are sparse and each gene is regulated by a few number of genes. In addition, the number of time points is far less than the number of selected genes.

Time-series microarray data is a high dimensional data where the number of covariates, *p*, exceeds the number of samples and time points, resulting in a highly under-determined problem, therefore, additional constraints are required to successfully reconstruct the underlying model. Restricting the number of gene regulators is biologically reasonable. This is due to the fact that only a limited number of genes will directly influence a gene's transcription during any biological process. Since a gene is usually regulated by only a few genes, sparsity constraint is imposed. Ideally sparsity constraint is equivalent to minimizing the ℓ_0 norm, which is the number of nonzero covariates. However, such a minimization problem is not a convex problem and not easily solved. Instead, a relaxation for the ℓ_0 norm is applied. In literature, a tractable approach is to minimize the ℓ_1 norm instead, which is the absolute values of coefficients, and is widely known as lasso [28] and its variants.

The Least Absolute Shrinkage and Selection Operator (LASSO) technique tends to shrink the weights such that only a few weights remain non-zero. A penalty term, that sums the absolute values of the weights, would be added to the standard squared error. The lasso estimate for the coefficients is given by

$$\hat{\beta} = \arg_{\beta} min \left\| \mathrm{X}_t - \sum_{j=1}^{d} B_{t-\mathrm{j}} \, \mathrm{X}_{t-j} \right\|^2 + \lambda \sum_{j=1}^{d} \sum_{i=1}^{p} \sum_{k=1}^{p} \left| \beta_{ik}^{(t-j)} \right| , \qquad (5\text{-}16)$$

where λ is a non-negative lasso regularization parameter. Since each gene has a separate model, (5-16) can be solved as separate *p* models, and can be rewritten for each gene *i* as following:

$$\widehat{\beta_{\iota}} = \arg_{\beta} min \left\| x_{i,t} - \sum_{j=1}^{d} b_{i,(t-\mathrm{j})} \, \mathrm{X}_{t-j} \right\|^2 + \lambda \sum_{j=1}^{d} \sum_{k=1}^{p} \left| \beta_{ik}^{(t-j)} \right|, \qquad (5\text{-}17)$$

or,

$$\widehat{\beta_\iota} = \arg min \left\{ \sum_{t=d+1}^{T} (x_{i,t} - \sum_{j=1}^{d} b_{i,(t-\mathrm{j})}^{T} \mathrm{X}_{t-j})^2 + \lambda \sum_{j=1}^{d} \sum_{k=1}^{p} \left| \beta_{i,k}^{(t-j)} \right| \right\}, \quad (5\text{-}18)$$

where $b_{i,(t-j)}$ is a row vector of the matrix B_{t-j} of the gene i . The parameter λ is multiplied by the penalty term to provide a trade-off between data-fit term and the penalty term. A solution to this equation can be found using Least Angle Regression (Lars) [29]. It was shown that Lars yields the entire lasso solution path with the computational cost of a single OLS.

One of the advantages of lasso is that it can be applied even if $p > N$, however, the maximum number of estimated non-zero coefficients will be N. In general, the maximum number of estimated non-zero coefficients for lasso is the minimum of (N, p). Lasso has been thoroughly studied in literature, and one of the most commonly used modifications of lasso is that due to Zou [30], who proposed an adaptive lasso penalty term that is weighted according to the initial estimates, and showed that if suitable weights are used, the adaptive lasso can achieve variable selection consistency. In [61] the properties of both lasso and adaptive lasso methods for multivariate time series models have been studied, and the necessary condition for consistent variable selection has been established. A variable selection procedure is said to be consistent if the probability of the procedure correctly identifying the set of non-zero covariates approaches unity when the sample size becomes very large.

a) *The Choice of Lasso Regularization Parameter λ*

The optimal choice of the regularization parameter λ is a crucial step in the lasso methods, since it controls the degree of penalization and hence, the degree of sparsity of the resulting model. The parameter λ is either chosen based on cross-validation or based on minimizing certain criterion such as Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC). AIC and BIC are based on the maximum likelihood estimates of the model parameters. In maximum likelihood, the idea is to estimate parameters so that, under the model, the probability of the observed data would be as large as possible. It is common to consider likelihoods on a log scale, and the logarithms of numbers between 0 and 1 are negative, so log-likelihoods are negative numbers. BIC tends to select more parsimonious models than AIC. It was previously shown that the choice of λ based on information criteria yields more accurate result than cross-validation [62] . Moreover, in [61] they

have observed that cross-validation results in overfitted models both for the lasso and adaptive lasso while BIC is providing the tuning parameter for which they are getting the true model size. In addition, BIC is consistent, that is, if there is a true underlying model, then with enough data the BIC will select that model. For a regression model, BIC is defined by finding the parameter θ_i that maximizes the following

$$\text{BIC} = \log L\left(Y/\widehat{\theta_i}\right) - \ df \times \log(n), \tag{5-19}$$

where $L\left(Y/\widehat{\theta_i}\right)$ is the likelihood function of the response Y. In the context of linear regression, maximizing (5-19) is equivalent to minimizing the following

$$\frac{RSS}{\sigma^2} + df \times log(N), \tag{5-20}$$

where RSS is the residual sum of squares after fitting, *N* is the number of samples , *df* is the degree of freedom. The BIC criterion for lasso [62] is easily calculated by minimizing (5-20). When the number of samples is sufficiently large, BIC selects the right model with a high probability; however, it will overestimate the number of covariates for sparse models [63]. Since BIC assumes that the prior probability is uniform, very small or very large model sizes have a smaller probability of occurrence compared to that of medium-sized models. Since GRNs are sparse, modified versions of BIC that take into account the sparseness have been applied. Frommlet *et al.* [64] have compared the modified versions of BIC for sparse models and have shown that mBIC2 outperforms other versions, where mBIC2 criterion is obtained by maximizing

$$S_k^j = \log L\left(Y/\widehat{\theta_J}\right) - df \times \log(N) - df\ (2\log(p) + c) + 2\log(df!) \tag{5-21}$$

and equivalently for linear regression, it is obtained by minimizing

$$\frac{RSS}{\sigma^2} + df \times log(N) + df \times (2\ log\ p + c) - 2\ log(df!). \tag{5-22}$$

In the above, j=1,2,.... corresponds to different *k*-dimensional models, and *c* a specified constant based on 200 samples. However, the number of samples *N* and the number of covariates can vary from one microarray setting to another. Hence, we now derive a unified scheme for the estimation of a valid value of *c* and the derivation is similar to that in [63].

Let $\widetilde{S_1}$denote the maximum of (5-21) over all the one-dimensional models (*df=1*) and let S_0 =log$L_o(Y/\hat{\mu},\hat{\sigma})$ be the value of the criterion for the null model involving no edges. Let *D* be the number of non-zero coefficients in the model chosen by our procedure. Then, the following holds:

$$P(D > 0) = P(\widetilde{S_1} > S_0) + P(D > 1, \widetilde{S_1} \leq S_0). \tag{5-23}$$

Consider one of the one-dimensional models and the corresponding value of S_1^j

$$S_1^j = \log L\left(Y/\widehat{\theta_J}\right) - \log(N) - (2\log(p) + c) \tag{5-24}$$

The corresponding model will be preferred over the model with zero covariates if $S_1^j > S_0$, or equivalently

$$\log \frac{L\left(Y/\widehat{\theta_J}\right)}{L_o(Y/\hat{\mu},\hat{\sigma})} > \log(N) + (2\log(p) + c) \tag{5-25}$$

Since, for any Gaussian random variable X of zero mean and variance σ^2, $\frac{X^2}{\sigma^2}$ follows a χ^2 distribution of *df* = 1. Then, under the null hypothesis of zero covariates in the model, $\log \frac{L\left(Y/\widehat{\theta_J}\right)}{L_o(Y/\hat{\mu},\hat{\sigma})}$ asymptotically has a χ^2-distribution with *df*=1. Thus, $P(S_1^j > S_0)$ is asymptotically equivalent to $P(Z > \sqrt{\log(N) + 2\log(p) + c})$, where *Z* is a Gaussian random variable with zero mean and unit variance.

For every $x > 0$,

$$P(Z > x) \leq \frac{1}{x\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{5-26}$$

Hence,

$$P(S_1^j > S_0) \leq \frac{e^{-\left(\frac{c}{2}\right)}}{p\sqrt{2\pi N(log(N)+2log(p)+c)}}, \tag{5-27}$$

$P(\widetilde{S_1} > S_0)$ is satisfied if at least one of the one-dimensional models S_1^j satisfies (5-27). Since Bonferroni inequality [65] states that

$$P(\bigcup_i A_i) \leq \sum_i P(A_i) \tag{5-28}$$

Hence, after applying Bonferroni inequality, $P(\widetilde{S_1} > S_0)$ is given by

$$P(\widetilde{S_1} > S_0) \leq p \frac{e^{-(\frac{c}{2})}}{p\sqrt{2\pi N(log(N)+2log(p)+c)}} \tag{5-29}$$

Thus, in order to control the degree of sparsity by α, where a smaller α yields a sparser network, we set α to be the right side expression of (5-29), that is,

$$\alpha = \frac{e^{-(\frac{c}{2})}}{\sqrt{2\pi N(log(N)+2log(p)+c)}} \tag{5-30}$$

$$\alpha \sqrt{2\pi N(log(N) + 2log(p))} = e^{-\left(\frac{c}{2}\right)} - \alpha \sqrt{c} \tag{5-31}$$

$$c + \log(c) = -2log(\alpha^2 \sqrt{2\pi n(log(n) + 2log(p))}) \tag{5-32}$$

Equation (5-32) is solved numerically to get *c*, and then, the mBIC2 criterion obtained by minimizing (5-22) is applied to estimate the optimal value of λ. However, BIC and its modifications do not always yield the optimum value for the regularization parameter λ. Our simulations have shown that, when the number of samples *N* is less than the covariates *p*, the BIC criterion will always choose the full network. This can be attributed to the fact that the residual sum of squares equals zero for the full network. These findings are consistent with that reported in [64]. Hence, when *N* is less than *p*, the cross-validation is employed rather than the mBIC2 criterion. Therefore, in this paper, depending on the number of samples compared to the number of covariates for each dataset, either cross-validation or mBIC2 criterion is applied to find the optimal value of λ.

b) *Backward Elimination*

After applying the lasso procedure, fine tuning of the resulting model is achieved by applying backward elimination. Backward elimination is a stepwise approach to variable selection that eliminates edges from the model, one by one, and whose removal leads to an improvement of the criterion. The elimination criterion is based on the BIC criterion. Backward elimination starts with the output model from lasso and sequentially removes edges that contribute least to the fit. The process stops when no further improvement can be achieved by the removal of an additional edge. It is performed with the function step of the statistical computing environment R. Subsequently, the

resulting model is fitted using OLS to find optimal coefficients. It is to be noted that, since the number of nonzero coefficients resulting from lasso is limited, the computational time for backward elimination is not prohibitively large.

5.2.4 Adaptive Lasso

The adaptive lasso technique [30] extends lasso by allowing different penalization parameters for different regression coefficients. The Adaptive lasso estimate for the coefficients is given by

$$\widehat{\beta} = \arg_{\beta} min \left\| \mathrm{X}_t - \sum_{j=1}^{d} B_{t-\mathrm{j}}\, \mathrm{X}_{t-j} \right\|^2 + \lambda \sum_{j=1}^{d} \sum_{i=1}^{p} \sum_{k=1}^{p} \left| w_{ik}^{j} \beta_{ik}^{(t-j)} \right| \quad (5\text{-}33)$$

where $w_{ik}^{j} = \frac{1}{\left|\beta_{ik,int}^{(t-j)}\right|^{\gamma}}$ and $\beta_{ik,int}^{(t-j)}$ is an initial estimate for the coefficients that can be computed using lasso itself or marginal regression. In adaptive lasso, the covariates with nonzero coefficients will be selected with probability tend to unity and the estimates of nonzero coefficients have the same asymptotic distribution as the correct model, provided that the initial estimates of the regression coefficients are consistent. However, in microarray data, the limited number of replicated samples and time points causes the initial estimates from ordinary least squares to be either unavailable or none-informative. Hence, consistent initial estimates of the regression coefficients are generally not available. The γ value is usually set to 1 [66], and the initial estimate $\beta_{ik,int}^{(t-j)}$ are available from applying the lasso technique. The adaptive lasso can also be solved using lars, where first the input genes are scaled such that $x_{ik}^{*} = \frac{x_{ik}}{w_{ik}}$, then the lasso problem is solved. Afterwards, the coefficients are rescaled such that $\beta_{ik}^{(t-j)} = \frac{\beta_{ik}^{(t-j)*}}{w_{ik}}$. Before applying the coefficient scaling, normalization is applied, so that the data is centered around the mean. Hence, the mean of gene expressions of each gene is set to zero, by subtracting the mean $\overline{X_t}$, where $\overline{X_t} = \frac{1}{T}\sum_{j=1}^{T} \mathrm{X}_j$. The weighted ℓ_1 minimization puts larger weights on the coefficients that are more likely to be zero, and puts smaller weights on the coefficients that are more likely to be non-zero, thus, promoting sparsity at the right positions.

5.3 Summary of the Proposed Approach DD-lasso

In this chapter, we have proposed a method for the reconstruction of gene regulatory network, which takes into account the different delays for various genes. The proposed method is now summarized in the form of an algorithm. The gene expression values are represented by a $p \times n \times T$ array, where p is the number of genes, n is the number of samples and T is the number of time points.

Step 1: Average the replicated samples of each gene to form a matrix M, which is a $p \times T$ array. Then, apply the cross-correlation between genes to this matrix to form a three-dimensional correlation matrix R of dimensions $p \times p \times d$, where d is the maximum delay between the various genes.

Step 2: Set the time delay τ=1, for a given gene i, shift this gene by delay τ and compute the pair-wise correlation between this gene and all the other genes without any delay. Repeat the same procedure for all the genes.

Step 3: Repeat step 2 for delay τ=2, 3, ..., d. Set the maximum delay d arbitrarily, depending on the biological knowledge and according to the number of time points of the given dataset.

Step 4: Compute all the values for the matrix R, for every pair of genes i and j. Then, compare their pair-wise correlations for different delays. The maximum correlation at a certain delay will indicate that, most probably, if there is a relation between genes i and j, this relation exists at that delay δ_{ij}, where $\delta_{ij} \in \{1, 2, . .., d\}$.

Step 5: For a given gene i, shift the other genes according to the delays δ_{ij}. Apply the lasso technique, where the input genes are the genes delayed by δ_{ij}. The original microarray data with replicated samples n is used in lasso.

Step 6: In the lasso technique, the total number of samples input to lasso is $N = n \times (T - d)$. In order to estimate the penalty parameter λ, if $N < p$, apply cross-validation; otherwise, apply the mBIC2 criterion.

Step 7: Filter the resulting model using backward elimination.

Step 8: Repeat Steps 5-7 for all the genes, resulting in the final network.

Since, the p regression problems are solved independently the proposed method can be implemented using parallel programming. In the next chapter the proposed method is applied to both synthetic and real data sets to evaluate is performance and compare it to existing methods.

Chapter 6

Experimental Results on Network Reconstruction

6.1 Synthetic and Real Datasets Description

In order to compare the performance of the proposed technique with that of the previous ones, we carry out a simulation study by generating synthetic GRN to mimic the possible gene networks behavior. Various artificial networks can be generated such as random networks, scale free networks and networks composed of small regulatory network motifs. It is known that the GRN has mostly a scale free network topology, and hence, all the synthetic GRN are generated in the form of scale free network. First, the structure of the GRN is generated so that the network topology is a scale-free network. Then, the relationships between various nodes are represented by time delayed linear equations. Scale-free networks are characterized by the majority of vertices having only a few connections, while a small number of vertices have a very large number of connections. It has been shown that many real biological networks exhibit such structure [67-69]. Scale-free networks are generated by the algorithm described by Albert and Barabasi [70]. The relationships between the various nodes are represented by time-delayed linear equations. The observations are generated according to a Vector Auto-Regressive (VAR) model with order 3 and a zero mean Gaussian noise with standard deviation of σ is added to the observations. That is, the observations are generated as follows:

$$\mathbf{X}_t = A_{t-1} \cdot \mathbf{X}_{t-1} + \cdots + A_{t\text{-d}} \cdot \mathbf{X}_{t-\text{d}} + \varepsilon \tag{6-1}$$

where $\mathbf{X}_t$ is the microarray gene values at time *t, d* is the maximum delay, the *A* matrices are adjacency matrices at various time delays and ε is a zero-mean Gaussian noise. In order to generate the time-series data, the obtained model is applied on a random initial vector. This VAR model is similar to the one used in [13]. Let the non-zero elements of the adjacency matrices be set at +0.8 or -0.8, representing activations or inhibitions, respectively. The number of genes *p* is fixed at 50, the number of time points *T* is chosen as 10 and 20, the number of samples *n* to be 4, 10, and 50, and the standard deviation σ of the Gaussian noise to be 0.1, 0.5 and 1. For each of the combinations of the

parameters, 100 networks are generated. Thus, the total number of networks generated is 1800. The network structure is generated using the existing package "igraph", where network generation follows a discrete time step model and at each time step, a single vertex is added. During the network generation, two edges are added at each step; however, some of the edges may get repeated. Hence, the actual number of edges varies, ranging between 88 and 96 edges. The output networks are checked to ensure that there are no two genes having the same exact linear relationships. In order to evaluate the performance of the proposed technique and compare it with that of the previous ones, the following metrics are used.

1. Precision: $P = \frac{TP}{(TP+FP)}$ is the fraction of retrieved instances that are relevant.

2. Recall: $R = \frac{TP}{(TP+FN)}$ is the fraction of relevant instances that are retrieved.

3. F1-measure: $F1 = \frac{2PR}{(P+R)}$ is a trade-off between the Recall and Precision.

where

TP, the true positive, is the number of correctly identified edges,

FP, the false positive, is the number of false edges, and

FN, the false negative is the number of missed edges.

The reconstructed networks from various methods are compared to the generated adjacency matrices, based on which all the metrics have been calculated.

In addition, two real datasets are used, where the first dataset resulted from an experiment on human hela cells, while the second is from an experiment to study yeast cell cycle. The first real dataset is extracted from human uterine cervical carcinoma cells. The thorough understanding of the regulation of the cell cycle division is crucial for studying diseases such as cancer development. In [71] the cell cycle synchronized by a double thymidine block has been examined, where the microarray data from the experiment conducted by [71] is used and downloaded from (http://genome-www.stanford.edu/ Human -CellCycle/HeLa). The samples are measured at time 0 and every hour for 46 hours, hence, 47 time points are available. There are two replicates at time 0,

that are averaged to get the values at time 0. In the second real dataset, yeast cell cycle regulation is studied by Spellman *et al.*[72]. This microarray experiment was designed to create a list of yeast genes whose transcription levels were expressed periodically within the cell cycle. In [72] the yeast cell cycle synchronized by alpha factor has been examined, where the samples are measured at time 0 and every 10 minutes, for a total of 18 time points are available, which covers two complete cycles of cell division.

6.2 Partial Correlation Dependency Networks

First, the replicated samples are averaged; then, Pearson correlations are applied to form a correlation matrix. Afterwards, using the inverse of the correlation matrix and using (5-2) partial correlations are evaluated. In order to determine the significant partial correlation coefficients a hypothesis test is implemented where the null hypothesis states that the true correlation is zero, H_0: $\rho = 0$, versus the alternative hypothesis that the correlation is non-zero, H_a: $\rho \neq 0$. Under the null hypothesis, the distribution of the test statistic, *S*, is assumed to follow student-t distribution of degree of freedom *df*, and is defined by

$$S = r\sqrt{\frac{df}{1-r^2}}, \tag{6-2}$$

where *r* is the observed partial correlation. The results for the partial correlation dependency networks are obtained by considering only the partial correlations with p-values less than 10^{-6}. As mentioned earlier, 100 networks are generated for each noise level σ. Hence, the results for TP rate, FP rate, Precision, Recall and F1-measure are averaged over 300 networks for each *n* and the results are given in Table 6-1.

The output networks is an undirected graph, and hence when compared to the generated adjacency matrix, the absence or existence of an edge is tested regardless of the direction of relation between any two genes. As seen from Table 6-1, since the data is averaged over the replicated sample, the results are the same regardless of the number of samples. An important reason for the false positives is the dependence of two nodes on a third node, and thus, the indirect relation between two nodes may be falsely detected as a direct relation.

Table 6-1 Partial corrlation results

	n	TP rate	FP rate	P	R	F1
T=10	4	56.225%	26.6392%	7.62538%	56.225%	0.134238
	10	55.715%	26.6623%	7.5504%	55.715%	0.13292
	50	55.0044%	26.367%	7.541%	55.0044%	0.13258
T =20	4	61.9027%	29.262$	7.63655%	61.9027%	0.13590
	10	62.571%	29.2983%	7.7030%	62.571%	0.137120
	50	62.384%	29.2486%	7.6969%	62.384%	0.13697

There is a need to distinguish between the direct and indirect types of relation, and to understand the directionality of relation, that is, which gene is affecting the other. The application of only partial correlations to generate dependency networks yields networks with poor performance, hence more advanced GRN reconstruction techniques are applied.

6.3 Network Reconstruction Results Using Synthetic data

Microarray data is characterized by a limited number of samples and time points; hence, a successful gene reconstruction technique that is based on microarray data should take into account this limited number of data points. In the proposed method, the limited number of samples present in a microarray data is not a constraint. This is in view of the fact that lasso is applied to reconstruct the GRN, where the input data to the lasso is arranged in such a way that the final number of samples is $N= n\times(T- d)$, instead of n. It is noted that the larger the maximum delay d examined, the smaller the number of samples N used. This data arrangement exploits all possible data points to efficiently estimate the GRN and the model parameters, and is used throughout the experiments. As mentioned earlier, when $N<p$, the information criteria will select the full model; consequently, only cross-validation can be applied. On the other hand, when $N \geq p,$ the performance of lasso based on the choice of λ through cross-validation, BIC, or mBIC2 needs to be studied. The cross-validation technique, BIC, or mBIC2 criterion that provides the best performance for lasso is then applied to the proposed DD-lasso, and its performance compared with that of the former. Next, the performance of DD-lasso is compared to that of existing methods. In the generated synthetic data, when n=4 and

T=10 $N < p$, while for the remaining datasets $N > p$.

6.3.1 The Performance of the Delay Detection

The ability of correlations to detect the delay between two genes correctly is examined using several techniques. In Pearson correlation method, the data is averaged over the replicated samples and then the Pearson correlations are applied to the averaged data. Using standard deviation (SD)-weighted correlation coefficient [58] and CORREP [59], the replicated samples are used directly. The results of the true delays detected for the three methods are averaged over 300 networks, for each sample size *n* and are shown in Table 6-2.

Table 6-2 TP rate of delays for the three correlation methods

	n	Pearson correlation TP rate	SD-weighted correlation coefficient TP rate	CORREP TP rate
T=10	4	73.7228%	70.9363%	72.84145%
	10	73.5212%	73.099%	65.7057%
	50	73.5174%	74.2695%	72.85811%
T =20	4	85.18696%	84.63426%	84.27514%
	10	85.07625%	86.64285%	83.59233%
	50	84.8201%	87.4956%	83.92043%

As seen from Table 6-2, the method using Pearson correlations has an average performance that is slightly affected by the sample size, since the microarray expression data of each gene is averaged over the samples prior applying the correlations. Using the SD-weighted correlations, the TP rate improves with the larger number of samples, however for small number of samples, it will have lower TP rate than that of Pearson correlation. The CORREP has the lowest TP rate. Hence, averaging the data, then applying Pearson correlations is the most suitable technique for microarray data that are characterized by small number of samples, and is applied in the proposed method. The

accuracy of the estimate of the delay based on cross-correlation improves with longer time series and higher Signal-to-Noise ratio (SNR).

6.3.2 The Performance of the Lasso Regularization Parameter Selection

a) Using Cross-validation technique

The results for lasso are obtained at a fixed delay of unity, and the penalty parameter λ is chosen separately for each gene based on a 10-fold cross-validation. For each noise level σ, as mentioned earlier, 100 networks are generated. Hence, the results for TP rate, FP rate, Precision, Recall and F1-measure are averaged over 300 networks for each *n* and the results are given in Table 6-3. The average TP rate, FP rate and F1-measure are shown in Figure 6.1.

Table 6-3 Results for 10-fold cross validation

	n	*N*	TP rate	FP rate	P	R	F1
T =10	4	36	63.9328%	28.606%	8.119%	63.9328%	0.144(0.013)
	10	90	62.979%	22.371%	10.0465%	62.979%	0.173(0.019)
	50	450	74.537%	38.813%	7.054%	74.537%	0.129(0.012)
T =20	4	76	63.674%	24.3229%	9.4295%	63.674%	0.164(0.019)
	10	190	70.925%	35.2715%	7.3590%	70.925%	0.133(0.013)
	50	950	83.136%	52.779%	5.8488%	83.136%	0.109(0.009)

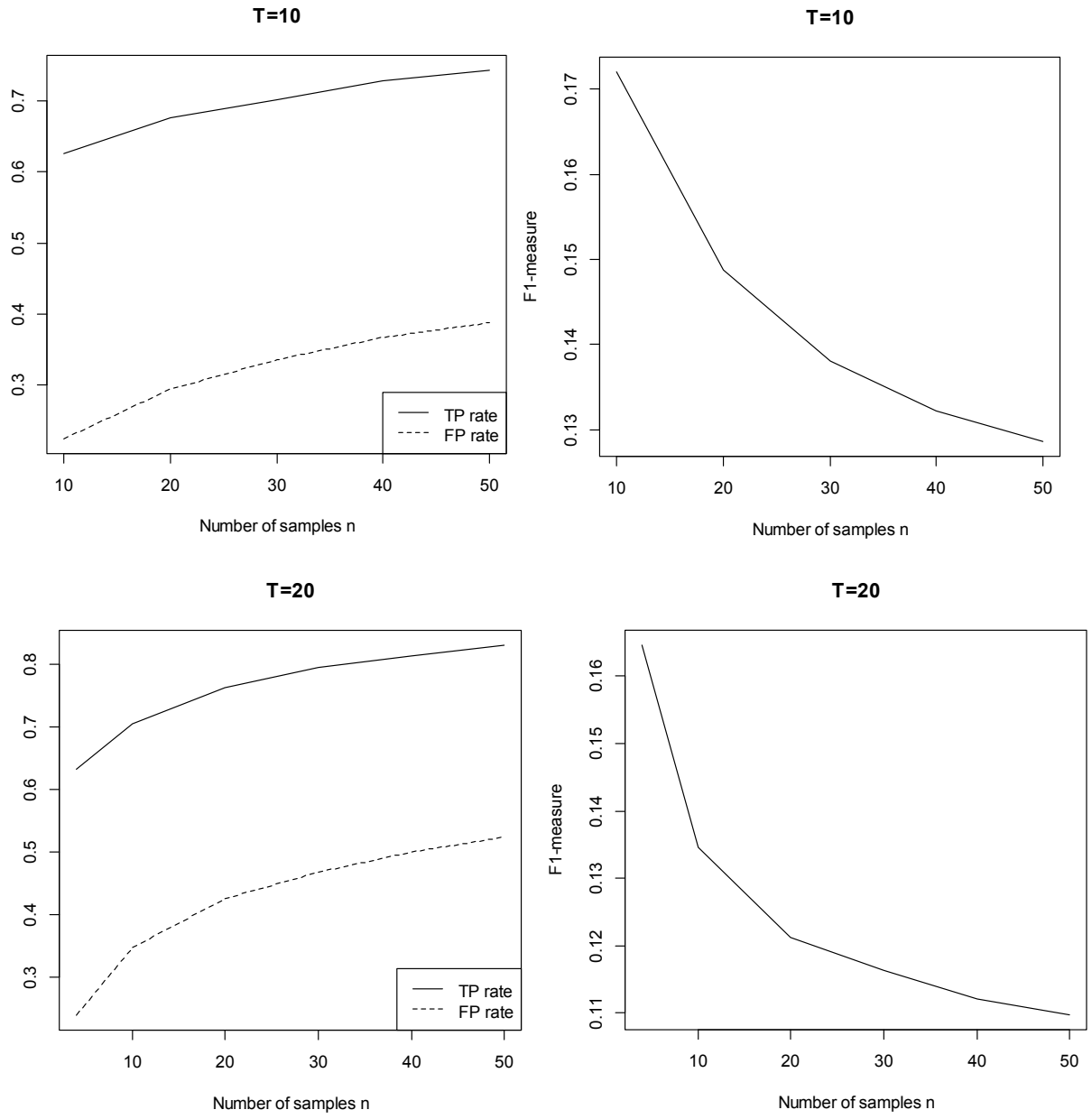

Figure 6.1 TP rate, FP rate and F1-measure at T=10 and T=20 for cross-validation

It is seen from Table 6-3 and Figure 6.1 that when n (or N) is large, so is the FP rate. This is due to the fact that as n increases, cross-validation selects a large number of edges resulting in a deteriorated performance. Hence, for large n (or N) a better criterion should be applied. This is true except for the case n=4 and T=10, because in this case $N<p$, and hence, the maximum number of

non-zero coefficients that can be inferred using lasso is N and not p for each gene.

b) Using BIC criterion

The penalty parameter λ is chosen separately for each gene based on the BIC criterion. The results for TP rate, FP rate, Precision, Recall and F1-measure are averaged over 300 networks for each n and the results are given in Table 6-4. The average TP rate, FP rate and F1-measure are shown in Figure 6.2. It is seen from this table and figure, the selection of λ based on the BIC criterion yields better results than that based on cross-validation. However, when n or N is very large, the FP rate is still large. As mentioned earlier, when $N<p$, the information criteria selects the full model; resulting in a deteriorated performance as shown for the case n=4 and T=10.

Table 6-4 Results for BIC criteria

	n	*N*	TP rate	FP rate	P	R	F1
T =10	4	36	84.330%	69.409%	4.5582%	84.330%	0.087(0.004)
	10	90	56.990%	11.767%	16.237%	56.990%	0.252(0.031)
	50	450	61.158%	18.233%	11.788%	61.158%	0.197(0.024)
T =20	4	76	58.524%	16.054%	12.691%	58.524%	0.208(0.027)
	10	190	61.208%	19.639%	11.038%	61.208%	0.187(0.024)
	50	950	68.759%	30.805%	8.1381%	68.759%	0.145(0.018)

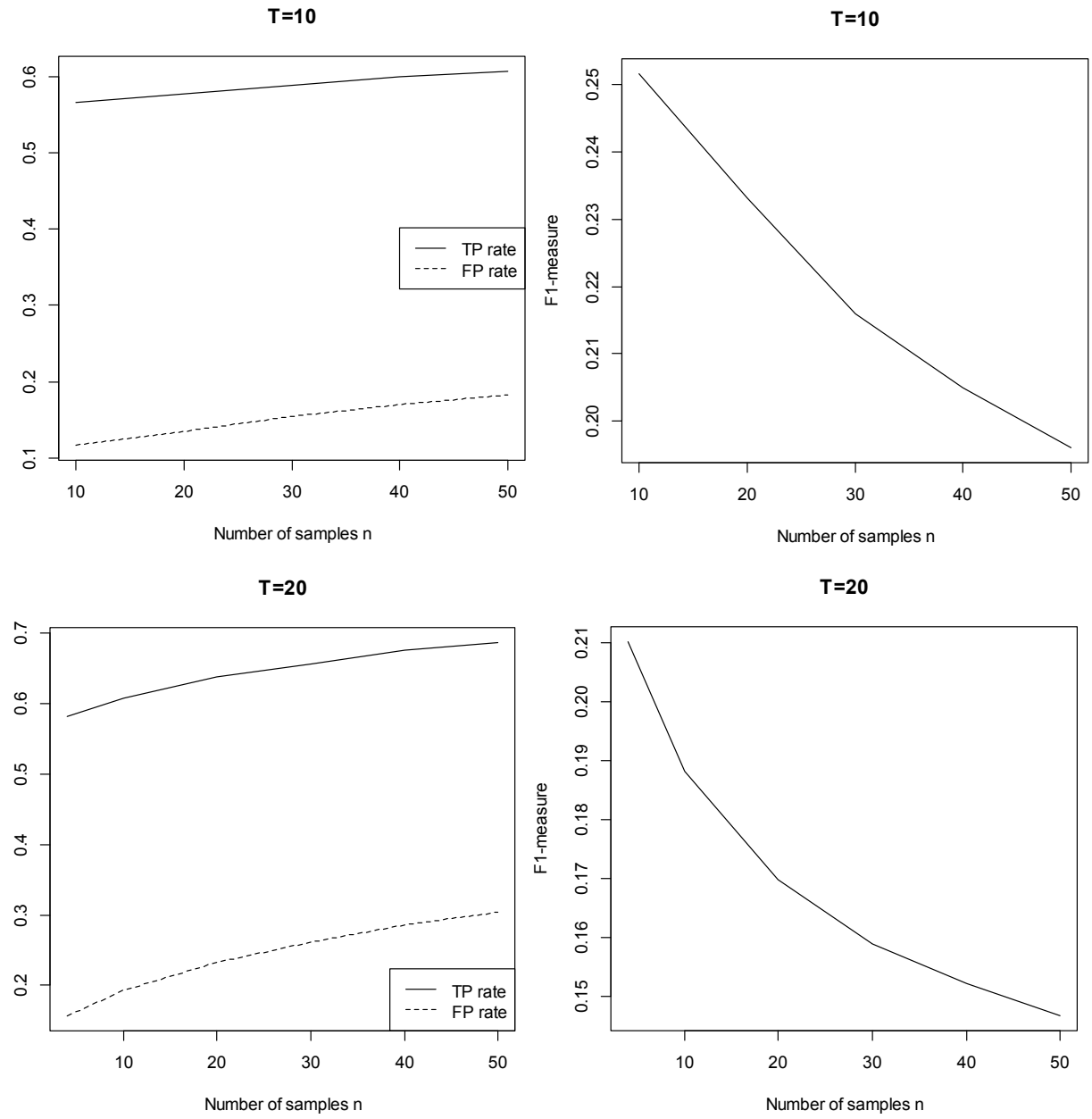

Figure 6.2 TP rate, FP rate and F1-measure at T=10 and T=20 for BIC criterion

c) Using mBIC2 criterion

The penalty parameter λ is chosen separately for each gene based on mBIC2 criterion. For the dataset of n = 10 and T = 20, Figure 6.3 shows the average TP and FP rates and Figure 6.4 shows the average Precision and Recall for α ranging from 10^{-7} to 0.1. Figure 6.5 shows the average F1-

measure as a function of α for the same dataset.

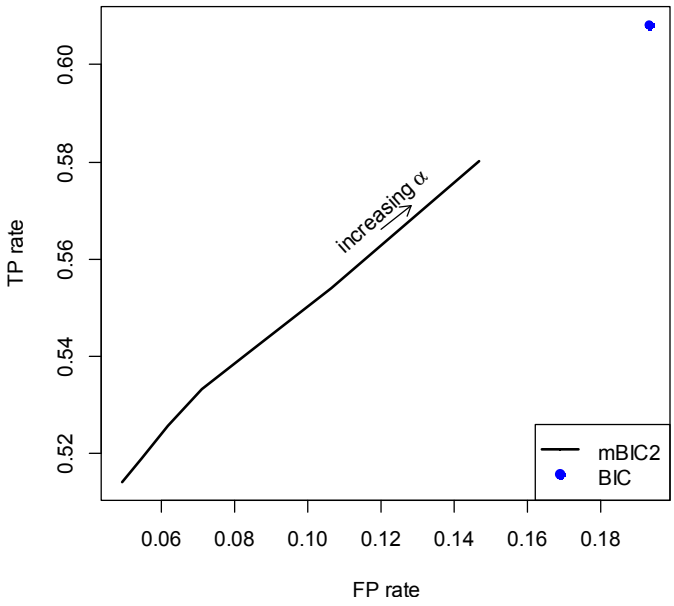

Figure 6.3 TP rate and FP rate for various α

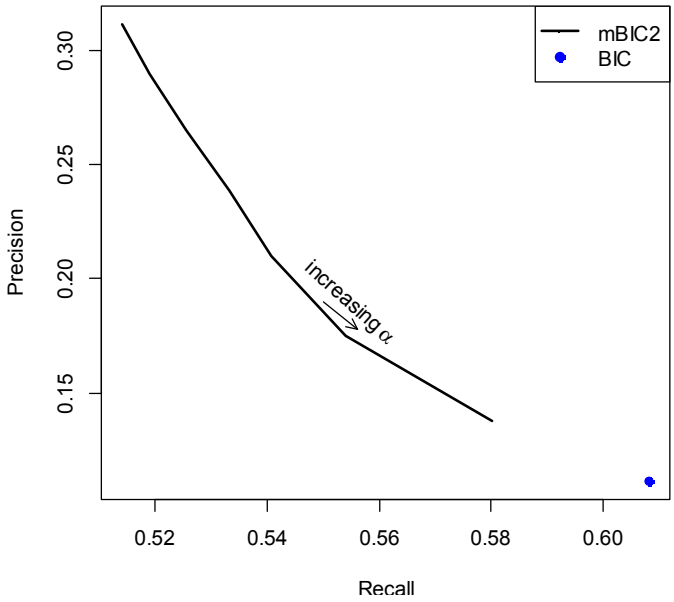

Figure 6.4 Precision, P, and Recall. R, for various α

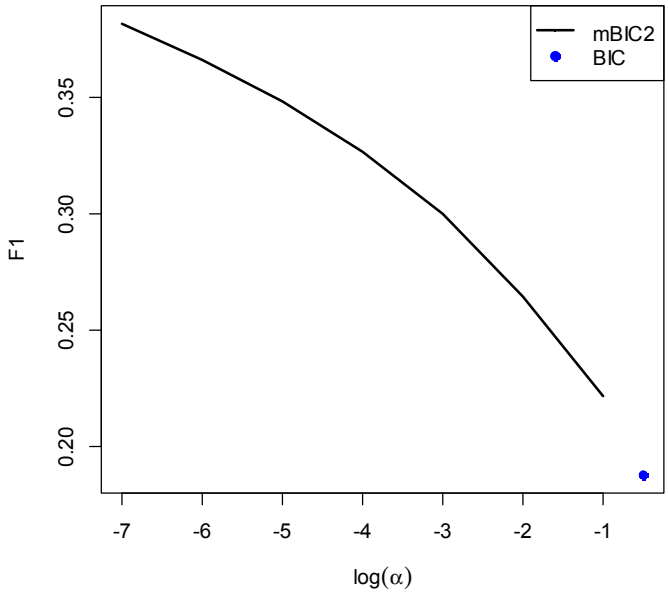

Figure 6.5 F1-measure for various **α**

For the purpose of comparison, the corresponding values of TP rate, FP rate, Precision and Recall using the BIC criterion are shown in Figures 6.3 and 6.4, while the F1-value for BIC criterion is shown in Figure 6.5; however, they are not dependent on α. It is seen from Figure 6.3 that, for any value of α the TP rate is higher than the FP rate. In addition, for smaller values of α both the TP and FP rates are smaller, and hence, the total number of edges is smaller, leading to a sparser network. These results are consistent with the findings in Section 5.4.1, namely, that α controls the degree of sparsity, and a smaller α yields a sparser network. In addition, irrespective of the value of α sparser networks are generated relative to that generated using the BIC criterion. Since the FP rate increases with N for a fixed p irrespective of the criterion used, the larger the sample size N compared to p, the smaller the α should be. A simple guide for the selection of α is to let $\alpha=10^{-\gamma}$, where $\gamma = (N/p)$. Furthermore, α can be set arbitrarily according to the degree of sparseness that is determined based on previous biological knowledge. In order to illustrate the performance of mBIC2, we choose α to be the geometric mean of 10^{-7} and 0.1, that is, α = 0.0001. The resulting models are neither very sparse nor are they very close to that of BIC criterion. Thus this value of α can serve as a guideline, when we have no previous knowledge of the degree of sparseness of the resulting network. For this

value of α, the results for TP rate, FP rate, Precision, Recall and F1-measure are averaged over 300 networks for each n and are given in Table 6-5. In addition for the same value of α, the average F1-measures are given in Table 6-6, where the standard deviation is shown in brackets. It is seen from Figures 6.3 to 6.5 and Tables 6-5 and 6-6 that the mBIC2 criterion provides better results compared to that using the BIC criterion or the cross-validation technique. Moreover, cross-validation takes more time to find the optimal λ than the criterion-based selection does. Hence, mBIC2 is recommended for sparse solutions, such as GRN reconstruction, and is used for DD-lasso. When $N<p$, the mBIC2 criterion selects the full model; resulting in a deteriorated performance as shown for the case $n = 4$ and $T = 10$. Thus, the cross-validation technique is applied when $N < p$, while mBIC2 criterion is used when $N \geq p$.

Table 6-5 Results for mBIC2 criteria

	n	TP rate	FP rate	P	R	F1
T=10	4	84.330%	69.4097%	4.5582%	84.330%	0.0865(0.004)
	10	50.235%	1.791%	54.154%	50.235%	0.516(0.052)
	50	53.729%	6.316%	25.594%	53.729%	0.345(0.043)
T=20	4	50.075%	4.247%	34.542%	50.075%	0.3993(0.072)
	10	53.373%	7.157%	23.698%	53.373%	0.3246(0.054)
	50	58.467%	15.497%	13.157%	58.467 %	0.2141(0.032)

Table 6-6 Results for the F1-measure of CV, BIC and mBIC2 criterion

	n	CV	BIC	mBIC2
T=10	4	**0.1439**(0.013)	0.08647(0.004)	0.0865(0.004)
	10	0.173(0.019)	0.252(0.031)	**0.516**(0.052)
	50	0.129(0.012)	0.197(0.024)	**0.345**(0.043)
T=20	4	0.164(0.019)	0.208(0.027)	**0.3993**(0.072)
	10	0.133(0.013)	0.187(0.024)	**0.3246**(0.054)
	50	0.109(0.009)	0.145(0.018)	**0.2141**(0.032)

6.3.3 The Performance of the Proposed Delay Detection-lasso (DD-lasso)

The maximum delay for the DD-lasso is chosen as 3 and the delay for lasso as unity. The

maximum delay for lasso is not chosen as 3, since the number of covariates to be determined will be *3p*, instead of *p*. This large number of unknowns will need larger number of samples, and for the current limited number of samples, lasso will yield a deteriorated performance compared to that of lasso when the delay is chosen as unity. The results for Precision, Recall, F1-measure, TP rate and FP rate are averaged over 300 networks for each *n* and the results are given in Tables 6-6 and 6-7. The average Precision and Recall for each *n* are shown in Figure 6.6.

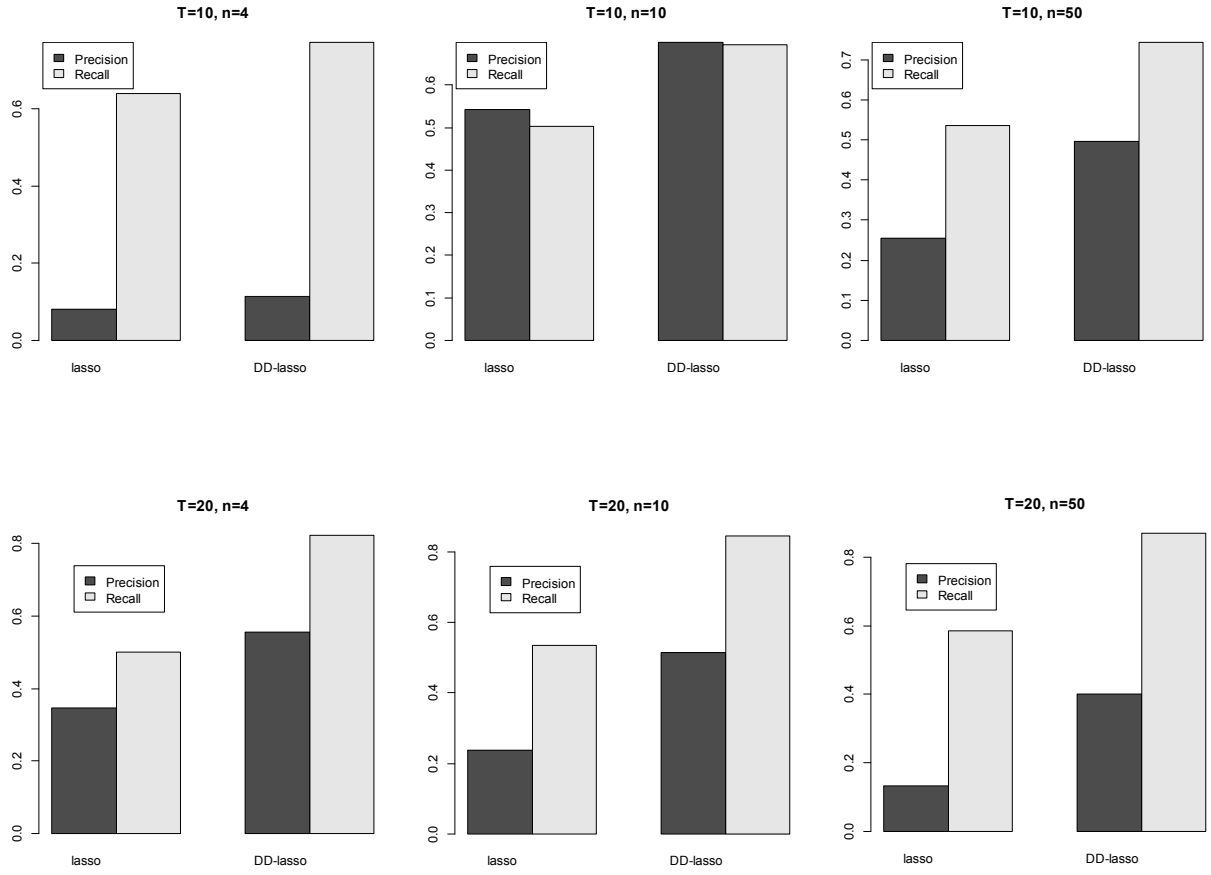

Figure 6.6 Bar plot of the average Precision and Recall of 300 networks for each *n*

It is seen from these tables and figures that our proposed DD-lasso significantly outperforms lasso in terms of all the parameters. It has better Precision, Recall, higher TP rate, and a lower FP rate at the same time. Finally, a back-ward elimination technique is applied to fine tune the results.

Table 6-7 P, R and F1 for DD-lasso

		Proposed DD- lasso			lasso		
	n	P	R	F1	P	R	F1
T=10	4	11.378%	77.1764%	**0.198**(0.016)	8.119%	63.9328%	0.1439(0.013)
	10	69.965%	69.413%	**0.694**(0.062)	54.154%	50.235%	0.516(0.052)
	50	49.590%	74.344%	**0.592**(0.065)	25.594%	53.729%	0.345(0.043)
T=20	4	55.510%	82.241 %	**0.656**(0.119)	34.542%	50.075%	0.3993(0.072)
	10	51.454%	84.442%	**0.632**(0.122)	23.698%	53.373%	0.3246(0.054)
	50	40.182%	86.899%	**0.541**(0.128)	13.157%	58.467 %	0.2141(0.032)

Table 6-8 TP rate and FP rate for DD-lasso

		Proposed DD- lasso		lasso	
	n	TP rate	FP rate	TP rate	FP rate
T=10	4	77.1764%	23.706%	63.9328%	28.606%
	10	69.413%	1.226%	50.235%	1.791%
	50	74.344%	3.090%	53.729%	6.316%
T=20	4	82.241%	2.919%	50.075%	4.247%
	10	84.442%	3.562%	53.373%	7.157%
	50	86.899%	5.906%	58.467%	15.497%

6.3.4 The Effect of Backward-Elimination

The performance of the proposed DD-lasso with backward elimination is compared with that of DD-lasso. The maximum delay for both is chosen as 3 and cross-validation technique applied when $n = 4$ and $T = 10$, while mBIC2 criterion is used for the remaining datasets. The results for Precision, Recall, TP rate and FP rate are averaged over 300 networks for each n and are given in Table 6-10. The average Precision and Recall for each n are shown in Figure 6.7, and the F1-measure is given in Table 6-9. It is seen from these figures and tables that the backward elimination improves the overall results, mainly by controlling the false positives. Backward Elimination controls the false positives and thus improves the precision while the recall is slightly affected and

hence improving the overall F1-measure. As mentioned earlier, the computational cost of backward elimination is minimal.

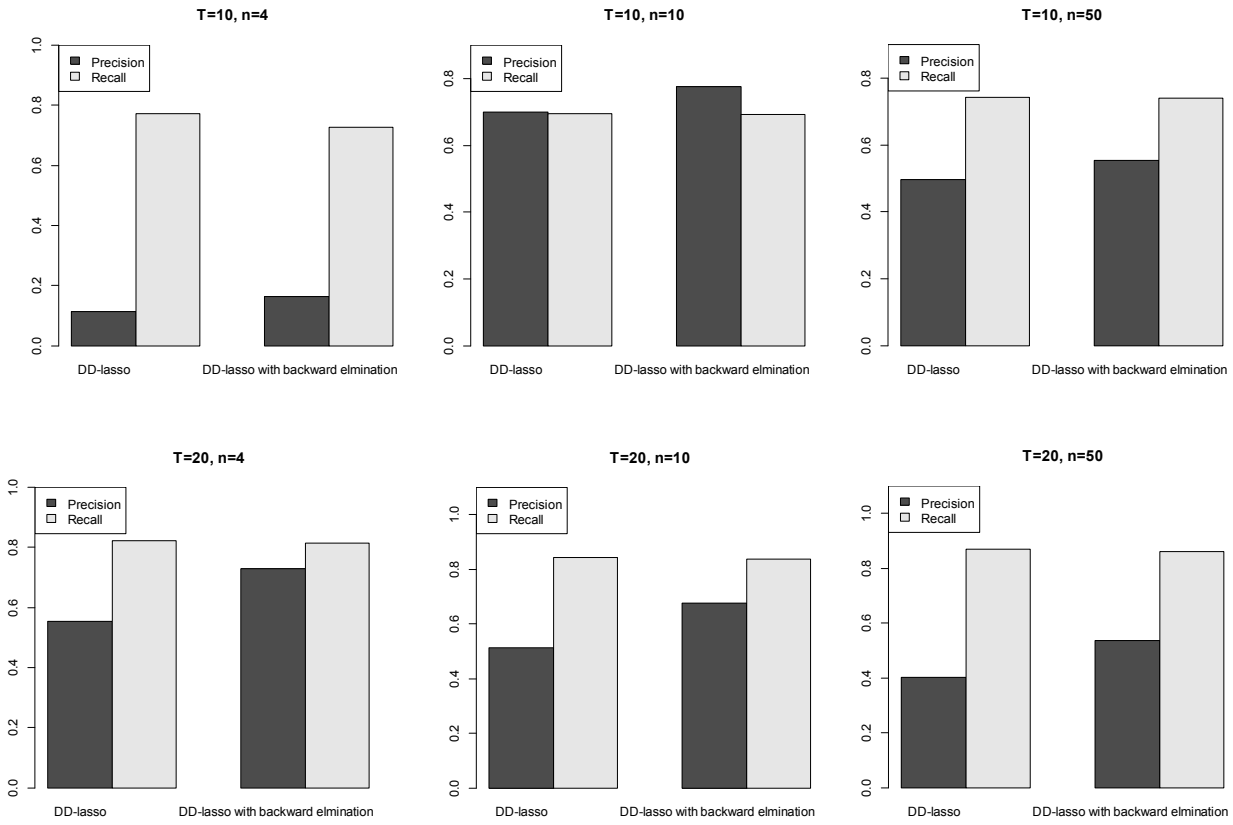

Figure 6.7 Bar plot of the average Precision and Recall of 300 networks for each *n*

Table 6-9 Results for the F1-measure of DD-lasso with and without backward elimination

	n	DD-lasso with backward elimination	DD- lasso without backward elimination
T=10	4	**0.267**(0.028)	0.198(0.016)
	10	**0.731**(0.059)	0.694(0.062)
	50	**0.633**(0.062)	0.592(0.065)
	60	**0.621**(0.06)	0.579(0.062)
T=20	4	**0.768**(0.102)	0.656(0.119)
	10	**0.745**(0.103)	0.632(0.122)
	50	**0.655**(0.126)	0.541(0.128)
	60	**0.637**(0.13)	0.524(0.131)

Table 6-10 Results of P, R, TP rate and FP rate for DD-lasso with and without backward elimination

		DD-lasso with backward elimination				DD- lasso without backward elimination			
	n	P	R	TP rate	FP rate	P	R	TP rate	FP rate
T=10	4	16.399%	72.614%	72.614%	14.712%	11.378%	77.1764%	77.1764%	23.706%
	10	77.717%	69.254%	69.254%	0.8005%	69.965%	69.413%	69.413%	1.226%
	50	55.497%	74.066 %	74.066 %	2.402%	49.590%	74.344%	74.344%	3.090%
	60	53.431%	74.533%	74.533%	2.610%	47.653%	74.839%	74.839%	3.340%
T=20	4	73.025%	81.530%	81.530%	1.258%	55.510%	82.241 %	82.241%	2.919%
	10	67.605%	83.657%	83.657%	1.693%	51.454%	84.442%	84.442%	3.562%
	50	53.717%	86.070%	86.070%	3.306%	40.182%	86.899%	86.899%	5.906%
	60	51.460%	85.787%	85.787%%	3.613%	38.3013%	86.9649%	86.9649%	6.4176%

6.3.5 The Robustness of DD-lasso for various values of d

Simulation studies of DD-lasso at other maximum delays, such as at d=4, and d=5 are carried out. The results for Precision, Recall, F1-measure, TP rate and FP rate are averaged over 300 networks for each *n*. The performance of DD-lasso at maximum delay of 3(the true delay) is compared with that of DD-lasso at d equals 4 and 5, and the results are shown in Figure 6.8, and Tables 6-11 and 6-12.

Table 6-11 Results of P, R and F1 for DD-lasso other delays

		DD-lasso with backward Elimination, d=4			DD- lasso with backward elimination, d=5		
	n	P	R	F1	P	R	F1
T=10	4	14.386%	62.248%	**0.233**(0.026)	13.026%	52.147%	0.2081(0.027)
	10	69.902%	58.192%	**0.6325**(0.062)	56.956%	47.792%	0.5149(0.059)
	50	48.292%	65.071%	**0.5526**(0.058)	42.983%	55.259%	0.4813(0.058)
T=20	4	68.882%	77.0472%	**0.7251**(0.105)	65.428%	73.359%	0.6889(0.107)
	10	63.58%	80.08%	**0.7060**(0.106)	60.762%	77.04%	0.6765(0.106)
	50	48.684%	82.501%	**0.6067**(0.119)	45.414%	79.588%	0.5728(0.117)

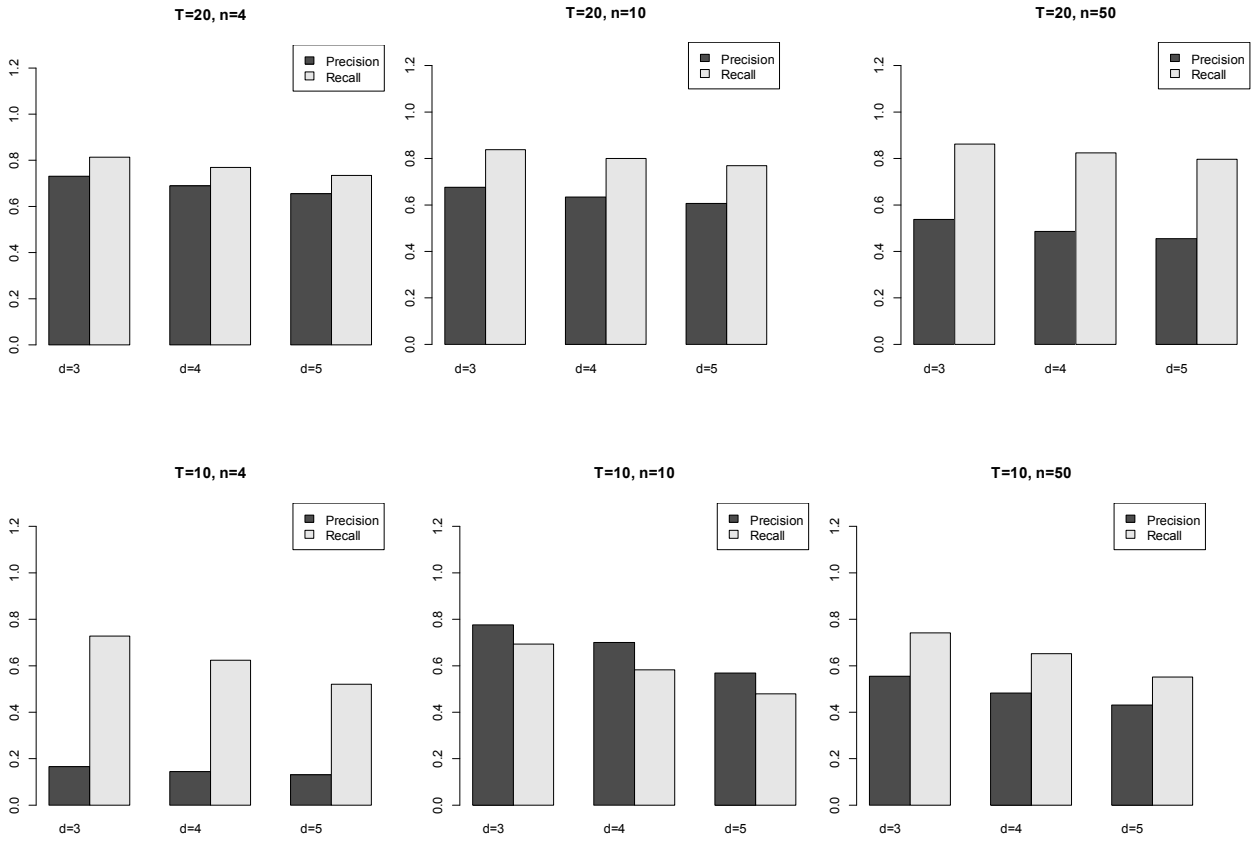

Figure 6.8 Precision and Recall for DD-lasso with backward elimination at different delays.

Table 6-12 TP rate and FP rate for DD-lasso with and without backward elimination

		DD-lasso with backward elimination, d=4		DD- lasso with backward elimination, d=5	
	n	TP rate	FP rate	TP rate	FP rate
T=10	4	62.248%	14.69%	52.147%	13.792%
	10	58.192%	1.014%	47.792%	1.509%
	50	65.071%	2.809%	55.259%	2.97%
T=20	4	77.0472%	1.449%	73.359%	1.624%
	10	80.08%	1.936%	77.04%	2.099%
	50	82.501%	3.814%	79.588%	4.179%

As seen from these figures and tables, the performance of DD-lasso is slightly affected by

estimating a wrong maximum delay, due to the introduction of more possible errors. However, the overall performance is still better than that of lasso without the delay detection and better than the previous methods. Hence, we can say that DD-lasso will yield a good performance, even if the maximum delay is not set to the correct value. At T=10, the time series is short (only 10 points) and when d is set to 5, only one half of the samples is used in lasso for network reconstruction. Hence, it is to be expected that the performance at d=5, would worsen due to the lower number of data points used, and not just because of the overestimation of d.

The proposed delay-detection framework can be similarly combined with adaptive lasso, instead of lasso, which will be shown in next section.

6.3.6 The Performance of the Proposed Adaptive DD-lasso and Adaptive Lasso

The maximum delay for the adaptive DD-lasso is chosen as 3 and the delay for adaptive lasso as unity. Lasso is used to estimate the initial values of the coefficients and γ is chosen as 1. The results for F1-measure, Precision, Recall, TP rate and FP rate are averaged over 300 networks for each *n* and are given in Tables 6-13 and 6-14.

Table 6-13 F1 for Adaptive DD-lasso

	n	Adaptive DD-lasso	Adaptive lasso
T=10	4	**0.20996**(0.006)	0.13938(0.017)
	10	**0.69101**(0.0627)	0.52107(0.051)
	50	**0.59244**(0.065)	0.34465(0.043)
T=20	4	**0.6622**(0.115)	0.41160(0.069)
	10	**0.6425**(0.118)	0.33332(0.052)
	50	**0.5499**(0.127)	0.2168(0.032)

Table 6-14 P, R, TP rate and FP rate for Adaptive DD-lasso

		Proposed Adaptive DD- lasso				Adaptive lasso			
	n	P	R	TP rate	FP rate	P	R	TP rate	FP rate
T=10	4	12.1865%	76.1500%	76.1500%	21.662%	7.8623%	61.954%	61.954%	28.838%
	10	71.3818%	67.4542%	67.4542%	1.10045%	55.798%	49.7493%	49.7493%	1.6468%
	50	49.639%	74.2327%	74.2327%	3.0749%	25.7353%	53.6782%	53.6782%	6.26039%
T=20	4	57.8224%	78.6017%	78.6017%	2.44209%	37.307%	47.8938%	47.8938%	3.5232%
	10	53.0715%	83.3513%	83.3513%	3.2335%	24.672%	52.9078%	52.9078%	6.67347%
	50	41.0985%	86.7509%	86.7509%	5.64706%	13.3660%	58.2851%	58.2851%	15.1575%

By comparing Tables 6-6 and 6-7 to Tables 6-13 and 6-14, it is seen that the adaptive lasso technique does not significantly improve the lasso results, while it has a higher computational cost. In addition, the proposed adaptive DD-lasso significantly outperforms the adaptive lasso technique in terms of all the parameters. It has better Precision, Recall, higher TP rate, and a lower FP rate at the same time. A back-ward elimination technique is applied to fine tune the results, where the performance of the proposed adaptive DD-lasso with backward elimination is given in Table 6-15, where the results for Precision, Recall, F1-measure, TP rate and FP rate are averaged over 300 networks for each *n*.

Table 6-15 P, R and F1 for Adaptive DD-lasso with backward elimination

	n	TP rate	FP rate	P	R	F1
T=10	4	72.010%	14.3657%	16.609%	72.010%	0.2695(0.028)
	10	67.4117%	0.76578%	77.940%	67.4117%	0.72120(0.062)
	50	73.993%	2.4205%	55.287%	73.993%	0.6312(0.0628)
T=20	4	78.4018%	1.2104%	72.6217%	78.4018%	0.7526(0.106)
	10	82.9145%	1.6607%	67.7410%	82.9145%	0.7432(0.104)
	50	86.0421%	3.3035%	53.7354%	86.0421%	0.6553(0.126)

As seen from Table 6-15, similar to DD-lasso with backward elimination, adaptive DD-lasso with backward elimination improves the overall results, mainly by controlling the false positives, while

the computational cost of backward elimination is minimal. Next, we will compare our approach with previous work that is concerned with the same problem.

6.3.7 Comparison of the Proposed Approach with Existing GRN Reconstruction Methods

The most two recent research works that consider time delays and VAR models, for microarray data, are that of Lozano *et al.* [13] and Shojaie *et al.* [14]. In order to compare the performance of our proposed method with these two methods, we repeat their procedures on the same synthetic datasets that we have used.

Lozano *et al.* [13] have used a group lasso penalty term in order to obtain a Granger graphical model. This term considers all the different time lags and indicates X to be Granger-causal for Y, if the average effect is significant. As shown earlier in this section, the criterion used for choosing the lasso penalty parameter λ is a crucial step, as it determines the degree of sparseness of the network. In [13], they vary λ in the range of $(k\ \lambda_{max},\ \lambda_{max})$, where k is a fraction, and then apply the BIC criterion in this range of λ, to choose the most consistent network. We use the same procedure and apply it to our synthetic data and compare it with the generated adjacency matrices. In our approach we use the existing package "Lars" for lasso, where lasso solution is found in steps and at each step, the parameters of the BIC criterion are already available; hence, λ is determined automatically. In addition, unlike group lasso, since the parameter values are readily available during the lasso solution, no further computational or time effort is needed.

Shojaie *et al*. [14] have proposed a truncating lasso (Tlasso) penalty for the estimation of graphical Granger models, where they require a large number of samples and also ignore all samples of further time points. In Tlasso, the coefficients are calculated at each time point till the truncation condition is satisfied. An iterative procedure is followed, where the old estimates are fed to the next iteration in order to estimate the new coefficients depending on the residual error. Although Tlasso exhibits a good performance when $n \geq p$, it gives a very poor performance when $n < p$. This is due to the fact that Tlasso does not take advantage of all the samples at different time points, but rather uses only a limited number of samples at certain time points. Since Tlasso can detect at most minimum of (n, p) edges, its performance deteriorates when $n < p$, which is typically

the case in a practical microarray data. The iterative procedure of Tlasso requires a large computational time to converge compared to the proposed method.

In [14], the authors choose the penalty parameter λ based on a percentile of a standard normal distribution. For the sake of completeness, we apply the methods of [13, 14] for $n = 60$ as well, to compare the performance, although it is not practical to have such a large number of samples for microarray data. The results for F1-measure, Precision, Recall, TP and FP rates are averaged over 300 networks for each n and given in Tables 6-16 and 6-17. The average Precision and Recall for each n are shown in Figure 6.9.

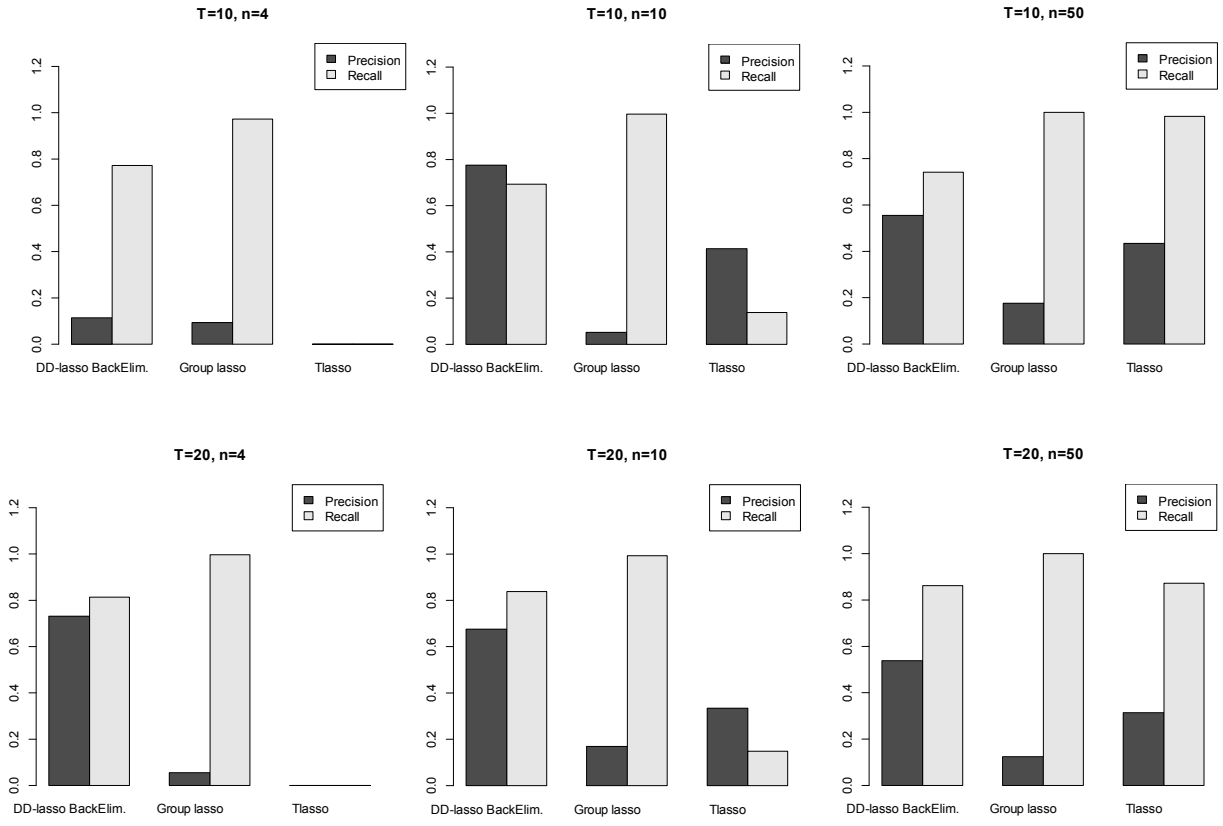

Figure 6.9 Bar plot of the average Precision and Recall

Table 6-16 Results for the F1-measure of Proposed DD-lasso, Group lasso and Tlasso

	n	DD-lasso with backward elimination	Group lasso	Tlasso
T=10	4	**0.267**(0.028)	0.165(0.02)	0(0)
	10	**0.731**(0.059)	0.098(0.015)	0.203(0.06)
	50	**0.633**(0.062)	0.2891(0.104)	0.601(0.051)
	60	**0.621**(0.06)	0.261(0.022)	0.573(0.046)
T=20	4	**0.768**(0.102)	0.103(0.019)	0(0)
	10	**0.745**(0.103)	0.282(0.074)	0.197(0.061)
	50	**0.655**(0.126)	0.219(0.056)	0.460(0.102)
	60	**0.637**(0.13)	0.299(0.02)	0.445(0.095)

Table 6-17 Results of P, R TP rate and FP rate for existing methods

		Group lasso				Truncating lasso			
	n	P	R	TP rate	FP rate	P	R	TP rate	FP rate
T=10	4	9.037%	97.396%	97.396%	39.267%	0	0	0	0
	10	5.1866%	99.814%	99.814%	73.573%	41.474%	13.676%	13.676%	0.763 %
	50	22.340%	99.963%	99.963%	14.5238%	43.437%	98.126%	98.126%	5.128%
	60	15.060%	100%	100%	22.322%	40.490%	98.941%	98.941%	5.797%
T=20	4	5.412%	99.732%	99.732%	70.975%	0	0	0	0
	10	16.646%	99.470%	99.470%	21.921%	33.359%	14.857%	14.857%	1.3278%
	50	13.067%	100%	100%	28.780%	31.449%	87.285%	87.285%	7.8179%
	60	17.607%	100%	100%	18.474%	29.818%	89.189%	89.189%	8.5892%

As seen from these figures and tables, group lasso has a very high TP rate; however, their FP rate is also very high. Hence, the overall F1 measure is low. Since in group lasso, unlike our approach, the number of covariates is $p \times d$, the performance deteriorates when the number of samples is not large enough. Tlasso underestimates the non-zero coefficients for $n < p$, resulting in a trivial solution of all zeros. However, it yields a good performance at $n = 50$, the number of samples they use in their paper. It is seen from the previous figures and tables that the proposed DD-lasso

surpasses the existing methods. The computational times for the various methods are given in Table 6-18.

Table 6-18 Computational time in seconds for the proposed and existing methods

	n	DD-lasso with backward elimination	Group lasso	Truncating lasso
T=10	4	37.19	**2.54**	17.69
	10	3.31	8.67	**1.34**
	50	**5.95**	16.77	76.13
	60	**7.30**	24.70	93.90
T=20	4	**3.49**	9.12	20.40
	10	3.98	19.82	**1.81**
	50	**11.86**	46.49	125.9
	60	**13.48**	55.01	109.40

It is seen from Table 6-18 that in the proposed DD-lasso, when $n = 4$ and $T = 10$, it takes more computational time than that for the remaining datasets. This is due to the fact that cross-validation technique is applied when $n = 4$ and $T = 10$, while mBIC2 criterion is applied for the remaining datasets. Since Tlasso does not use all the data points, but only the first few time points and is based on an iterative method, for $n = 10$, Tlasso method converges quickly leading to a small computational time. However, for large datasets, our proposed method outperforms the other methods in terms of the computational cost. For various settings of synthetic data, our proposed approach surpasses the existing techniques in terms of performance as well as complexity, and provides a more consistent framework for GRN reconstruction.

6.4 Results of Network Reconstruction Using Real data

The proposed approach is now applied on two real datasets. The first dataset resulted from an experiment on human HeLa cells, while the second is from an experiment to study yeast cell cycle.

6.4.1 Dataset 1

Our proposed DD-lasso with backward elimination is now applied to real microarray data extracted from human uterine cervical carcinoma cells [71], wherein the hela cell cycle synchronized by a double thymidine block has been examined. A thorough understanding of the

regulation of the cell cycle division is crucial for studies such as cancer development. The microarray data used is downloaded from (http://genome-www.stanford.edu/ Human - CellCycle/HeLa). In this dataset, samples are measured at time 0 and every hour for 46 hours, and hence 47 time points are available. There are two replicates that are averaged to get the values at time 0. Sambo *et al*. [66] extracted a subset of nine genes from the human cell cycle genes for which the regulatory network is determined in the BIOGRID database (www.thebiogrid.org), which represents the biological knowledge. They have proposed a search-based algorithm, called the CNET, which searches over the space of all possible graphs, to find the candidate graph with the highest score. The same nine-gene network has also been examined in [13] and [14]. Our proposed DD-lasso is now applied to these nine genes and the resulting network as well as that from the three previous algorithms are compared with the BIOGRID network. The nine genes are: CCNE1, CCNA2, CCNB1, CDC2, E2F1, PCNA, CDC6, RFC4, and CDKN3. The maximum delay d is set at 3, and we study as to whether a gene is affected by the other genes with up to 3 hours of delay between the effect and the response. Since the BIOGRID database is frequently updated according to new biological findings, we compare the resulting networks with the BIOGRID interaction network that was last updated in March 2012. The resulting network from our proposed DD-lasso as well as that from the previous methods are shown in Figure 6.10, and the results for Precision, Recall and F1-measure for the various methods calculated using the BIOGRID database, are given in Table 6-19. It is seen from this table that the proposed DD-lasso integrated with backward elimination results in a network that is most consistent with the biological knowledge as compared to that resulting from the existing methods.

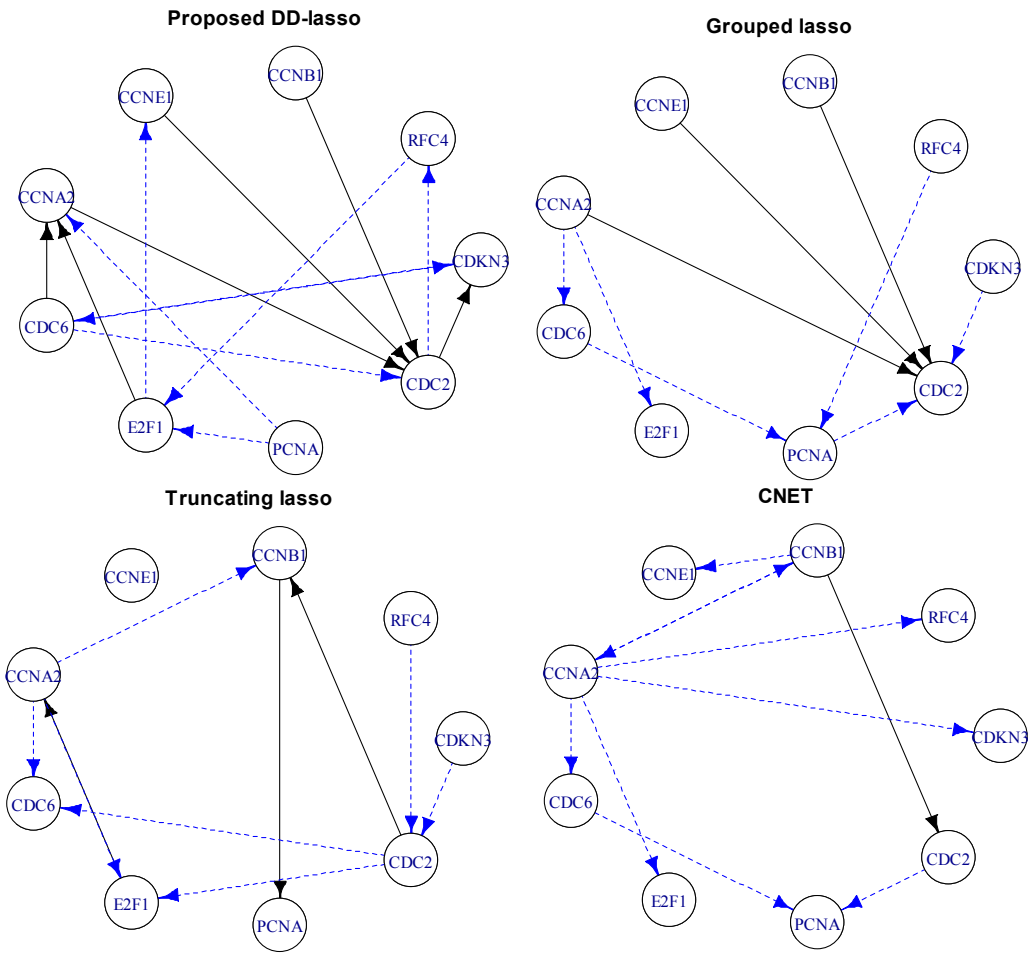

Figure 6.10 Hela cell cycle network, where true edges are solid lines, while false edges are dashed lines

Table 6-19 Results for the hela cell cycle

	Proposed DD-lasso with backward elimination	Group lasso [13]	Tlasso [14]	CNET [66]
P	42.85%	33.333%	30%	10%
R	54.54%	27.27%	27.27%	9.09%
F1	0.48	0.3	0.2857	0.0952

6.4.2 Dataset 2

We now apply our method to the microarray data of Spellman *et al*. [72], where yeast cell cycle regulation is studied. This microarray experiment was designed to create a list of yeast genes with

transcription levels expressed periodically within the cell cycle. In [72], the yeast cell cycle synchronized by alpha factor has been examined, where the samples are measured at time 0 and every 10 minutes, for a total of 18 time points to cover two complete cycles of cell division. Zoppoli *et al*. [73] extracted a subset of eleven genes from the yeast cell cycle genes. The GRN reconstruction in [73] is based on information theory approach, and the algorithm is called Time Delay-ARACNE. The eleven genes are CLN1, CLN2, CLN3, SWI4, SWI6, *MBP1,* CLB5, CLB6, SIC1, *CDC28,* and *Cdc6*, and are part of the G1 step of yeast cell cycle. Our proposed DD-lasso with backward elimination is applied to these eleven genes and the resulting network as well as that due to group lasso [13] and Zoppoli *et al*. [73] are compared with BIOGRID network. Since, only one sample is used, Tlasso [14] completely fails and cannot be applied to this real data set. The maximum delay *d* is set at 3, and we study as to whether a gene is affected by the other genes with up to 30 minutes of delay between the effect and the response. The resulting networks from the three methods are shown in Figure 6.11, and the results for Precision, Recall and F1-measure, using the BIOGRID database, are given in Table 6-20. It is seen from this table that the proposed DD-lasso integrated with backward elimination results in a network that is most consistent with the biological knowledge as compared to that resulting from the existing method.

Table 6-20 Results for the yeast cell cycle

	Proposed DD-lasso with backward elimination	Group lasso [13]	Time Delay-ARACNE [73]
P	70.37%	52.941%	76.470%
R	31.147%	29.50%	21.311%
F1	0.432	0.378	0.3333

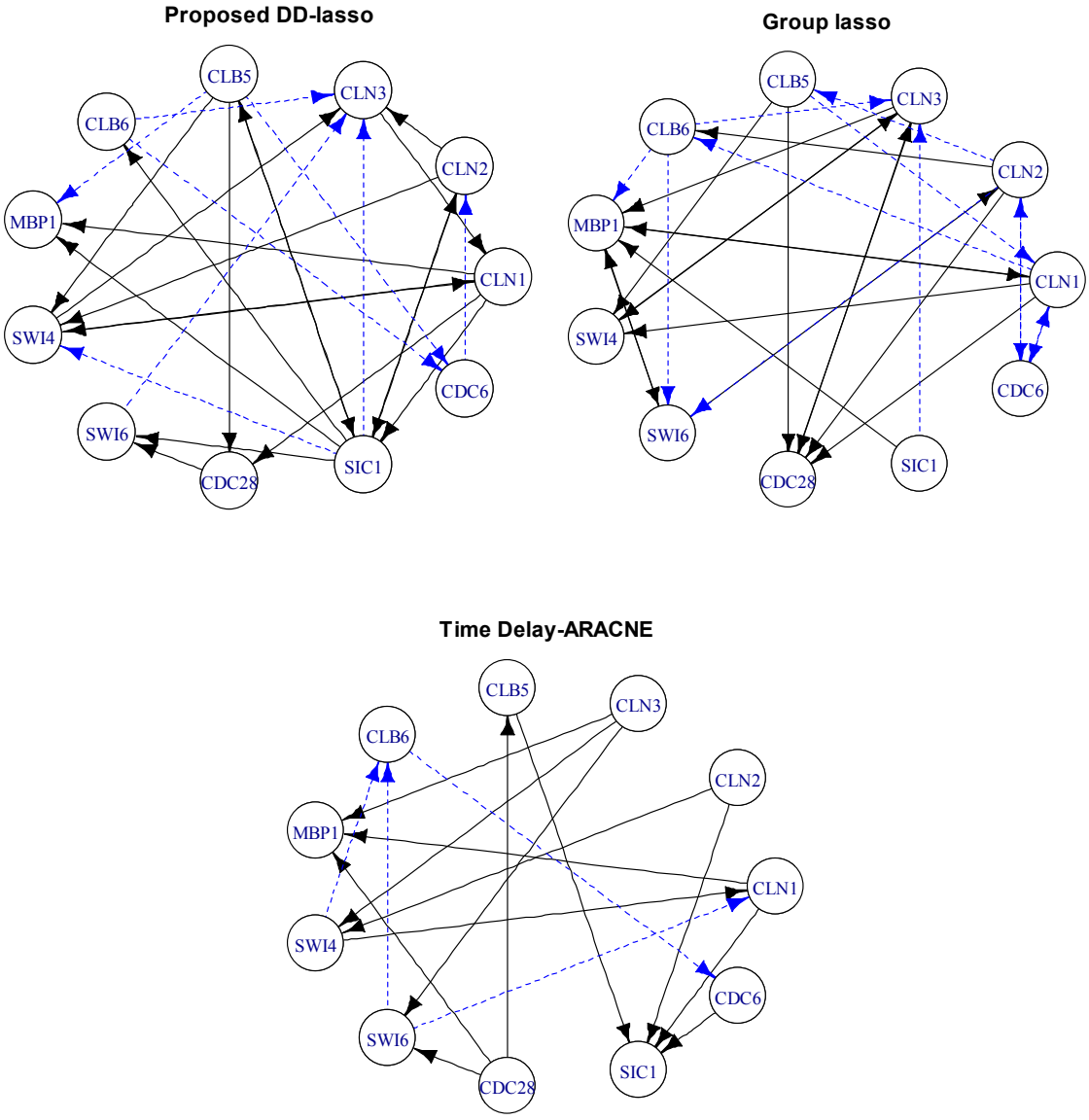

Figure 6.11 Yeast cell cycle network, where true edges are solid lines, while false edges are dashed lines

6.5 Summary

In this chapter, we have applied the proposed DD-lasso method to both synthetic and real data. The experimental results on the synthetic data have shown that the possible delays between genes are most suitably determined by applying Pearson cross-correlations, wherein the accuracy of the estimate of the delay improves with longer time series. Next, the interactions among genes have been modeled by VAR model, where the coefficients are estimated using the lasso technique. It has been shown that choosing the lasso penalty parameter based on mBIC2 criterion outperforms those based on cross-validation and BIC criterion. Using synthetic data, it has been shown that the proposed DD-lasso improves not only the Precision but also the Recall, and thus the overall F1-measure, compared to that due to lasso, and other existing methods. Moreover, the networks reconstructed from real data using DD-lasso have been shown to be more consistent with the biological knowledge as compared to that resulting from the existing methods. In addition, simulation studies of DD-lasso at other maximum delays, such as d=4, and d=5 have been carried out. These results have shown that, although the performance of DD-lasso can be slightly affected by overestimating d, it still has an overall performance that is superior to that of lasso and other previous methods.

Chapter 7
Conclusion

7.1 Concluding remarks

In this research work, we have addressed two main problems in microarray data analysis. The first problem has been concerned with identifying differentially expressed genes for time series data, whether single or multiple time-series, whereas the second one with modeling significant genes by reconstructing gene regulatory network for further understanding of the underlying biological processes.

A VSP method that identifies significant genes for one group time course microarray and for multiple time-series microarray data has been proposed. For each gene, the F-statistic is computed, whether RM F-statistic for single group time series or mixed design F-statistic for multiple group time series data. A moderation scheme has been proposed and applied to the given F-statistic, followed by carrying permutations in order to evaluate the p-value for each gene. Based on the p-values, the significance of each gene is determined and the most significant genes are identified. The experimental results on the synthetic data have shown that the pooled p-values procedure is able to detect more true positives than the gene-wise p-values method does, and hence, used for the analysis of microarray data. A new correction factor has been proposed to modify the F-statistic, wherein a different correction factor, is applied to each F-statistic for each gene. The new correction factor has been shown to outperform the other correction techniques in terms of the sensitivity. Furthermore, the proposed algorithm outperforms the existing time-series analysis techniques in terms of both the sensitivity and the specificity. Moreover, the proposed algorithm when applied to real data has been able to detect those genes identified as significant by previous techniques. In addition, it has been shown to identify other significant genes consistent with existing biological knowledge, but missed by other techniques. The proposed technique is fully automatic, does not rely on any prior assumptions and does not need any parameters to be set. Furthermore, the correction term that has been introduced is easily implementable and can be applied to any statistic.

The algorithm presented is carried out for one group time course microarray, and it extended to multi-biological groups. The algorithm presented can be employed as a step before gene clustering or reconstructing networks.

The inference of the underlying interactions between various genes is an ultimate goal for different scientists. A step toward this goal is to infer the gene regulatory networks using linear models. In the GRN reconstruction, the linear dynamic network architectures, which mostly benefit from the dynamic behavior found in the time-course data, has been studied. A challenging problem was the determination as to which genes and their lags are relevant, particularly when there is large number of genes and moderate sample size (relative to the number of genes and lags). An integrated solution has been proposed that infers various interactions between genes, while taking into consideration the varied possible lag for each gene, which has been termed DD-lasso. In this method, the possible delays between genes are first determined using cross-correlations, wherein the accuracy of the estimate of the delay improves with longer time series and higher Signal-to-Noise ratio (SNR). In order to calculate the cross- correlations, the microarray expression data of each gene is averaged over the samples, hence, the number of samples does not have much influence on the accuracy of the delay detection. Since the total number of samples of the covariates, $x_{i,t}$ is $N=n\times(T- d),$ the larger the maximum delay, d, examined, the less the number of samples used. Next, the interactions among genes have been modeled by VAR model, where the coefficients are estimated using the lasso technique. Regularization and variable selection are essential to infer parsimonious models that facilitate model interpretation. Just as in the case of previously reported works, if there are enough samples, it has been shown that choosing the lasso penalty parameter based on BIC criterion outperforms those based on cross-validation. In order to choose the appropriate lasso penalizing parameter λ, a modified BIC criterion for sparse solutions, mBIC2, has been employed. The proposed DD-lasso approach has been applied to a wide variety of synthetic data and to two common real datasets.

Although correlation has been used in the literature for testing linear dependencies, in our method correlation is used for a completely different purpose, namely, for time delay detection. The only purpose of the proposed delay detection scheme is to refine the input data to the algorithm that will

reconstruct the GRN. In this thesis, the algorithms used to reconstruct the GRN are lasso and adaptive lasso. The delay detection scheme has been integrated with lasso, since it is one of the very popular methods and can easily be implemented. Similarly, delay-detection can be integrated with smoothly clipped absolute deviation [74], Dantzig selector [75], elastic net [76], bridge regression [77] or any variable selection method that accepts delays as part of the GRN model.

We have compared our method with Group lasso [13] and Tlasso [14], since these two methods are also based on lasso and take into account the various time delays as we do in the present work. The main drawback of Tlasso is that it ignores many data points, thus not fully exploiting the dataset. Further, its performance is poor when n is small, for example at n=10, and completely fails when n is very small (n=4 or smaller). Group lasso can be applied for small n; however, its overall performance is inferior to that of the proposed DD-lasso. In our simulations, we have set d=3, which is the maximum delay used to generate the synthetic data. In addition, simulation studies of DD-lasso at other maximum delays, such as d=4, and d=5 have been carried out. These results have shown that, although the performance of DD-lasso can be slightly affected by overestimating d, it still has an overall performance that is superior to that of lasso and other previous methods. Our experiments have shown that the effect of noise with respect to the number of time points and samples is minimum.

Using synthetic data, it has been shown that the proposed DD-lasso improves not only the Precision but also the Recall, and thus the overall F1-measure, compared to that due to lasso, and other existing methods. Moreover, the networks reconstructed from real data using DD-lasso have been shown to be more consistent with the biological knowledge as compared to that resulting from the existing methods. Our proposed algorithm is able to detect the relationships between genes with various delays between them. In addition, since our proposed DD-lasso method is integrated with the existing package of 'Lars' [29], it can be efficiently and easily implemented. Our proposed technique is fully automatic, and does not rely on any prior assumptions. It surpasses the existing techniques in terms of the performance as well as the complexity and computational time, for most of the cases, and provides a more consistent framework for GRN reconstruction. The method successfully assists in understanding the gene interactions, thus providing valuable information to

the pharmaceutical and biotechnology industries for designing new drugs for complex diseases.

7.2 Scope for further investigation

The present work can be extended in the future in various ways. The proposed VSP method for identifying differentially expressed genes can be extended by identifying genes of some specific pattern, not only the genes that are differentially expressed. In addition, the proposed delay detection scheme can be applied to other gene regulatory network reconstruction methods such as elastic net and bridge regression and compare their performance with that of DD-lasso.

Furthermore, non-linear interaction can be considered in our models for enhancing the reconstructed models of gene regulatory networks and better mimicking the real underlying biological networks. The data insufficiency problem affects the accuracy of the modeling of GRNs using microarray data alone. Hence, in order to obtain more reliable networks the integration of diverse types of data with microarray data is a promising approach. The various types of data include information from scientific literature and biological databases (text-mining information), sequence information, data transcription factor (TF) binding data, gene functional annotations, Chip-on-chip data and protein–protein interaction data. Thus, we can design methods that deal with different types of biological information simultaneously, and expanding the system to protein interaction networks and eventually metabolic networks.

References

[1] G. K. Geiss, R. E. Bumgarner, B. Birditt, T. Dahl, N. Dowidar, D. L. Dunaway, H. P. Fell, S. Ferree, R. D. George, T. Grogan, J. J. James, M. Maysuria, J. D. Mitton, P. Oliveri, J. L. Osborn, T. Peng, A. L. Ratcliffe, P. J. Webster, E. H. Davidson, L. Hood and K. Dimitrov, "Direct multiplexed measurement of gene expression with color-coded probe pairs," *Nat Biotech*, vol. 26, pp. 317-325, 2008.

[2] Z. Wang, M. Gerstein and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nat. Rev. Genet.*, vol. 10, pp. 57-63, 2009.

[3] S. D. Peddada, E. K. Lobenhofer, L. Li, C. A. Afshari, C. R. Weinberg and D. M. Umbach, "Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference," *Bioinformatics,* vol. 19, pp. 834-841, 2003.

[4] J. D. Storey, W. Xiao, J. T. Leek, R. G. Tompkins and R. W. Davis, "Significance analysis of time course microarray experiments," *Proc. Natl. Acad. Sci. U. S. A.,* vol. 102, pp. 12837-12842, 2005.

[5] Y. Tai and T. Speed , "A multivariate empirical Bayes statistic for replicated microarray time course data," *Ann Statist,* vol. 34, pp. 2387-2412, 2006.

[6] C. Angelini, D. De Canditiis, M. Mutarelli and M. Pensky, "A Bayesian approach to estimation and testing in time-course microarray Experiments," *Statistical Applications in Genetics and Molecular Biology,* vol. 6, 2007.

[7] R. Opgen-Rhein and K. Strimmer, "Inferring gene dependency networks from genomic longitudinal data: a functional data approach." *Revstat,* vol. 4, pp. 53-56, 2006.

[8] A. de la Fuente, N. Bing, I. Hoeschele and P. Mendes, "Discovery of meaningful associations in genomic data using partial correlation coefficients," *Bioinformatics,* vol. 20, pp. 3565-3574, 2004.

[9] A. Wille, P. Zimmermann, E. Vranova, A. Furholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem and P. Buhlmann, "Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana," *Genome Biol.,* vol. 5, pp. R92, 2004.

[10] M. Bansal, G. Gatta and D. di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics,* vol. 22, pp. 815-822, 04/01/, 2006.

[11] R. Guthke, U. Moller, M. Hoffmann, F. Thies and S. Topfer, "Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection." *Bioinformatics,* vol. 21, pp. 1626-1634, 2005.

[12] X. Li, S. Rao, W. Jiang, C. Li, Y. Xiao, Z. Guo, Q. Zhang, L. Wang, L. Du, J. Li, L. Li, T. Zhang and Q. Wang, "Discovery of Time-Delayed Gene Regulatory Networks based on temporal gene expression profiling." *BMC Bioinformatics,* vol. 7, pp. 26, 2006.

[13] A. Lozano, N. Abe, Y. Liu and S. Rosset, "Grouped graphical Granger modeling for gene expression regulatory networks discovery," *Bioinformatics,* vol. 25, pp. 110-118, 2009.

[14] A. Shojaie and G. Michailidis, "Discovering graphical Granger causality using the truncating lasso penalty," *Bioinformatics,* vol. 26, pp. 517-523, 2010.

[15] E. van Someren, B. Vaes, W. Steegenga, A. Sijbers, K. Dechering and M. Reinders, "Least absolute regression network analysis of the murine osteoblast differentiation network," *Bioinformatics,* vol. 22, pp. 477-484, 2006.

[16] V. G. Tusher, R. Tibshirani and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proc. Natl. Acad. Sci. U. S. A.,* vol. 98, pp. 5116-5121, 2001.

[17] T. Park, S. Yi, S. Lee, S. Lee, D. Yoo, J. Ahn and Y. Lee, "Statistical tests for identifying differentially expressed genes in time-course microarray experiments." *Bioinformatics,* vol. 19, pp. 694-703, 2003.

[18] C. Angelini, L. Cutillo, D. De Canditiis, M. Mutarelli and M. Pensky, "BATS: a Bayesian user-friendly software for analyzing time series microarray experiments." *BMC Bioinformatics,* vol. 9, pp. 415, 2008.

[19] P. Baldi and A. D. Long, "A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes," *Bioinformatics,* vol. 17, pp. 509-519, 2001.

[20] B. Efron, R. Tibshirani, J. Storey and V. Tusher, "Empirical Bayes analysis of a microarray experiment," *Journal of the American Statistical Association,* vol. 96, pp. 1151-1160, 2001.

[21] P. Broberg, "Statistical methods for ranking differentially expressed genes," *Genome Biol.,* vol. 4, pp. R41, 2003.

[22] G. Smyth, "Linear models and empirical bayes methods for assessing differential expression in microarray experiments." *Statistical Applications in Genetics and Molecular Biology,* vol. 3, 2004.

[23] X. Cui, G. Hwang, J. Qiu, N. Blades and G. Churchill, "Improved statistical tests for differential gene expression by shrinking variance components estimates," *Biostat,* vol. 6, pp. 59-75, 2005.

[24] G. Wright and R. Simon, "A random variance model for detection of differential gene expression in small microarray experiments," *Bioinformatics,* vol. 19, pp. 2448-2455, 2003.

[25] J. D. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *Proc. Natl. Acad. Sci. U. S. A.,* vol. 100, pp. 9440-9445, 2003.

[26] S. Dudoit, Y. Yang, T. Speed and M. Callow, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," *Stat Sinica,* vol. 12, pp. 111-140, 2002.

[27] N. Holter, A. Maritan, M. Cieplak, N. Fedoroff and J. Banavar, "Dynamic modeling of gene expression data," *Proc. Natl. Acad. Sci. U. S. A.,* vol. 98, pp. 1693-1698, 2001.

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society,* vol. 58, pp. 267-288, 1994.

[29] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *Ann. Statist.,* vol. 32, pp. 407-499, 2004.

[30] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association,* vol. 101, pp. 1418-1429, 2006.

[31] O. ElBakry, M. O. Ahmad and M. N. S. Swamy, "Identification of differentially expressed genes for time-Course microarray data based on modified RM ANOVA," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 9, pp. 451-466, 2012.

[32] E. S. Edgington, *Randomization Tests.* CRC Press, 1995.

[33] E. R. Girden, *ANOVA: Repeated Measures.* SAGE, 1992.

[34] S. Geisser and S. W. Greenhouse, "An extension of Box's results on the use of the F distribution in multivariate analysis," *Ann. Math. Statist.,* vol. 29, pp. 885-891, 1958.

[35] I. Berkovits, G. R. Hancock and J. Nevitt, "Bootstrap resampling approaches for repeated measure designs: relative robustness to sphericity and normality violations," *Educational and Psychological Measurement,* vol. 60, pp. 877-892, 2000.

[36] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap.* Chapman & Hall/CRC, 1993.

[37] F. J. Anscombe, "Sequential estimation," *J. R. Statist. Soc,* vol. 15, pp. 1-29, 1953.

[38] N. Mukhopadhyay, S. Datta and S. Chattopadhyay, *Applied Sequential Methodologies: Real-World Examples with Data Analysis.* CRC Press, 2004.

[39] E. K. Lobenhofer, L. Bennett, P. L. Cable, L. Li, P. R. Bushel and C. A. Afshari, "Regulation of DNA replication fork genes by 17{beta}-estradiol," *Mol. Endocrinol.,* vol. 16, pp. 1215-1229, 2002.

[40] M. T. Lee, F. C. Kuo, G. A. Whitmore and J. Sklar, "Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations," *Proc. Natl. Acad. Sci. U. S. A.,* vol. 97, pp. 9834-9839, 2000.

[41] W. Rensink, S. Iobst, A. Hart, S. Stegalkina, J. Liu and C. Buell, "Gene expression profiling of potato responses to cold, heat, and salt stress," *Functional & Integrative Genomics,* vol. 5, 2005.

[42] J. Adamski, Z. Ma, S. Nozell and E. N. Benveniste, "17{beta}-Estradiol inhibits class II major histocompatibility complex (MHC) expression: influence on histone modifications and CBP recruitment to the class II MHC promoter," *Mol. Endocrinol.,* vol. 18, pp. 1963-1974, 2004.

[43] M. T. Stang, M. J. Armstrong, G. A. Watson, K. Y. Sung, Y. Liu, B. Ren and J. H. Yim, "Interferon regulatory factor-1-induced apoptosis mediated by a ligand-independent fas-associated death domain pathway in breast cancer cells," *Oncogene,* vol. 26, pp. 6420-6430, 2007.

[44] T. Dubois, S. Howell, E. Zemlickova and A. Aitken, "Identification of casein kinase Ialpha

interacting protein partners," *FEBS Lett.,* vol. 517, pp. 167-171, 2002.

[45] J. Dejmek, A. Safholm, C. Kamp Nielsen, T. Andersson and K. Leandersson, "Wnt-5a/Ca2+-induced NFAT activity is counteracted by Wnt-5a/Yes-Cdc42-casein kinase 1{alpha} signaling in human mammary epithelial cells," *Mol. Cell. Biol.,* vol. 26, pp. 6024-6036, 2006.

[46] E. Woo, D. G. Jeong, M. Lim, S. Jun Kim, K. Kim, S. Yoon, B. Park and S. Eon Ryu, "Structural insight into the constitutive repression function of the nuclear receptor Rev-erbβ" *J. Mol. Biol.,* vol. 373, pp. 735-744, 2007.

[47] W. Tang, M. Norlin and K. Wikvall, "Regulation of human CYP27A1 by estrogens and androgens in HepG2 and prostate cells," *Arch. Biochem. Biophys.,* vol. 462, pp. 13-20, 2007.

[48] F. Martínez-Arribas, D. Agudo, M. Pollán, F. Gómez-Esquer, G. Díaz-Gil, R. Lucas and J. Schneider, "Positive correlation between the expression of X-chromosome *RBM* genes (*RBMX, RBM3, RBM10*) and the proapoptotic *Bax* gene in human breast cancer," *J. Cell. Biochem.,* vol. 97, pp. 1275-1282, 2006.

[49] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.,* vol. 20, pp. 53-65, 1987.

[50]O. ElBakry, M.O. Ahmad and M.N.S. Swamy," Inference of Gene Regulatory Networks with Variable Time Delay from Time-Series Microarray Data," accepted with minor modifications in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

[51] R. Opgen-Rhein and K. Strimmer, "From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data," *BMC Systems Biology,* vol. 1, pp. 37, 2007.

[52] A. Dobra, C. Hans, B. Jones, J. R. J. R. Nevins, G. Yao and M. West, "Sparse graphical models for exploring gene expression data," *Journal of Multivariate Analysis,* vol. 90, pp. 196-212, 2004.

[53] G. Koh, H. F. C. Teong, M. Clement, D. Hsu and P. S. Thiagarajan, "A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk," *Bioinformatics,* vol. 22, pp. 271-280, 2006.

[54] C. Granger, "Investigating Causal Relations by Econometric Models and Cross-spectral Methods," *Econometrica,* vol. 37, pp. 424-438, 1969.

[55] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 24, pp. 320-327, 1976.

[56] H. Meyr and G. Spies, "The structure and performance of estimators for real-time estimation of randomly varying time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 32, pp. 81-94, 1984.

[57] J. S. Bendat and A. G. Piersol, *Random Data :Analysis and Measurement Procedures.* Hoboken, N.J.: Wiley, 2010.

[58] J. Yao, C. Chang, M. Salmi, Y. Hung, A. Loraine and S. Roux, "Genome-scale cluster

analysis of replicated microarrays using shrinkage correlation coefficient," *BMC Bioinformatics,* vol. 9, pp. 288, 2008.

[59] D. Zhu, Y. Li and H. Li, "Multivariate correlation estimator for inferring functional relationships from replicated genome-wide data," *Bioinformatics,* vol. 23, pp. 2298-2305, 2007.

[60] R. S. Bowden and B. R. Clarke, "A single series representation of multiple independent ARMA processes," *Journal of Time Series Analysis,* vol. 33, pp. 304-311, 2012.

[61] S. Chand., "Goodness of fit and lasso variable selection in time series analysis", thesis University of Nottingham 2011.

[62] H. Zou, T. Hastie and R. Tibshirani, "On the "degrees of freedom" of the lasso," *Annals of Statistics,* vol. 35, pp. 2173-2192, 2007.

[63] M. Bogdan, J. K. Ghosh and R. W. Doerge, "Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci," *Genetics,* vol. 167, pp. 989-999, 2004.

[64] F. Frommlet, F. Ruhaltinger, P. TwarÃg and M. Bogdan, "Modified versions of Bayesian information criterion for genome-wide association studies," *Comput. Stat. Data Anal.,* vol. 56, pp. 1038-1051, 2012.

[65] A. Prekopa, "Boole-Bonferroni inequalities and linear programming," *Oper. Res.,* vol. 36, pp. 145-162, 1988.

[66] F. Sambo, B. Di camillo and G. Toffolo, "CNET: an algorithm for Reverse Engineering of Causal Gene Networks," *NETTAB2008,* 2008.

[67] R. Albert, "Scale-free networks in cell biology," *J. Cell. Sci.,* vol. 118, pp. 4947-4957, 2005.

[68] D. E. Featherstone and K. Broadie, "Wrestling with pleiotropy: Genomic and topological analysis of the yeast gene expression network," *Bioessays,* vol. 24, pp. 267-274, 2002.

[69] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A. -. Barabasi, "The large-scale organization of metabolic networks," *Nature,* vol. 407, pp. 651-654, 2000.

[70] R. Albert and A. Barabási, "Topology of Evolving Networks: Local Events and Universality," *Phys. Rev. Lett.,* vol. 85, pp. 5234-5237, 2000.

[71] M. L. Whitfield, G. Sherlock, A. J. Saldanha, J. I. Murray, C. A. Ball, K. E. Alexander, J. C. Matese, C. M. Perou, M. M. Hurt, P. O. Brown and D. Botstein, "Identification of genes periodically expressed in the human cell cycle and their expression in tumors," *Molecular Biology of the Cell,* vol. 13, pp. 1977-2000, 2002.

[72] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, "Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell,* vol. 9, pp. 3273-3297, 1998.

[73] P. Zoppoli, S. Morganella and M. Ceccarelli, "TimeDelay-ARACNE: Reverse engineering of

gene networks from time-course data by an information theoretic approach," *BMC Bioinformatics,* vol. 11, pp. 154, 2010.

[74] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association,* vol. 96, pp. 1348-1360, 2001.

[75] E. Candes and T. Tao, "The Dantzig selector: statistical estimation when p is much larger than n," *The Annals of Statistics,* vol. 35, pp. 2313-2351, 2007.

[76] H. Zou and T. Hastie, "Regularization and variable selection via the Elastic Net," *Journal of the Royal Statistical Society ,* vol. 67, pp. 301-320, 2005.

[77] W. Fu, "Penalized Regressions: The bridge versus the lasso," *Journal of Computational and Graphical Statistics,* vol. 7, pp. 397-416, 1998.

[78] O. Elbakry, M.O. Ahmad and M.N.S. Swamy, "Inference of Gene Regulatory Networks from Time-Series Microarray Data", *Proc. 8th IEEE International NEWCAS,* 2010.

Appendix

The R Code of the Proposed Methods

1. The VSP method for the identification of differentially expressed genes

```
VSP<-function(dt,nsamp, npr, nt)
{
d<-dim(dt)
ng<-d[1]  #from Sim48
nc<-d[2]-1    #number of columns
nF<-numeric(ng)
p<-numeric(ng)
acc<-numeric(ng)
ytot<-numeric(ng)##Y totoal
sstb<-numeric(ng) #res error excpet the term of Yj
sro<-numeric(ng) #treatment
sto<-numeric(ng) #res error
yi<-numeric(nrep)#Sum for three replicates
y<-array(0,dim=c(nsamp,ng,nt))
r<-array(0,dim=c(nsamp,nt))
tb<-matrix(nrow = ng, ncol=npr)
T<-numeric(nt)
for(j in 1:ng)
```

```
{
g1<-dt[j,] #read row
ii<-seq(1,((nt*nrep)-nrep+1), by = nrep)
for(k in 1:nt)
  {
  i<-ii[k]
  y[,j,k]<-as.numeric(as.character(g1[1,(i+1):(i+nsamp)]))
  }
yi<-rowSums(y[,j,])/6
ytot[j]<-sum(yi)/nrep
sstb[j]<-sum((y[,j,]-ytot[j])^2)-6*sum((yi-ytot[j])^2)
T<-colSums(y[,j,])  ##Yj
sto[j]<-nrep*sum((T/nrep-ytot[j])^2)
sro[j]<-sstb[j]-sto[j] #add correction factor
}##endfor
#shrinkage toward median
cor<-median(sro)
lamda<-sro/(sro+cor)
dd<-lamda*sro+(1-lamda)*cor
nF<-2*sto/(dd)
for(j in 1:ng)
{
```

```
########permuatation
  for(k in 1:npr) ##check for first 100 permutations
  {
  for(i in 1:nsamp){
  r[i,]<-sample(y[i,j,],nt)
  }
  T<-colSums(r)
  st<-nrep*sum((T/nrep-ytot[j])^2)
  srr<-sstb[j]-st
  la<-srr/(srr+cor)
  sr<-la*srr+(1-la)*cor #lamda for each gene
  tb[j,k]<-2*st/sr
  }##endfor
}##endfor
```

2. The DD-lasso method for the reconstruction of gene regulatory networks

```
DD.lasso<-function(dat,del){
nm <- dim(dat)
##dat in format nsamp,ng,nt
nsamp<- nm[1]
ng<- nm[2]
nt<- nm[3]
##############3d correlation matrix##want to reconstruct the delay values
```

```
In<-array(0,dim=c(ng,ng,del))## delay time points
##Take average of dat and correlate
X<-matrix(nrow=ng,ncol=nt)
for(i in 1:nt){
X[,i]<-colMeans(dat[,,i])
}#endfor
###In[,,1]<-R #one time lag
#lag fist gene in gen and correlate with the rest and so on
#need to compute correlation for replicated data
for(k in 1:del)#numbering for delay
{
 for(i in 1:ng)#numbering for each delayed gene
 {
 vec<-X[i,1:(nt-k)]# (k) delay, zero padding
 vec2<-X[,(1+k):(nt)]
 In[i,,k]<-cor(t(vec2), vec) ##In row X(t-1) or X(t-2) Column X(t)
 }#endfor
}#endfor
D<-matrix(nrow = ng, ncol=ng)
for(i in 1:ng)#numbering for each delayed gene
 {
 for(j in 1:ng)
```

```
  {
  D[j,i]<-which.max(abs(In[i,j,])) #Delay between two gene#adjust row and column
  }#endfor
}#endfor
###In D, row is X(t) with the delayed t-1, ot t-2 or t-3
##Delay the X(t) according to D, then apply lasso
CC<-matrix(data=0,nrow=ng,ncol=ng)#matrix of coefficients
for(j in 1:ng)#Get model for each column
  {
  num<-ng #Number of genes included in the model
  y<-c(dat[,j,(del+1):nt])##The gene to be modeled
  le<-nt-(max(del))
  X2<-array(0,dim=c(num,le*nsamp)) ##Without AR term
  for(i in 1:num) #Get the delayed gene
  {
  ###According to each D[j,i] delay i
  dd<-D[j,i]
  X2[i,]<-c(dat[,i,(del-dd+1):(nt-dd)]) #shifted by delay value
  }#endfor
  w<-lasso.back(t(X2),y) ##
  CC[j,]<-w  ##
}#endfor
```

```
return(CC)

}#endfunction
```