Modeling and Characterization of Dynamic Changes in Biological Systems from Multi-platform Genomic Data

Bai Zhang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Yue Wang, Chair
Jason Xuan
William T. Baumann
Ge Wang
Chang-Tien Lu

September 13, 2011
Arlington, Virginia

Keywords: differential dependency networks, structural changes in graphical models, biological networks, echo state networks, DNA copy number changes

Modeling and Characterization of Dynamic Changes in Biological Systems from Multi-platform Genomic Data

Bai Zhang

(ABSTRACT)

Biological systems constantly evolve and adapt in response to changed environment and external stimuli at the molecular and genomic levels. Building statistical models that characterize such dynamic changes in biological systems is one of the key objectives in bioinformatics and computational biology. Recent advances in high-throughput molecular and genomic profiling technologies such as gene expression and copy number microarrays provide ample opportunities to study cellular activities at the individual gene and network levels. The aim of this dissertation is to formulate mathematically dynamic changes in biological networks and DNA copy numbers, to develop machine learning algorithms to learn these statistical models from high-throughput biological data, and to demonstrate their applications in systems biological studies.

The first part (Chapters 2 – 4) of the dissertation focuses on the dynamic changes taking placing at the biological network level. Biological networks are context-specific and dynamic in nature. Under different conditions, different regulatory components and mechanisms are activated and the topology of the underlying gene regulatory network changes. We report a differential dependency network (DDN) analysis to detect statistically significant topological changes in the transcriptional networks between two biological conditions. Further, we formalize and extend the DDN approach to an effective learning strategy to extract structural

changes in graphical models using ℓ_1-regularization based convex optimization. We discuss the key properties of this formulation and introduce an efficient implementation by the block coordinate descent algorithm. Another type of dynamic changes in biological networks is the observation that a group of genes involved in certain biological functions or processes coordinate to response to outside stimuli, producing distinct time course patterns. We apply the echo stat network, a new architecture of recurrent neural networks, to model temporal gene expression patterns and analyze the theoretical properties of echo state networks with random matrix theory.

The second part (Chapter 5) of the dissertation focuses on the changes at the DNA copy number level, especially in cancer cells. Somatic DNA copy number alterations (CNAs) are key genetic events in the development and progression of human cancers, and frequently contribute to tumorigenesis. We propose a statistically-principled *in silico* approach, Bayesian Analysis of COpy number Mixtures (BACOM), to accurately detect genomic deletion type, estimate normal tissue contamination, and accordingly recover the true copy number profile in cancer cells.

Acknowledgments

First, I would like to thank my adviser, Dr. Yue Wang, for his guidance, support, and trust during the course of my Ph.D. study. It is a great honor for me to have the opportunity to work with him closely for the past five years. The joy and enthusiasm Dr. Wang has for his research was contagious and motivational, and I am also thankful for the excellent example he has provided not only as a successful researcher but also as a person of great character.

Further, I would like to thank Dr. Jason Xuan for his expert guidance, constant help, and numerous inspiring discussions. I want to express my great gratitude to my Ph.D committee members Dr. William T. Baumann, Dr. Ge Wang, and Dr. Chang-Tien Lu for their invaluable guidance and insightful suggestions on my dissertation. I am also very grateful for the tremendous help on my dissertation research from Dr. David J. Miller and Dr. Huai Li. I enjoyed every moment working with every member of the Computational Bioinformatics and Bio-Imaging Laboratory. Special thanks go to Guoqiang Yu and Ye Tian for their selfless contributions in our collaborative research. Also, I want to thank Yuanjian Feng and Yitan Zhu for their generous help in so many different ways. I am very thankful for Chen Wang and Tsung-Han Chan for many inspiring and fun discussions.

I owe my deepest gratitude to my parents, Jingcheng Zhang and Yongrui Bai, for their love, support and encouragement. Finally and most importantly, I want to thank my wife, Yi Liu, for her love, trust, and support in these years, which have made this dissertation possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Motivation

Biological systems constantly evolve and adapt in response to changed environment and external stimuli. The ability to react, adjust, and select quickly is crucial to an organism's survival. In evolutionary biology, the change in the inherited traits of a population of organisms and adaptation to their new habitat take place over many successive generations. In short time scale, biological systems are also frequently challenged by their external environmental changes and inputs, and are forced to react to such changes at the molecular level. Such dynamic changes at the molecular level in biological systems are the focus of this dissertation.

Dynamic changes are common in gene regulatory networks. Gene regulatory networks are context-specific and dynamic in nature [1, 2]. Under different conditions, different regulatory components and mechanisms are activated and the topology of the underlying gene

regulatory network changes. For example, in response to diverse conditions in the yeast, transcription factors alter their interactions and rewire the signaling networks [3]. Transcriptional networks also exhibit network topological changes between disease and normal conditions, or across different stages of cell development. Such biological network changes are very informative and provide great biological insights. For example, a deviation from normal regulatory network topology may reveal the mechanism of pathogenesis [4], and the genes that undergo the most network topological changes may serve as biomarkers for the disease state or as targets for drug discovery or therapeutic intervention.

Another type of dynamic changes in gene regulatory networks is that a group of genes involved in certain biological functions or processes coordinate to response to outside stimuli, producing distinct time course patterns. Understanding and modeling the mechanisms that orchestrate the activities of genes and proteins in cells are the key goals in systems biology studies [5]. For example, in [6], it has been shown that messenger RNA levels of 800 genes vary periodically in yeast to work cooperatively to maintain the cell cycle. Analyzing and modeling the time course patterns of these cell cycle-regulated genes provides great insight into cell cycle regulation. Outside stimuli can also trigger dynamic changes in gene regulatory networks. For instance, the injection of cardiotoxin into the mouse muscle damages the muscle tissue, and inducts the staged muscle regeneration [7]. Genes in damage muscle cells first initiate necrosis of the damaged tissue and activation of an inflammatory response, and then activate myogenic cells to proliferate, differentiate, and fuse, leading to new myofiber [8]. In this complex process, hundreds of genes interact and coordinate to carry out muscle regeneration, characterized by the time course patterns of the messenger RNA levels of these genes.

Further, such dynamic changes can also be observed at the DNA level, especially in cancer cells. DNA copy number change is an important form of structural variation in human genomes. Germline copy number variants (CNVs) are associated with phenotype variations

and somatic copy number alterations (CNAs) are considered hallmarks of tumorigenesis. The coverage of copy number changes varies from a few hundred to several million nucleotide bases, and somatic CNAs in tumors exhibit highly complex patterns as compared with germline CNVs [9]. Although copy number analysis indicates monoclonal origin of lethal metastatic prostate cancer, metastatic cancer cells from different sites of the same patient still carry distinctive DNA copy number profiles [10]. Such observations suggest the evidence of frequent mutations in cancer cell DNAs and genomic evolution of cancer cells in the metastatic process.

1.1.2 Microarray Technology

Recent advances in high-throughput genomic technologies such as gene expression microarrays provide ample opportunities to study cellular activities at the individual gene expression and network levels. Microarray gene expression profiling simultaneously measures the expression levels of tens of thousands of genes under different experimental conditions, enabling studies on the phenotypic outcomes of certain treatment responses, disease progression, and developmental stages and the underlying gene expression patterns functionally associated with these phenotypes. These technologies also present new demands and challenges for data analysis to extract meaningful statistical and biological information from high throughput and high-dimensional data [1]. These data analysis tasks include signal pre-processing, clustering, visualization, classification, gene biomarker identification, and gene network modeling.

Similar progress has been made in measuring DNA copy numbers. The advent of oligonucleotide-based single nucleotide polymorphism (SNP) arrays provides high-density and allelic-specific genomic profile and enables researchers to study copy number changes in a genome-wide scale. Affymetrix offers several DNA analysis arrays for single nucleotide polymorphism

(SNP) genotyping and copy number variation (CNV) analysis, most notably GeneChip Human Mapping 100K Array Set, GeneChip Human Mapping 500K Array Set, Genome-Wide Human SNP Array 5.0, and Genome-Wide Human SNP Array 6.0. The new Affymetrix Genome-Wide Human SNP Array 6.0 features 1.8 million genetic markers, including more than 906,600 single nucleotide polymorphisms (SNPs) and more than 946,000 probes for the detection of copy number variation. Such high resolution DNA copy number data enable researchers to detect relatively short DNA copy number alterations, and at the same time demand efficient algorithms to bridge the gap between experimental data and biological knowledge.

1.2 Research Topics

1.2.1 Differential Dependency Networks

Significant efforts have been made to acquire data under different conditions and to construct static networks that can explain various gene regulation mechanisms. However, gene regulatory networks are dynamic and condition-specific; under different conditions, networks exhibit different regulation patterns accompanied by different transcriptional network topologies. Thus, an investigation on the topological changes in transcriptional networks can facilitate the understanding of cell development or provide novel insights into the pathophysiology of certain diseases, and help identify the key genetic players that could serve as biomarkers or drug targets.

Here we report a differential dependency network (DDN) analysis to detect statistically significant topological changes in the transcriptional networks between two biological conditions. We propose a local dependency model to represent the local structures of a network by a set of conditional probabilities. We develop an efficient learning algorithm to learn

the local dependency model using the Lasso technique. A permutation test is subsequently performed to estimate the statistical significance of each learned local structure. We expect DDN to emerge as an important bioinformatics tool in transcriptional network analyses. While we focus specifically on transcriptional networks, the DDN method we introduce here is generally applicable to other biological networks with similar characteristics.

1.2.2 Learning Structural Changes in Graphical Models

Inspired by the structural / topological changes in biological networks, we further formulate network structural changes as a general machine learning problem, which can be applied to areas other than biological networks. For instance, in web search or collaborative filtering, useful information can be acquired by observing how certain events (*e.g.*, launch of an advertisement campaign) trigger changes in dependence patterns of search keywords or preference for products reflected in the associated structural changes. Graphical models are widely used in scientific and engineering research to represent conditional independence structures between random variables. In many controlled experiments, environmental changes or external stimuli can often alter the conditional dependence between the random variables, and potentially produce significant structural changes in the corresponding graphical models. Therefore, it is of great importance to be able to detect such structural changes from data, so as to gain novel insights into where and how the structural changes take place and help the system adapt to the new environment. Here we report an effective learning strategy to extract structural changes in Gaussian graphical model using ℓ_1-regularization based convex optimization. We discuss the properties of the problem formulation and introduce an efficient implementation by the block coordinate descent algorithm.

1.2.3 Echo State Networks

Further, we study the problem of modeling time course data in gene regulatory networks. A gene regulatory network is a collection of genes in a cell that interact, often nonlinearly, with each other (indirectly through their RNAs and proteins). The challenges of modeling gene regulatory networks lie in the nonlinearity of the dynamic systems and high level of inherent noise. Echo state networks (ESNs) are a novel form of recurrent neural networks (RNNs) that provide an efficient, powerful computational model approximating nonlinear dynamical systems. A unique feature of an ESN is that a large number of neurons (the "reservoir") are used, whose synaptic connections are generated randomly, with only the connections from the reservoir to the output modified by learning. Why a large, randomly generated, fixed RNN gives such excellent performance in approximating nonlinear systems is still not well understood. In Chapter 4, we apply random matrix theory to examine the properties of random reservoirs in ESNs under different topologies (sparse or fully-connected) and connection weights (Bernoulli or Gaussian). We quantify the asymptotic gap between the scaling factor bounds for the necessary and sufficient conditions previously proposed for the echo state property. We then show that the state transition mapping is contractive with high probability when only the necessary condition is satisfied, which corroborates and thus analytically explains the observation that in practice one obtains echo states when the spectral radius of the reservoir weight matrix is smaller than 1.

1.2.4 Bayesian Analysis of Copy Number Mixtures

Finally, we examine the DNA copy number changes in cancer cells. Copy number change is an important form of structural variations in human genomes. Somatic copy number alterations can cause overexpression of oncogenes and loss of tumor suppressor genes in tumorigenesis. Recent development of SNP array technology has facilitated studies on copy

number changes in a genome-wide scale with high resolution. We report here a statistically-principled *in silico* approach, Bayesian Analysis of COpy number Mixtures (BACOM), to accurately estimate genomic deletions and normal tissue contamination in cancer cell. Tumor samples often consist of mixed cancer and normal cells. Such tissue heterogeneity will cause inaccurate analysis of copy number changes in clinical samples and could significantly confound subsequent marker identification and diagnostic classification rooted in specific cells. BACOM is applied to the normal tissue contamination correction problem to recover the true copy number profile in cancer cells. Moreover, utilizing the discrepancy in the estimated normal tissue fractions and heterogeneity of tumor samples, we search for the evidence of the sequence of genomic change events in metastatic prostate cancer samples.

1.3 Outline of the Dissertation

The remainder of this dissertation proceeds as follows. In Chapter 2, we report a differential dependency network (DDN) analysis to detect statistically significant topological changes in the transcriptional networks between two biological conditions. This is followed by a general machine learning framework to learn structural changes in graphical models in Chapter 3. In Chapter 4, we study the theoretical properties of echo state networks with applications to gene regulatory network time course data modeling. In Chapter 5, we propose Bayesian analysis of copy number mixtures to correct normal cell contamination and characterize tumor evolution. We conclude this dissertation with summary of contributions and future work in Chapter 6.

Chapter 2

Differential Dependency Network Analysis to Identify Topological Changes in Biological Networks

2.1 Introduction

Recent advances in high-throughput genomic technologies such as gene expression microarrays provide ample opportunities to study cellular activities at the individual gene expression and network levels. Microarray gene expression profiling simultaneously measures the expression levels of tens of thousands of genes under different experimental conditions, enabling studies on the phenotypic outcomes of certain treatment responses, disease progression, and developmental stages and the underlying gene expression patterns functionally associated with these phenotypes. These technologies also present new demands and challenges for data analysis to extract meaningful statistical and biological information from high throughput and high-dimensional data [1]. These data analysis tasks include signal pre-processing,

clustering, visualization, classification, gene biomarker identification, and gene network modeling.

Gene network modeling and analysis attempts to explain the mechanisms that orchestrate the activities of genes and proteins in cells, and is one of the key goals in systems biology studies [5]. Several computational approaches have been proposed to model gene regulatory networks [11], such as Bayesian networks [12–14], probabilistic Boolean networks [15], state-space models [16] and network component analysis [17]. These methods attempt to construct a static network that can explain various gene regulation programs. While the inference of transcriptional networks using data from composite conditions could sometimes be contradictory due to changes in the underlying topology, most network learning algorithms assume an invariant network topology [13, 15, 16].

However, gene regulatory networks are context-specific and dynamic in nature [1, 2]. Under different conditions, different regulatory components and mechanisms are activated and the topology of the underlying gene regulatory network changes. For example, in response to diverse conditions in the yeast, transcription factors alter their interactions and rewire the signaling networks [3]. Therefore, some methods have been proposed to learn condition-specific transcriptional networks in yeast [18, 19]. It is important to focus on the topological changes in transcriptional networks between disease and normal conditions, or across different stages of cell development. For example, a deviation from normal regulatory network topology may reveal the mechanism of pathogenesis [4], and the genes that undergo the most network topological changes may serve as biomarkers for the disease state or as targets for drug discovery or therapeutic intervention.

Several methods have been proposed to utilize network topology information to carry out various bioinformatics tasks. Liu et al. introduced a topology-based cancer classification method [20], where correlation networks were first constructed and later used to perform classification. Fuller et al. developed weighted gene co-expression network analysis strate-

gies, using single network analysis and differential network analysis, to identify physiologically relevant modules [21]. Qiu et al. proposed an ensemble dependence model to detect the dependence changes of gene clusters between cancer and normal conditions for cancer classification, and further extended the dependence model to dependence networks [22, 23]. Wei and Li introduced a Markov random field model for network-based analysis of genomic data that utilizes the known pathway structures to identify differentially expressed genes and sub-networks [24, 25].

In this chapter, we discuss differential dependency network (DDN) analysis as a new method to model and detect the statistically significant topological changes in transcriptional networks between two conditions. This discussion is based on the work proposed in [26]. We use local dependency models to characterize the dependencies of genes in the network and extract and represent local network substructures. Local dependency models decompose the entire network into a series of local networks, which serve as the basic network elements for subsequent statistical testing. Local dependency models select the number of dependent variables automatically by the Lasso method [27], and thereby learn the local network structures. Subsequently, we perform permutation tests on the local dependency models under two conditions and assign the p-values to the local structures. It may seem straightforward to construct an entire network under each condition and compare the differences between the two networks [21, 23]. However, in realistic applications this approach runs into the difficulty that the network structure learning can be inconsistent with a limited number of data samples.

When applied to the very high dimensional data produced by gene expression microarrays, the properties of the data impose additional constraints and complications [1]. The detection procedure proposed here assures the statistical significance of the detected network topological changes by performing a permutation test on individual local structures. We also pinpoint "hot spots" in the network where the genes exhibit network topological changes between two

conditions above a given significance level. Lastly, we extract and visualize the DDN, *i.e.*, the sub-networks exhibiting the most significant topological changes. We demonstrate the usefulness of the proposed method on both simulated and real microarray data. Tested on a simulation dataset, the proposed algorithm accurately captured the genes with network topological changes. When applied to the estrogen-dependent T-47D estrogen receptor-positive (ER+) breast cancer cell line dataset and normal adult rat mammary glands exposed to excess E2 *in utero* dataset, the DDN analysis obtained biological meaningful and promising results.

2.2 Preliminaries

2.2.1 Probabilistic Graphical Models and Dependency Networks

The probabilistic graphical models are diagrammatic representations of probability distributions of a set of random variables. In a probabilistic graphical model, each node represents a random variable (or a group of random variables), and edges (either directed or undirected) express dependent relationships between these variables [28].

Probabilistic graphical models have been widely used to represent biological networks. Because microarray data are very noisy, the probabilistic nature of graphical models can capture the noise in the data and intrinsic uncertainties in the models. Further, the diagrammatic representations of graphical models naturally visualize the relationships of genes, which can facilitate new insights and motivate new biological hypotheses. Typical examples of probabilistic graphical models are Bayesian networks, Markov networks, linear Gaussian networks, and dependency networks [28, 29].

Dependency networks were first proposed to encode and learn probabilistic relationships by Heckerman [30]. Unlike Bayesian networks, the graph of a dependency network can be cyclic.

Dependency networks are considerably easier to learn from data. More specifically, given a set of random variables $\mathbf{X} = \{X_1, X_2, \cdots, X_M\}$, a dependency network for $\mathbf{X}$ is modeled by a set of local conditional probability distributions, one for each node given its parents, denoted as $\mathbf{Z}_i$, which satisfies

$$P(X_i|\mathbf{Z}_i) = P(X_i|\mathbf{X}_{-i}), \tag{2.1}$$

where $\mathbf{X}_{-i} = \{X_1, X_2, \cdots, X_{i-1}, X_{i+1}, \cdots, X_M\}$ and $\mathbf{Z}_i \subset \mathbf{X}_{-i}$. $P(X_i|\mathbf{Z}_i)$ also represents the local structure of node X_i, *i.e.* the relationship of node X_i and its parents $\mathbf{Z}_i$ on the graph. Dependency networks are constructed by learning each conditional probability distribution independently, resulting in significant efficiency gains when compared with Bayesian network approach.

2.2.2 Graph Structure Learning and ℓ_1-regularization

Efficiently learning the structure of graph models is often very challenging in general. It has been proved that learning the structure of a Bayesian network is a NP-hard problem [31]. In gene regulatory network modeling, the network structure is of great interest, but learning the network structure is especially difficult in this case because the samples are usually very limited and the random variables, e. g. genes and proteins, are numerous.

Recently, ℓ_1-regularization has drawn great interest in statistics and machine learning community [27, 32–36]. Penalty or constraint on ℓ_1-norm of the regression coefficients has two very useful properties: sparsity and convexity. ℓ_1-norm constraint tends to make some coefficients exactly zeros, leading to a parsimonious solution, which naturally performs varible selection or sparse linear model estimation. Further, convex nature of ℓ_1-norm constraint makes the problem computationally tractable, which can be solved readily by many existing convex optimization methods [37].

Lasso is a linear regression minimizing squared error loss with ℓ_1-norm constraint, proposed

by Tibshirani [27]. The theoretical analysis of Lasso shows that sparsity pattern of the Lasso estimator is asymptotically identical to the true sparsity pattern under certain conditions [35]. On the algorithmic side, a very efficient algorithm, least angle regression (LARS), was proposed and can be modified to solve Lasso problems. LARS has very nice geometric interpretation and also gives the whole solution path with the same computational complexity as ordinary least squares, which makes it computationally appealing.

The idea of ℓ_1-regularization has also been applied to graph structure learning. For instance, ℓ_1-regularization was used to learn the structures of linear Gaussian networks [33], Markov networks [38], and directed acyclic graphs [34].

2.3 Method

2.3.1 Local Dependency Model in DDN

Inspired by the formulation of dependency networks, we propose a local dependency model to describe the dependencies of genes in a transcriptional network. Unlike a conventional dependency network approach, where there is only one conditional probability distribution for each node given its parents, our local dependency model allows more than one conditional probability distributions for each node. Mathematically, suppose there are M genes in the network of interest, and the dependencies of gene i on other genes are formulated by a set of conditional probabilities,

$$\mathbb{P}_i = \{P(X_i|\mathbf{Z}_{i,1}, \mathbf{Z}_{i,2}, \cdots, \mathbf{Z}_{i,s_i})\}, i = 1, 2, \cdots, M, \tag{2.2}$$

where $\mathbf{Z}_{i,1}, \mathbf{Z}_{i,2}, \cdots, \mathbf{Z}_{i,s_i}$ are some subsets of $\mathbf{X}_{-i}$ and s_i is the number of conditional probabilities for random variable X_i. We use X_i to refer both to the expressions of gene i and to its corresponding node on the graph. This modification is primarily based on the following

considerations. First, our goal is not to construct the entire network that represents the full joint distribution of all variables, rather we wish to model the local structures for further statistical testing. Second, many genes are highly correlated and the data points are very limited when extracting most biological networks. Through our experiments, we found that the conventional approach misses some meaningful dependency connections in data-sparse situations. For example, regulator genes R1 and R2 have the same target gene A, and the expression patterns of R1, R2 and A are highly correlated. When the data points are few, the standard approach may only select one of the dependencies, for instance, gene A on gene R1, even though the dependency of gene A on gene R2 is only slightly less significant than the dependency of gene A on gene R1. However, the dependencies of gene A on genes R1 and R2 are both important, and we want to keep the rich structural information for later step to assess the topological changes. Therefore, to retain more meaningful local structure information, instead of selecting the best local structure, we select a set of sufficiently good local structures for further statistical testing. We achieve this goal by allowing each node to be modeled by more than one conditional probability distribution.

2.3.2 Local Structure Learning

Now the question is how to learn the local dependency models for DDN. We consider a linear regression model in which the variable X_i is predicted by a linear function of $\mathbf{Z}_i$

$$X_i = \boldsymbol{\beta}^T \mathbf{Z}_i + \epsilon_i, \tag{2.3}$$

where $\mathbf{Z} \in \{\mathbf{Z}_{i,1}, \mathbf{Z}_{i,2}, \cdots, \mathbf{Z}_{i,s_i}\}$ is a column vector of random variables, $\boldsymbol{\beta}$ is a column vector of unknown parameters, T presents matrix transposition. The random error ϵ_i is independent of $\mathbf{Z}_i$ and is assumed to have normal distribution $N(0, \sigma_i^2)$. The local conditional probability is therefore

$$P(X_i|\mathbf{Z}_i) = N(\boldsymbol{\beta}^T \mathbf{Z}_i, \sigma_i^2). \tag{2.4}$$

Learning the structure of the local dependency model requires the selection of a $\mathbf{Z}_i$ that shows good predictability of $\mathbf{X}_i$. Given a predefined maximum size of $\mathbf{Z}_i$, K, we examine all C_{M-1}^K combinations of the elements in $\mathbf{X}_{-i}$ with size K. K can be empirically set to a positive integer between 1 and $M-1$. When $K=1$, the proposed local dependency model only considers pairwise relationships. When $K=M-1$, the proposed local dependency model is equivalent to standard dependency networks.

Suppose one K-combination of $\mathbf{X}_{-i}$ is $\{X_{k_1}, X_{k_2}, \cdots, X_{k_K}\}$, where $k_1, \cdots, k_K \in \{1, 2, \cdots, i-1, i+1, \cdots, M\}$, and there are N expression samples. Lower case letter $x_i(j)$ denotes the j-th sample value taken by the variable X_j, $j = 1, 2, \cdots, N$. We perform a ℓ_1 constrained regression of X_i on $\mathbf{Z}_i = \{X_{k_1}, X_{k_2}, ..., X_{k_K}\}$

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \text{argmin} \left\{ \sum_{j=1}^{N} \left(x_i(j) - \sum_{l=1}^{K} \beta_l x_{k_l}(j) \right)^2 \right\}, \text{ s.t. } \sum_{l=1}^{K} |\beta_l| \leq t. \tag{2.5}$$

The above equation is known as the Lasso estimator, which minimizes ℓ_2 norm loss with constraint on the ℓ_1 norm of $\boldsymbol{\beta}$. The nature of ℓ_1 constraint tends to make some coefficients in $\hat{\boldsymbol{\beta}}_{\text{Lasso}}$ exactly zero, and hence it automatically selects a subset of features and leads to a simpler model that avoids overfitting the data, and therefore usually has better generalization performance. The parameter $t \leq 0$ controls the amount of shrinkage that is applied to the estimates. In our software implementation, parameter t is determined by 5-fold cross-validation. Solving the Lasso estimation is a convex optimization problem, and can be solved very efficiently. We adopt LARS method to solve this problem. The detailed procedure of LARS can be found in [32].

We also use a prescreening strategy to reduce the computational burden. We first regress X_i on $\mathbf{Z}_i = \{X_{k_1}, X_{k_2}, \cdots, X_{k_K}\}$, using the ordinary least square method

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = \text{argmin} \left\{ \sum_{j=1}^{N} \left(x_i(j) - \sum_{l=1}^{K} \beta_l x_{k_l}(j) \right)^2 \right\}. \tag{2.6}$$

If the corresponding mean square error (MSE) is above a predetermined threshold T, which means X_i cannot be accurately predicted by the subset $\{X_{k_1}, X_{k_2}, \cdots, X_{k_K}\}$, then subset $\{X_{k_1}, X_{k_2}, \cdots, X_{k_K}\}$ will be discarded. If the MSE is below T, we will then perform the ℓ_1 constrained regression of X_i.

We perform the above prescreening and the local structure learning with the Lasso on each of K-combinations of $\mathbf{X}_{-i}$, and obtain predictor sets $\mathbf{Z}_{i,1}$, $\mathbf{Z}_{i,2}$, $\cdots$, $\mathbf{Z}_{i,s_i}$ and the conditional probability distributions $\mathbb{P}_i = \{P(X_i|\mathbf{Z}_{i,1}), P(X_i|\mathbf{Z}_{i,2}), \cdots, P(X_i|\mathbf{Z}_{i,s_i})\}$ for node X_i.

To measure how well variables $\mathbf{Z}_i$ can predict X_i, or how well the local dependency model fits gene expression microarray data, we further introduce the definition of coefficient of determination (COD)

$$COD = \frac{var[X_i] - var[X_i - f_{X_i|\mathbf{Z}_i}(\mathbf{Z}_i)]}{var[X_i]}, \tag{2.7}$$

where $var[\cdot]$ is the variance of the random variable and $f_{X_i|\mathbf{Z}_i}(\cdot)$ is the best function in a given function class that minimizes the residual variance. COD has been successfully used in non-linear signal processing and probabilistic Boolean network inference [15, 39]. Here we only use linear functions, and $var[X_i - f_{X_i|\mathbf{Z}_i}(\mathbf{Z}_i)]$ is an estimate of σ_i^2.

Here we use a simple example to illustrate how to use the structural learning procedure described above to obtain $\mathbb{P}_i = \{P(X_i|\mathbf{Z}_{i,1}), P(X_i|\mathbf{Z}_{i,2}), \cdots, P(X_i|\mathbf{Z}_{i,s_i})\}$ for node X_i. Suppose the node under examination is X_1, and the predefined maximum size of $\mathbf{Z}_i$ is $K = 2$. We examine all C_{M-1}^2 combinations of the elements in $\mathbf{X}_{-1}$: $\{X_2, X_3\}$, $\{X_2, X_4\}$, ..., $\{X_2, X_{M-1}\}$, ..., $\{X_{M-3}, X_{M-2}\}$, $\{X_{M-3}, X_{M-1}\}$, $\{X_{M-2}, X_{M-1}\}$. In the prescreening step, we use ordinary least square regression to regress X_1 on each of these C_{M-1}^2 combinations, and calculate the MSE for each regression model. For the simplicity of discussion, suppose the true connection is between X_1 and X_2, and $X_3, ..., X_{M-1}$ are independent with X_1. Then after this prescreening step, only $\{X_2, X_3\}$, $\{X_2, X_4\}$, ..., $\{X_2, X_{M-1}\}$ are retained (since X_3, ..., X_{M-1} do not have any information at all to predict X_1). Subsequently, we apply Lasso

regression to regress X_1 to each of the remaining $\{X_2, X_3\}$, $\{X_2, X_4\}$, ..., $\{X_2, X_{M-1}\}$. As we mentioned earlier, the ℓ_1 constraint in Lasso has the sparse property and selects the true predictors, and for each of these sets, only X_2 will have non-zero coefficients, the coefficients for X_3, X_4, ..., X_{M-1} will be zero. Then the identified substructure in this simple example is $\mathbb{P}_1 = \{P(X_1|X_2)\}$.

2.3.3 Detection of Statistically Significant Topological Changes

To detect the statistically significant network topological changes between two experimental conditions, we assume there are M genes in the network of interest, and N_1 samples from condition 1 and N_2 samples from condition 2. We further denote the datasets from two conditions by $\mathbf{D}^{(m)} = [\mathbf{x}^{(m)}(1), \mathbf{x}^{(m)}(2), \cdots, \mathbf{x}^{(m)}(N_m)]$, where superscript $^{(m)}$ indicates condition m, $m = 1, 2$. The bold face lower case letter $\mathbf{x}^{(m)}(j)$ denotes the column vector $[x_1^{(m)}(j), x_2^{(m)}(j), \cdots, x_M^{(m)}(j)]^T$, where lower case letter $x_i^{(m)}(j)$ denotes the j-th sample value taken by variable X_i under condition m, and the superscript T denotes matrix transpose.

By applying the learning procedure to datasets $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, respectively, we obtain $\mathbb{P}_i^{(1)} = \{P(X_i|\mathbf{Z}_{i,1}^{(1)}), P(X_i|\mathbf{Z}_{i,2}^{(1)}), \cdots, P(X_i|\mathbf{Z}_{i,s_i^{(1)}}^{(1)})\}$ under condition 1 and $\mathbb{P}_i^{(2)} = \{P(X_i|\mathbf{Z}_{i,1}^{(2)}), P(X_i|\mathbf{Z}_{i,2}^{(2)}), \cdots, P(X_i|\mathbf{Z}_{i,s_i^{(2)}}^{(2)})\}$ under condition 2 for each node i, $i = 1, 2, ..., M$. Then we take the union of the local structures learned under two conditions

$$\mathbb{P}_i = \mathbb{P}_i^{(1)} \cup \mathbb{P}_i^{(2)}, \qquad i = 1, 2, \cdots, M, \tag{2.8}$$

for further statistical testing.

Continuing the simple example in the previous sub-section, we now illustrate how to generate the set $\mathbb{P}_i$ for statistical testing. We first apply the structural learning procedure to data under condition 1, and obtain $\mathbb{P}_1^{(1)} = \{P(X_1|X_2)\}$ as demonstrated earlier. Then we apply the same structural learning procedure to data under condition 2. Suppose under condition

2, new dependence patterns (local structures) emerge: the connection between X_1 and X_2 remains the same, and two new connections X_1 – X_3 and X_1 – X_4 appear. Then $\mathbb{P}_1^{(2)}$ could take the form (one of the possible forms) $\mathbb{P}_1^{(2)} = \{P(X_1|X_2), P(X_1|X_3, X_4)\}$. Hence, the set of local structures for testing is $\mathbb{P}_1^{(1)} \cup \mathbb{P}_1^{(2)} = \{P(X_1|X_2), P(X_1|X_3, X_4)\}$. Then we want to assess how significant $P(X_1|X_2)$ and $P(X_1|X_3, X_4)$ are different between two conditions. In other words, what is the statistical significance that the connections X_1 – X_2, X_1 – X_3 and X_1 – X_4 are different given the data under these two conditions.

For each conditional probability distribution in $\mathbb{P}_i$, $i = 1, 2, ..., M$, for instance, $P(X_i|\mathbf{Z}_i) \in \mathbb{P}_i$, we perform a permutation test to assess how significantly it is different between two conditions. Given samples $\{[x_i^{(1)}(j^{(1)}),\ \mathbf{z}_i^{(1)}(j^{(1)})]^T,\ j^{(1)} = 1,\ 2,\ \cdots,\ N_1\}$ under the first condition and $\{[x_i^{(2)}(j^{(2)}),\ \mathbf{z}_i^{(2)}(j^{(2)})]^T,\ j^{(2)} = 1,\ 2,\ \cdots,\ N_2\}$ under the second condition, we calculate $COD^{(1)}$ and $COD^{(2)}$, using Equation (2.7). A test statistic $\hat{\theta}$ is defined by the absolute difference of the coefficients of determination under two conditions

$$\hat{\theta} = |COD^{(1)} - COD^{(2)}|. \tag{2.9}$$

We want to test the null hypothesis, H_0, of no difference between $P^{(1)}(X_i|\mathbf{Z}_i)$ and $P^{(2)}(X_i|\mathbf{Z}_i)$. We first combine $\{[x_i^{(1)}(j^{(1)}), \mathbf{z}_i^{(1)}(j^{(1)})]^T, j^{(1)} = 1, 2, \cdots, N_1\}$ and $\{[x_i^{(2)}(j^{(2)}), \mathbf{z}_i^{(2)}(j^{(2)})]^T, j^{(2)} = 1, 2, \cdots, N_2\}$, and then randomly permute samples from two conditions and divide the data into two sets of N_1 and N_2 samples, respectively. We perform the above procedure B times, where B is set to 5000 in our software implementation, and calculate $\hat{\theta}_b^*$, $b = 1, 2, \cdots, B$ according to Equation (2.9). An estimate of the achieved significance level (ASL) of the test is

$$ASL = \frac{\sum_{b=1}^{B} \mathbf{1}_{\{\hat{\theta}_B^* \geq \hat{\theta}\}}}{B}, \tag{2.10}$$

where the random variable $\hat{\theta}_B^*$ is generated by permutation and $\mathbf{1}_{\{\hat{\theta}_B^* \geq \hat{\theta}\}}$ denotes the indicator function, which takes 1 when $\hat{\theta}_B^* \geq \hat{\theta}$ and 0 otherwise. The smaller the value of ASL, the stronger the evidence against H_0 is. Equation (2.10) also is an estimate of the p-value. The

detailed permutation procedure is described in [40]. This detection procedure is performed on every local structure in $\mathbb{P}_i$, $i = 1, 2, \cdots, M$, and each local structure is assigned a p-value.

2.3.4 Identification of "Hot Spots" in the Network and Extraction of the DDN

Given a user defined p-value cutoff, we obtain a set of statistically significant differential local structures. The nodes in these differential local structures are identified as "hot spots" in the network, which are the genes undergoing topological changes defined by a specified significance level. These genes may correspond to the genes in disease- or process-related pathways.

DDN is the focused sub-network that exhibits the topological changes. We consider a connection to exist from each element in $\mathbf{Z}_i$ to X_i under one specific condition if the variance of $P(X_i|\mathbf{Z}_i)$ is below the user-defined threshold T for that condition (see Section 2.5.1 for discussions on the selection of T). We use different colors to represent connections appearing under different conditions. DDN provides a way to visualize the topological changes, and when applied to disease studies, DDN extracts and focuses on the disease-related pathways that may contribute to the understanding of the mechanism of the disease.

We summarize the algorithm of differential dependency network analysis in a flowchart, as illustrated in Figure 2.1.

2.4 Software Development

DDN (Differential Dependency Network) is a caBIG® (cancer Biomedical Informatics Grid) analytical tool for detecting and visualizing statistically significant topological changes in

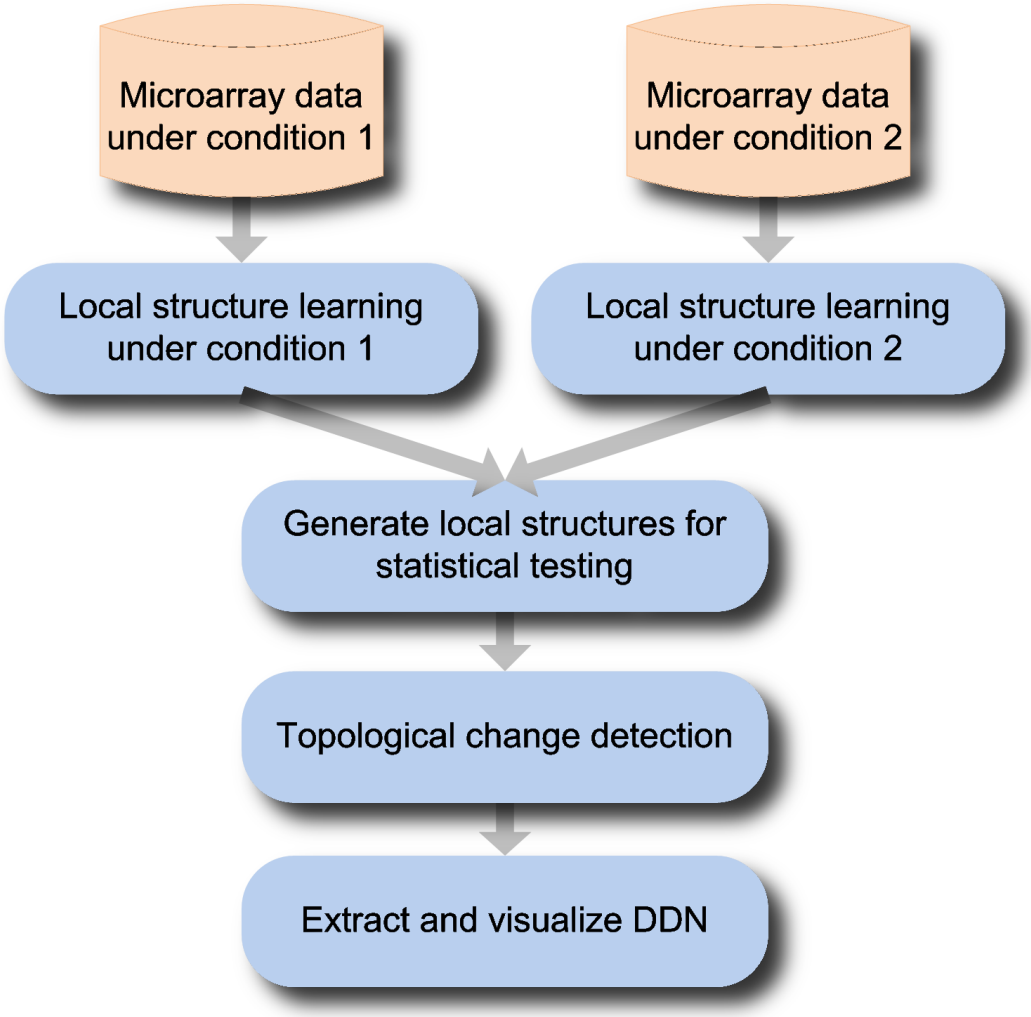


Figure 2.1: The flowchart of differential dependency network analysis.

transcriptional networks representing two biological conditions. Developed under caBIG®'s In Silico Research Centers of Excellence (ISRCE) Program, DDN enables differential network analysis and provides an alternative way for defining network biomarkers predictive of phenotypes. DDN also serves as a useful systems biology tool for users across biomedical research communities to infer how genetic, epigenetic or environment variables may affect biological networks and clinical phenotypes. Besides the standalone Java application, we have also

developed a Cytoscape plug-in, CytoDDN, to integrate network analysis and visualization seamlessly.

2.4.1 DDN: A caBIG® Standalone Java Package

caBIG® (cancer Biomedical Informatics Grid®) initiative was launched by the National Cancer Institute as a 21st century information system to transform the way scientists and physicians do cancer research. caBIG® is an open source network that enables members of the cancer community researchers, physicians, and patients to share data and knowledge.

As a caBIG® analysis tool, we developed the DDN software under compatibility requirements. The package contains 4 major components: C shared library created by Matlab compiler; Java Driver with GUI; C Driver and Matlab compiled JNI (Java Native Interface) shared library depended on C shared library.

Figure 2.2 is a screen shot of the graphical user interface of DDN Java package. The GUI prompts users to input the gene expression data of the two comparing conditions and the corresponding list of genes. In the parameter settings, users can define the highest order of the network (Max_K),

2.4.2 CytoDDN: A Cytoscape Plug-in

Cytoscape is an open source bioinformatics software platform, first launched in 2002 [41], for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data. The platform has already become a widely-adopted tool in the bioinformatics community.

Cytoscape includes a flexible plug-in architecture that enables developers to add extra functionality beyond that provided in the core. We developed the CytoDDN plug-in using Java

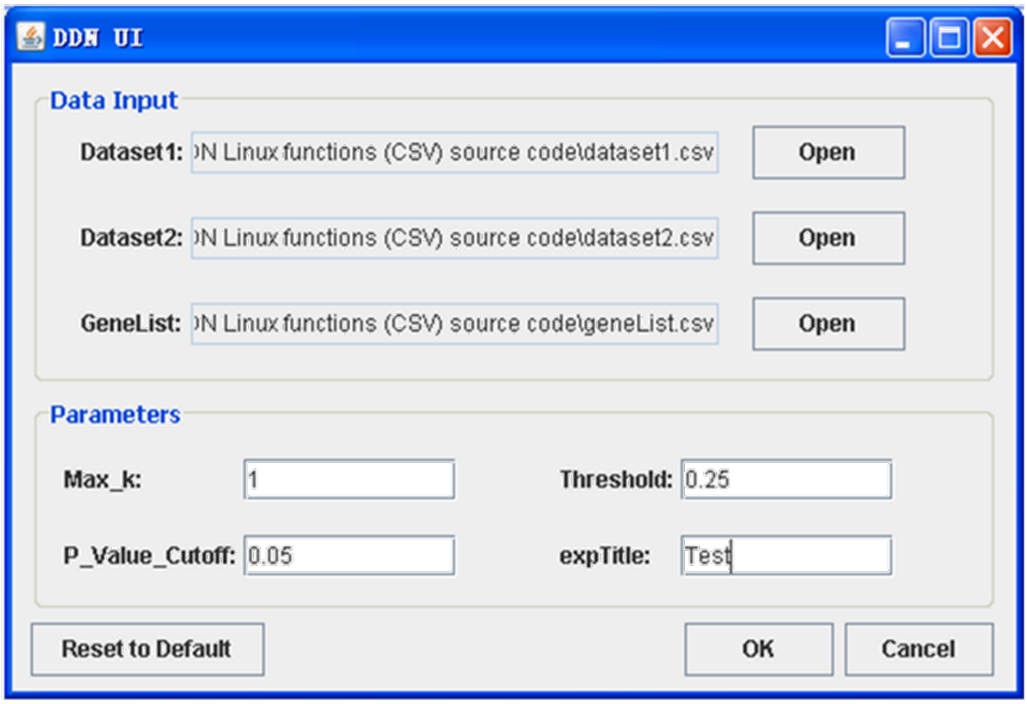


Figure 2.2: User interface of standalone DDN Java package.

programming language and the Cytoscape open API under a GNU Public License (GPL).

CytoDDN focuses on small-to-moderate scale problems that can be calculated in several seconds. Therefore, we set $K = 1$ in CytoDDN for fast processing. For higher order network detection, a longer computation time is expected and the standalone package is recommended.

The CytoDDN plug-in is platform independent and easy to install by placing the jar file in fold "plugins" under the installation directory of Cytoscape. After installation, the "DDN" menu will show up in the "Plugins" drop down menu. Figure 2.3 is a screen shot of CytoDDN.

Clicking on the DDN menu will invoke the plug-in and create the CytoDDN window, where the program prompts the user to select the files containing gene expression data and related gene list. Values in the input boxes are: default p-value cutoff, threshold, and number of permutations; these can be changed when necessary. The "Perform DDN analysis" button

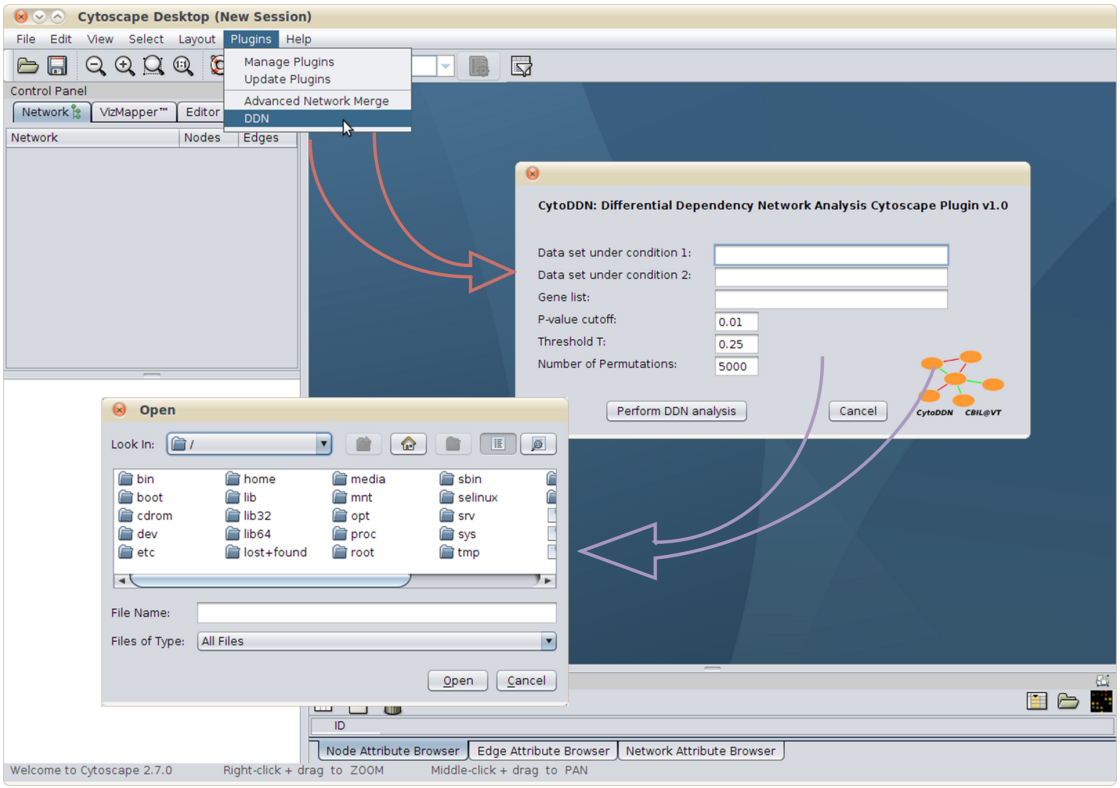


Figure 2.3: A screen shot of CytoDDN.

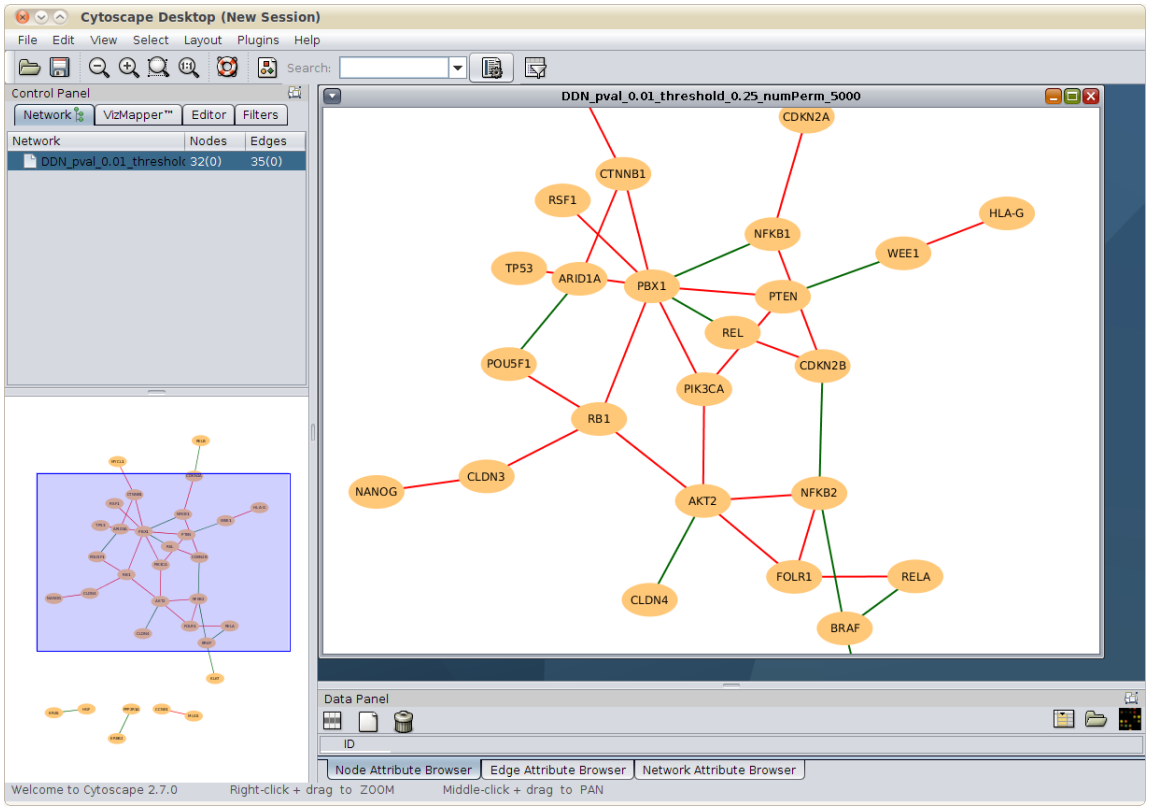


Figure 2.4: Illustration of a CytoDDN result.

starts the analysis, after which the DDN will be visualized in Cytoscape, as shown in Figure 2.4.

With the DDN solution visualized in Cytoscape, users can further adjust the layout of the network and export the information for other analysis.

2.5 Experiments and Results

2.5.1 A Simulation Experiment

We first use a simulation experiment to illustrate the concept of differential dependency network and analyze the DDN algorithm using the known ground truth.

Experiment Data

We used the software SynTReN [42] to generate one simulation dataset of a sub-network drawn from an existing signaling network in *Saccharomyces cerevisiae*. Then we changed part of network topology and used SynTReN to generate another dataset according to this modified network. SynTReN is a network generator that creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates experimental data. Network topologies are generated by selecting subnetworks from previously described regulatory networks. Interaction kinetics are modeled by equations based on Michaelis-Menten and Hill kinetics.

The network topology under two conditions is shown in Figure 2.5. The network contains 20 nodes that represent 20 genes. The black lines indicate the regulatory relationships that exist under both conditions. The red and green lines are the regulatory relationships that only exist under conditions 1 and 2, respectively. The sub-network comprised of nodes MBP1_SWI6, CLB5, CLB6, PHO2, FLO1, FLO10 and TRP4 and green and red lines is the DDN that our algorithm tries to identify from expression data.

Application of DDN Analysis

The parameters for our algorithm are: threshold T is 0.25, p-value cutoff is 0.01 and the maximum size of $\mathbf{Z}_i$, K, is 2. Figure 2.6 shows the DDN between the two conditions extracted by the proposed algorithm. The DDN shows network topological changes and the genes involved therein. The red lines in Figure 2.6 represent the connections that exist only under condition 1, and the green lines represent the connections that exist only under condition 2. Compared with the known network topology shown in Figure 2.5, the proposed algorithm correctly identified and extracted all the nodes with topology changes and 9 of 10 differential connections, with only the connection between PHO2 and TRP4 under condition 1 falsely

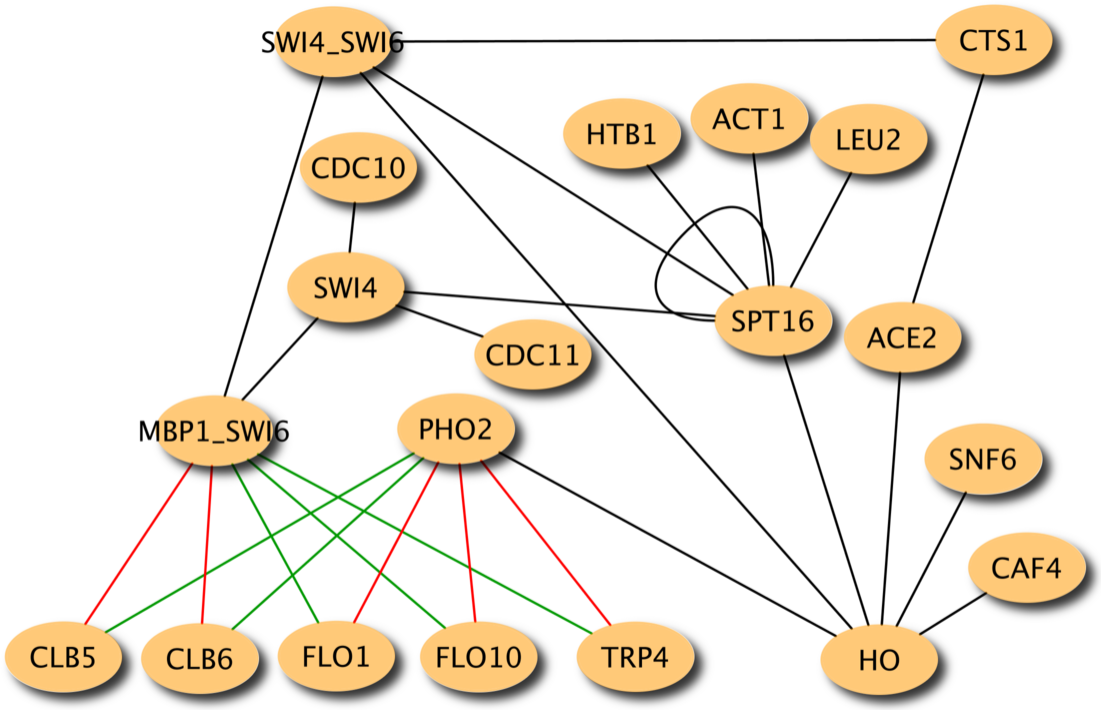


Figure 2.5: The network topology under two conditions in the simulation study. Nodes in the network represent genes. Lines in the network indicate regulatory relationships between genes. The black lines are the regulatory relationships that exist under both conditions. The red and green lines represent the regulatory relationships that exist only under condition 1 and under condition 2, respectively. The differential dependency network between the two conditions is the sub-network comprised of nodes MBP1_SWI6, CLB5, CLB6, PHO2, FLO1, FLO10 and TRP4 and green and red lines.

missed, and the connection between PHO2 and SWI4 under condition 1 and the connection between MBP1-SWI6 and SWI4 under condition 2 falsely detected. Moreover, our algorithm picked up all genes involved in topological changes, including some genes that did not show a significant difference in fold-change or t-tests, such as CLB6, FLO1 and MBP1_SWI6. This indicates that our algorithm can successfully detect these interesting genes using their topological information, even though the means of their expressions did not change substantially

between the two conditions. Therefore, this method is able to identify biomarkers that cannot be picked up by traditional gene ranking methods, providing a complimentary approach for biomarker identification problem.

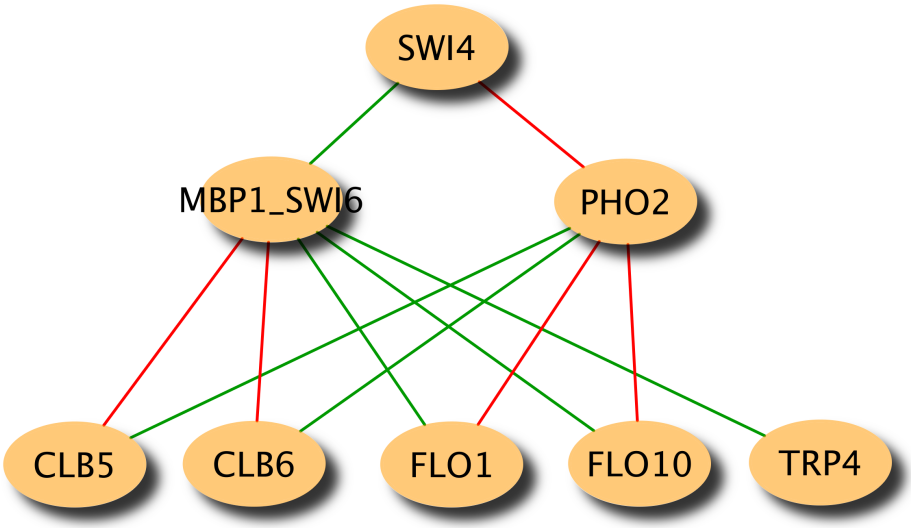


Figure 2.6: The differential dependency network extracted by the proposed algorithm in the simulation study. The red lines represent the connections (dependencies) that only exist under condition 1, and the green lines represent the connections (dependencies) that only exist under condition 2. The proposed differential dependency network analysis successfully detected 9 of 10 connections that are different between two conditions and all the genes involved in the network topology changes. The connections between PHO2 and SWI4 under condition 1 (red) and between MBP_SWI6 and SWI4 under condition 2 (green) were falsely detected and the connection between PHO2 and TRP4 under condition 1 (red) was falsely missed.

Algorithm Analysis

To investigate the effects of threshold T on the results of the proposed algorithm, we performed the DDN analysis on the simulation data given different thresholds. In this simulation experiment, we know the ground truth, *e.g.* the underlying network topology and how the network topology changes between two conditions. We can demonstrate the effectiveness of this method by showing the precision-recall curves of the DDN analysis (Figure 2.7 and Figure 2.8) [43]. In Figure 2.7, the precision and recall were calculated to assess the detection of the changes of gene-gene connections. In Figure 2.8, the precision and recall were calculated to assess the detection of the "hot spots", *e.g.* genes involved in topological changes. T= 0.25 was used in the simulation experiment in this section. From Figure 2.7 and Figure 2.8 we can see that the DDN analysis can successfully retrieve most of the changes in the network between two conditions, while keeping the precision relatively high.

Another parameter in the DDN algorithm is the p-value cutoff. The local structures with p-values smaller than the user-defined p-value cutoff (0.01 in this experiment) are called significant. A natural question is how many of the detected significant local structures are falsely discovered, in other words, are truly null features. To explore this question, we first need to distinguish two related but distinct concepts: false positive rate and false discovery rate (FDR). The false positive rate is the rate that truly null features are called significant, while the false discovery rate is the rate that significant features are truly null [44]. The p-value is a measure of significance in terms of the false positive rate; the q-value is a measure of the FDR. We adopted the q-value estimation algorithm detailed in [44], to estimate the number of false discoveries in the DDN results. At the given p-value cutoff in this experiment, the estimated number of false discovery is 1.

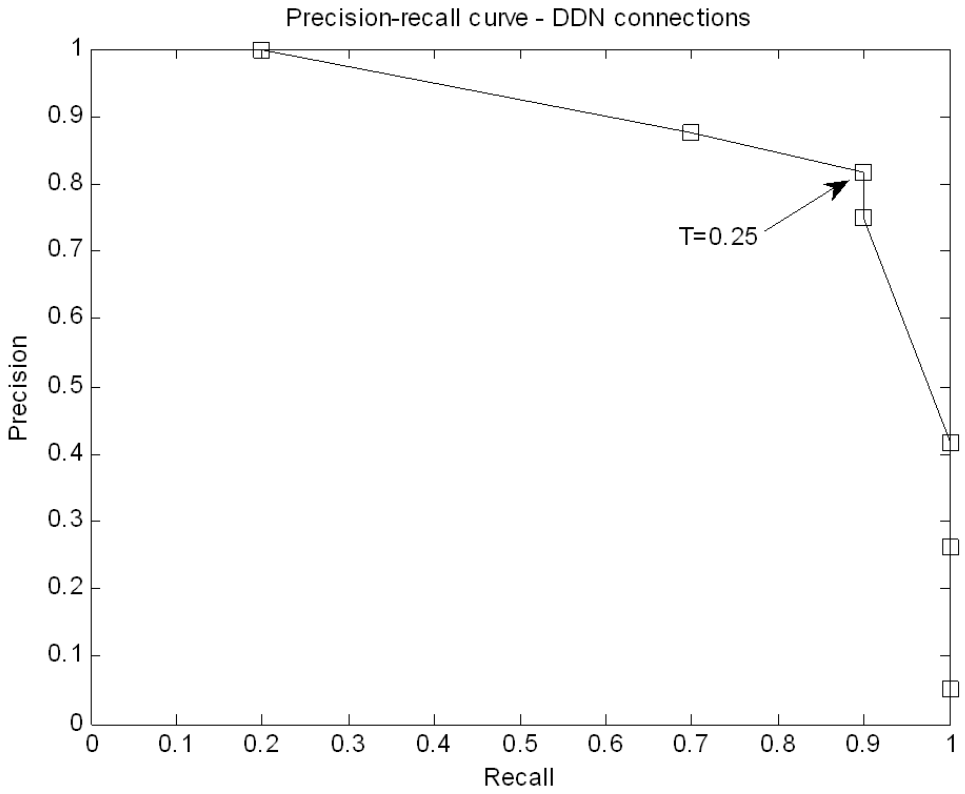


Figure 2.7: Precision-recall curve of DDN analysis. The precision and recall were calculated based on the detected changes in gene-gene connections between two conditions.

2.5.2 Breast Cancer Dataset Analysis

Experiment Background and Data

We further applied our method to the dataset from an estrogen receptor-positive (ER+) breast cancer cell study by Lin et al. [45]. In this dataset, the estrogen-dependent T-47D ER+ breast cancer cell line was treated with 17β-estradiol (E2) and with E2 in combination with the pure anti-estrogen ICI 182,780 (ICI, Faslodex, Fulvestrant). Samples were then harvested on an hourly basis for the first 8 hours (0-8 hours) and bi-hourly for the next 16 hours (10-24 hours) for a total of 16 time points under each condition. Experiments were performed on microarrays generated by spotting the Compugen 19 K human oligo library,

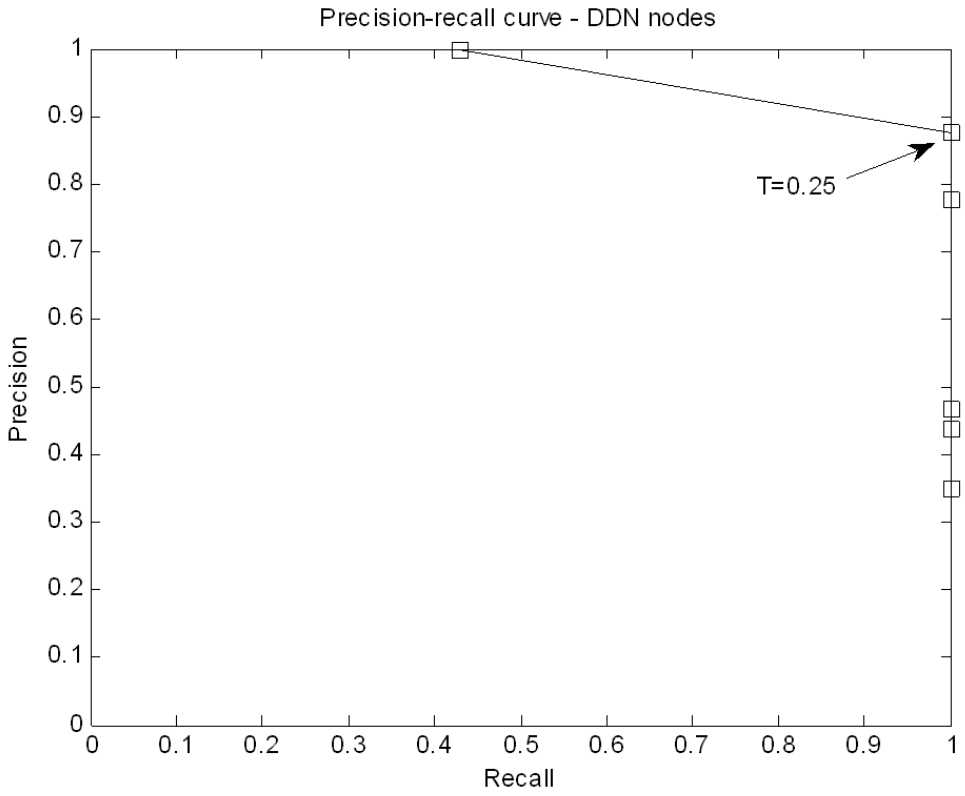


Figure 2.8: Precision-recall curve of DDN analysis. The precision and recall were calculated based on the detected hot-spots under two conditions.

made by Sigma-Genosys, on poly-L-lysine-coated glass slides. In this study, we are interested in the cellular response to the drug ICI, which inhibits E2 signaling through the ER [46].

Application of DDN Analysis

We first selected 55 genes that are reported in the literature to be relevant to breast cancer and responsiveness to ICI, for example [47–49]. We then applied our differential dependency network analysis to the data under two conditions (E2 vs. E2+ICI). The parameters in our algorithm are: threshold T is 0.25, p-value cutoff is 0.01, and K is 2.

The differential dependency network under these two conditions is shown in Figure 2.9.

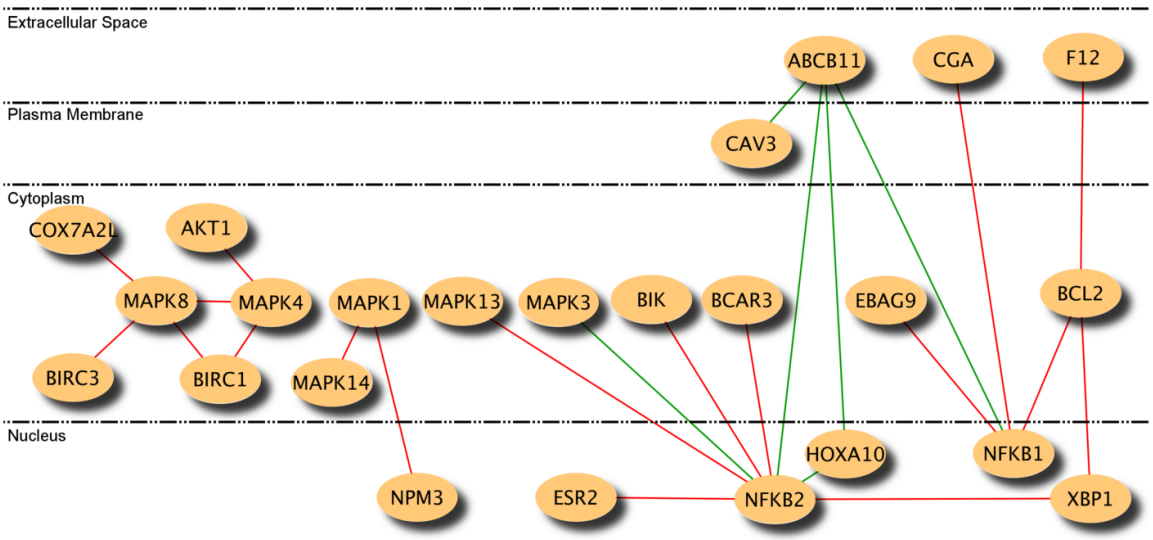


Figure 2.9: Differential dependency network between breast cancer cell line treated with E2 and cell line treated with E2+ICI. The red lines represent the connections that exist only in breast cancer cell line treated with E2, and the green lines represent the connections that exist only in breast cancer cell line treated with E2+ICI.

In Figure 2.9, there are 18 red connections in the differential dependency network, which implies that these connections exist only under E2 condition and disappear after the addition of ICI. Since ICI 182,780 is an estrogen receptor antagonist, which works both by down-regulating and by degrading the estrogen receptor alpha (ER-alpha), it is plausible that these connections disappear because ICI is blocking or inactivating their connections. For example, as a transcription factor, XBP1 can directly regulate gene expression through binding to its response element [50], or it can act as a co-regulator of other transcription factors, most notably ER-alpha, to enhance their transcriptional activity [51, 52]. Because BCL2 contains response elements for both ER-alpha and XBP1 [53, 54], the connection between XBP1 and BCL2 in the differential dependency network may either be direct or involve ERalpha as a latent variable, or intervening gene. In direct support of this predicted edge, we have shown that constitutive over-expression of XBP1 in a different breast cancer cell line (MCF-7) led

to significantly increased mRNA and protein expression of both ERalpha and BCL2 and functionally conferred antiestrogen resistance and estrogen-independence [53, 54].

Novel relationships between these genes identified by our differential dependency network analysis will also serve as useful guidance for future studies. For example, BCAR3 is a well-established effector of cell motility, estrogen independence, and antiestrogen resistance in ER+ breast cancer cell lines [55–58]. Expression of NFKB2 and its activator BCL3 are also associated with estrogen independence in breast cancer cell lines [59], and these nuclear factor κB subunits appear to be selectively activated in clinical breast cancer [60]. However, there is no experimental evidence linking BCAR3 with NFKB2, so the suggestion that these two genes exhibit differential dependence under E2-treated conditions (Figure 2.9) provides a starting point for biological studies of their relationship.

Additional relationships that may be completely new to breast cancer are also identified by this method. For example, MAPK8 (also known as JNK1) has been shown to be activated by BIRC1 (also known as NAIP) during its inhibition of caspase-mediated cell death [61]. In chronic fatigue syndrome, growth factor receptor signaling can activate MAPK4, which via Ras and/or PI3K can subsequently increase AKT1 activity [62]. And finally, in B cells from patients with chronic lymphocytic leukemia NFKB1 (p50) homodimers are able to stimulate transcription from the BCL2 promoter through binding to another member of the BCL family (BCL3) [63].

2.5.3 *In utero* Excess E2 Exposed Adult Mammary Glands Analysis

Experiment Background and Data

The level of estrogenicity of the *in utero* environment significantly affects the developmental programming of the mammary gland and its susceptibility to tumorigenesis later in life. An elevated in utero estrogenic environment may increase later susceptibility to develop breast cancer. The key transcription factors and signaling that mediates the effects of *in utero* estrogenic environment on later estrogen sensitivity and breast cancer risk are unknown. Transcriptome analysis of mRNA from normal adult rat mammary glands exposed to excess E2 *in utero* and vehicle controls may help to shed light on the important genes and pathways. In this gene expression dataset, there are five samples of normal adult rat mammary glands exposed to excess E2 *in utero* and five samples of vehicle controls.

Application of DDN Analysis

We applied our DDN analysis to this dataset. The parameters in our algorithm are: threshold T is 0.4, p-value cutoff is 0.05, and K is 1. The differential dependency network of control group vs. excess E2 *in utero* group is shown in Figure 2.10. Since the exposure was *in utero*, but the differential transcriptome analysis done in adulthood, the altered expression of these genes over time could be, at least in part, a consequence of transcriptional programming regulated by promoter methylation status. Many of these genes are known to be regulated by promoter methylation, *e.g.*, ER [64,65], BCL2 [65,66], LEP (leptin) [67], and EGR1 [68]. AKT1 can regulate methylation patterns in some promoters, which may explain the nature of the AKT1-EGR1 edge present only in the control mammary glands, providing a testable hypothesis [69].

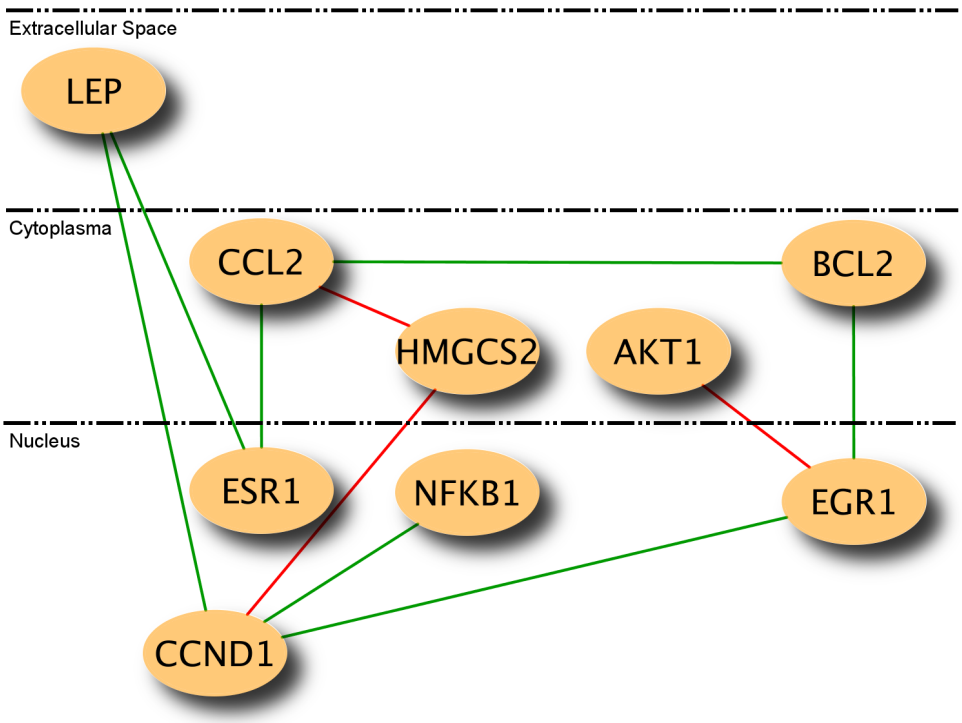


Figure 2.10: Differential dependency network between control group and excess E2 *in utero* group. The red lines represent the connections that exist only in control group, and the green lines represent the connections that exist only in excess E2 *in utero* group.

2.5.4 A Case Study on Juvenile Dermatomyositis

Experiment Background and Data

Identifying the transcriptional network differences between normal muscle and muscular dystrophy is of great interest. We performed DDN analysis between normal human muscle (NHM) samples and juvenile dermatomyositis (JDM) samples using an in-house data set. We selected 49 genes that have been previously reported in literature to be related to myogenesis [70–72].

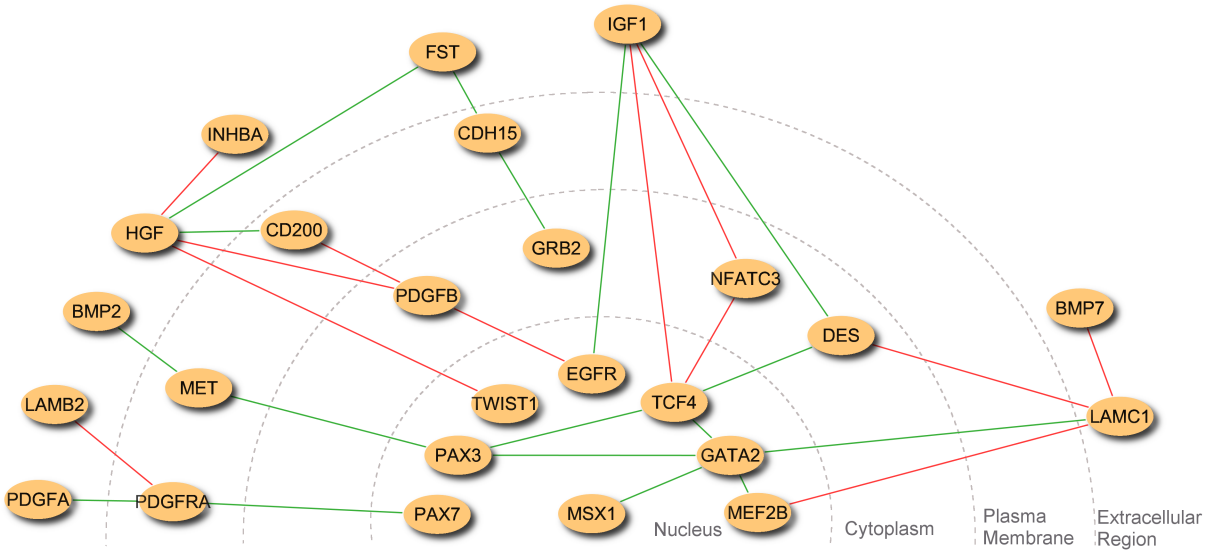


Figure 2.11: Differential dependency network between NHM and JDM, green edges are connections that only exist in JDM, red edges are connections that only exist in NHM. The layout of the network is arranged according to their cellular components in the gene ontology. Parameters: $K = 1, T = 0.5, p = 0.05$.

Application of DDN Analysis

DDN between NHM and JDM detects the transcriptional network changes in normal muscle cells and juvenile dermatomyositis samples, as shown in Figure 2.11.

Compared with NHM, significant increase in gene activities is detected in JDM. In Figure 2.11, we can see that 25 genes are extracted out of 49 genes and most connections are green. Thus, when compared with normal muscle, these gene interactions appear in the dermatomyositis samples. This observation may be due to the reasons that JDM is an auto-immune disease that is expected to exhibit higher level of immune response, and in damaged muscle cells myogenesis is subsequently initiated in an attempt to repair the damage muscle cells and restore normal muscle functions. Therefore, compared with normal muscle cells, higher molecular activities and more gene interactions/cooperation in immune response and

myogenesis are observed.

From annotations of these 25 genes, we see that the identified genes are all closely related to muscle generation, growth and transcription. Hence, in damaged muscle cells, muscle cell renewal process may be triggered and muscle generation and growth activities could be higher than normal, explaining why more muscle growth related gene connections are detected in JDM samples.

2.6 Conclusions and Discussions

In this chapter, we discuss a systematic approach to detect the statistically significant changes in transcriptional networks between two different experimental conditions. We tested our algorithm on simulation data and two real datasets. From the simulation study, we see that the proposed algorithm can efficiently and accurately capture the topological changes. This approach utilizes the network structure information and provides an alternative way for biomarker identification.

The high level of correlation among genes is a common feature of microarray data. Therefore, we propose a local dependency model that allows multiple predictor sets for each node. Accordingly, a local structure learning algorithm is also represented. Lasso is used to select features for the predictor sets [27], an approach that has been successfully applied to variable selection and graph structure learning [33]. In the linear Gaussian case, under certain conditions it is proved that the probability of estimating the correct neighborhood converges exponentially to 1. Consequently, it is possible to obtain a consistent estimation of the full edge set [33]. In microarray data, the so-called irrepresentable condition [35] or the neighborhood stability assumption [33] can easily be violated in the presence of highly correlated genes [1]. Some modified algorithms have been proposed to deal with the highly

correlated cases, for example, elastic net [36] and network-constrained regularization [24], both of which tend to group highly correlated predictors in the regression process. However, neither of these two approaches are suitable for our problem because the grouping of highly correlated variables can be different under two conditions and this makes the later statistical testing problematic. The local structure learning algorithm proposed here attempts to alleviate the effects of highly correlated data and to preserve local structure information for further statistical testing.

Some issues are worth further exploration. Currently, only linear relationships are considered. As shown in [17] (Equations (7) and (8)), the relationship between transcription factor activities and gene expression levels can be approximated by a log-linear model. Further, from a biochemistry perspective, transcription factors regulate promoter activity through binding to the promoter region, which is modeled by the Hill equation (Supplementary Equation (E1) in [17]). The mRNA level in the cell is a balance between the rate of mRNA synthesis (promoter activity) and the rate of mRNA degradation. It was argued that on time scales > 10 min, the mRNA levels reach a quasi-steady state, and the relationship between transcription factor activities and gene expression can be approximated by a log-linear model (Supplementary Equations (E2), (E3), and (E5) in [17]). Taking logarithm of the raw intensity data is a common step in the pre-processing of the gene expression microarray data and we applied the log-transform to the raw intensity data in previous experiments in this chapter. Therefore, such understanding serves as a motivation of applying linear regression models for characterizing the relationships in the transcriptional networks. Moreover, when applying to real biological experiment data, we often have limited samples (usually tens of samples under each condition). Therefore it is quite challenging to learn nonlinear models from so few samples without overfitting the data.

The advances in system identification methods are very encouraging, but it is difficult to apply these new methods directly to the experiment settings we discussed in this chapter,

because most of these methods deal with dynamic systems, while the biological data we try to model here are not time-course data, often independent samples from two different groups (*e.g.* drug treated group *v.s.* control group). On the other hand, as the cost of the gene expression microarrays decreases and more and more time course biological data become available, it will be of great scientific significance to more accurately model the nonlinear interactions in the biological networks and apply state-of-the-art system identification methods to model gene-gene interactions along the time course. How such nonlinear, dynamic systems can be modeled efficiently and correctly remains an open problem.

Another limitation of this approach is that the current dependency model cannot handle self-loops in the networks. Since self-feedback loops are not rare in transcriptional networks, it will be very interesting to detect and model this type of connections using computational approaches.

In summary, DDN analysis presents a new approach to extract knowledge of a biological network by emphasizing the dynamic nature of cellular networks and utilizing a network's structural information. It also provides an alternative and promising approach to identify possible biomarkers and drug targets.

Chapter 3

Learning Structural Changes of Gaussian Graphical Models between Two Conditions

3.1 Introduction

Controlled experiments are very common yet effective tools in scientific research. For example, in the studies of disease or drug effectiveness using case-control experiments, the changes of the conditional dependence between measurement variables are often reflected in the structural changes in the corresponding graphical models that can reveal crucial information about how the systems responds to external stimuli or adapts to changed conditions. The ability to detect and extract the structural changes from data can facilitate the generation of new insights and new hypotheses for further studies.

Consider the example of gene regulatory networks in systems biology. Gene regulatory networks are context-specific and dynamic in nature, that is, under different conditions, different

regulatory components and mechanisms are activated and accordingly the topology of the underlying gene regulatory network changes [26]. For example, in response to diverse conditions in the yeast, transcription factors alter their interactions and rewire the signaling networks [3]. Such changes in network structures provide great insights into the underlying biology of how the organism responses to outside stimuli. In disease studies, it is important to examine the topological changes in transcriptional networks between disease and normal conditions where a deviation from normal regulatory network topology may reveal the mechanism of pathogenesis, and the genes that undergo the most network topological changes may serve as potential biomarkers or drug targets.

Similar phenomena also appear in other areas. For instance, in web search or collaborative filtering, useful information can be acquired by observing how certain events (*e.g.*, launch of an advertisement campaign) trigger changes in dependence patterns of search keywords or preference for products reflected in the associated structural changes.

The conditional dependence of a set of random variables are often mathematically characterized by graphical models, such as Bayesian networks and Markov networks, and various methods have been proposed to learn graphical model structures from data [73, 74]. Although learning the graphical models under two conditions can be separately achieved and the structural and parametric differences can be subsequently compared, such technically convenient framework would completely collapse when the structural and parametric inconsistencies due to limited data samples and noise effects are significant and hinder an accurate detection of true and meaningful structural and parametric changes.

In this chapter, we report an effective learning strategy to extract structural changes of Gaussian graphical models in controlled experiments using convex optimization. We discuss the properties of the problem formulation and introduce an efficient block coordinate descent algorithm. We demonstrate the principle of the approach on a numerical simulation experiment, and we then apply the algorithm to the modeling of gene regulatory networks

under different conditions and obtain promising yet biologically plausible results.

3.2 A Revisit on Gaussian Graphical Model Structural Learning

The structures of graphical models in many cases are unknown and need to be learned from data. In this chapter, we focus on Gaussian graphical models, in which the nodes (variables) are Gaussian, and their dependence relationships are linear. Assume we have a set of p random variables of interest, $\mathbb{X} = \{X_1, X_2, ..., X_p\}$, and N observations, $\boldsymbol{x}_j = [x_{1j}, x_{2j}, ..., x_{Nj}]^T$, $j = 1, 2, ..., p$. Let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_p]$ be the data matrix.

Learning the structures of graphical models efficiently is often very challenging. Recently, ℓ_1-regularization has drawn great interest in statistics and machine learning community [27, 32, 35, 36]. Penalty on the ℓ_1-norm of the regression coefficients has two very useful properties: sparsity and convexity. The ℓ_1-norm penalty tends to make some coefficients exactly zeros, leading to a parsimonious solution, which naturally performs variable selection or sparse linear model estimation. Further, the convex nature of ℓ_1-norm penalty makes the problem computationally tractable, which can be solved readily by many existing convex optimization methods.

Several ℓ_1-regularization approaches have been successfully applied to graphical model structure learning [34, 38, 75], especially Gaussian graphical models [33, 76, 77]. Meinshausen and Buhlmann proposed to use lasso to identify the neighborhood of the nodes in Gaussian graphs [33]. The neighborhood selection of a node X_j, $j = 1, 2, ..., p$, is solved by applying lasso to learn the prediction model of variable X_j, given all remaining variables $\mathbb{X}_{-j}$.

The lasso estimate $\hat{\boldsymbol{\beta}}$ is given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}:\beta_j=0} \|\boldsymbol{x}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{3.1}$$

where $\lambda > 0$ is the Lagrange multiplier.

If the k^{th} element of $\hat{\boldsymbol{\beta}}$ is non-zero, then there is an edge between node j and node k. This procedure is performed on each of the p random variables, and thereby the structure of the Gaussian graphical model is learned. Meinshausen and Buhlmann also showed that under certain conditions, the proposed neighborhood selection scheme is consistent for sparse high-dimensional graphs [33].

3.3 Problem Formulation

Now we consider the problem of learning structural changes of a graphical model between two conditions. This is equivalent to investigating how the conditional dependence and independence of a set of random variables change under these two conditions. Similarly, we have a set of p random variables of interest, $\mathbb{X} = \{X_1, X_2, ..., X_p\}$, and we observed N_1 samples under condition 1 and N_2 samples under condition 2. Without loss of generality, we assume $N_1 = N_2 = N$, which means we have balanced observations from two conditions. Under the first condition, for variable X_j, we have observations $\boldsymbol{x}_j^{(1)} = [x_{1j}^{(1)}, x_{2j}^{(1)}, ..., x_{Nj}^{(1)}]^T$, $j = 1, 2, ..., p$, while under the second condition, we have $\boldsymbol{x}_j^{(2)} = [x_{1j}^{(2)}, x_{2j}^{(2)}, ..., x_{Nj}^{(2)}]^T$, $j = 1, 2, ..., p$. Further, let $\mathbf{X}^{(1)} = [\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(1)}, ..., \boldsymbol{x}_p^{(1)}]$ be the data matrix under condition 1 and $\mathbf{X}^{(2)} = [\boldsymbol{x}_1^{(2)}, \boldsymbol{x}_2^{(2)}, ..., \boldsymbol{x}_p^{(2)}]$ be the data matrix under condition 2.

Further, denote

$$\boldsymbol{y}_j = \begin{bmatrix} \boldsymbol{x}_j^{(1)} \\ \boldsymbol{x}_j^{(2)} \end{bmatrix}, \tag{3.2}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}^{(2)} \end{bmatrix}, \tag{3.3}$$

and

$$\begin{aligned} \boldsymbol{\beta} &= \begin{bmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \end{bmatrix} \\ &= [\beta_1^{(1)}, \beta_2^{(1)}, ..., \beta_p^{(1)}, \beta_1^{(2)}, \beta_2^{(2)}, ..., \beta_p^{(2)}]^T. \end{aligned} \tag{3.4}$$

By location and scale transformations, we can always assume that the variables have mean 0 and unit length,

$$\begin{aligned} \sum_{i=1}^{N} x_{ij}^{(1)} = 0, \qquad \sum_{i=1}^{N} (x_{ij}^{(1)})^2 = 1, \\ \sum_{i=1}^{N} x_{ij}^{(2)} = 0, \qquad \sum_{i=1}^{N} (x_{ij}^{(2)})^2 = 1, \end{aligned} \tag{3.5}$$

where $j = 1, 2, ..., p$.

We formulate the problem of learning structural changes between two conditions as a convex optimization problem. We solve the following optimization problem for each node (variable) X_j, $j = 1, 2, ..., p$.

$$f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1 \tag{3.6}$$

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) \\ &= \arg\min_{\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}} \frac{1}{2}\|\boldsymbol{y}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \\ &\qquad + \lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1 \\ &\text{s.t. } \beta_j^{(1)} = 0, \beta_j^{(2)} = 0 \end{aligned} \tag{3.7}$$

In (3.7), we learn the structures of the graphical model under two conditions jointly. The ℓ_2-loss function and the first ℓ_1-regularization term, $\lambda_1\|\boldsymbol{\beta}\|_1$, lead to the identification of

sparse graph structure. The second ℓ_1-regularization term, $\lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1$, encourages sparse changes in the model structure and parameters between two conditions, and thereby suppresses the structural and parametric inconsistencies due to noise in the data and limited samples. The objective function (3.6) is non-differentiable, continuous, and convex.

The optimization problem (3.7) may appear similar to the fused lasso [78], which was applied to protein mass spectroscopy and DNA copy number detection [79]. The fused lasso encourages the flatness of the coefficient profile β_j as a function of the index j. Kolar *et al.* investigated learning of varying-coefficient varying-structure models from time-course data, in which $\boldsymbol{\beta}_t$ is a function of time t, and proposed a two-stage procedure that first identifies jump points and then identifies relevant covariates [80]. The total variation norm (TV-norm) of $\boldsymbol{\beta}_t$ is used to encourage sparse changes along the time course.

Besides targeted at different applications, the objective function (3.6) has two important technical differences from the above two approaches. First, the penalty term $\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1$ has *block-wise separability*, which means the non-differentiable objective function $f(\boldsymbol{\beta})$ can be written in the form

$$f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + \sum_{m=1}^{M} h_m(\boldsymbol{b}_m), \tag{3.8}$$

where $g(\boldsymbol{\beta})$ is convex and differentiable, $\boldsymbol{b}_m$ is some subset of $\boldsymbol{\beta}$, $h_m(\boldsymbol{b}_m)$ is convex and non-differentiable, and $\boldsymbol{b}_{m_1}$ and $\boldsymbol{b}_{m_2}$, $m_1 \neq m_2$, do not have overlapping members [81].

We rewrite the objective function (3.6) as

$$\begin{aligned} & f(\boldsymbol{\beta}) \\ =& \frac{1}{2}\|\boldsymbol{y}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{k=1}^{p} (|\beta_k^{(1)}| + |\beta_k^{(2)}|) \\ & + \lambda_2 \sum_{k=1}^{p} (|\beta_k^{(1)} - \beta_k^{(2)}|) \end{aligned}$$

Therefore, the non-differentiable part of $f(\boldsymbol{\beta})$ can be written as the sum of p terms with

non-overlapping members, $(\beta_k^{(1)}, \beta_k^{(2)})$, $k = 1, 2, ..., p$. Each $(\beta_k^{(1)}, \beta_k^{(2)})$, $k = 1, 2, ..., p$, is a coordinate block.

We will show in Section 4 that this property is essential for the convergence of the block coordinate descent algorithm to solve problem (3.7). On the other hand, Friedman *et al.* has shown that coordinate-wise descent does not work in fused lasso, since the non-differentiable penalty function is not separable [79].

Additionally, the k^{th} column of matrix $\mathbf{X}$, $\boldsymbol{x}_k$, and the $(k+p)^{\text{th}}$ column of $\mathbf{X}$, $\boldsymbol{x}_{k+p}$, are orthogonal, *i.e.*, $\boldsymbol{x}_k^T \cdot \boldsymbol{x}_{k+p} = 0$, $k = 1, 2, ..., p$. This simplifies the derivation of closed-form solutions to the sub-problems in each iterations of the block coordinate descent.

We summarize our discussions above as three properties of problem (3.7).

Property 1 (Convexity)**.** *The objective function* (3.6) *is continuous and convex.*

Property 2 (Block-wise Separability)**.** *The non-differential part of the objective function* (3.6)*,* $\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1$*, is block-wise separable.*

Property 3 (Orthogonality in the Coordinate Block)**.** $\boldsymbol{x}_k^T \cdot \boldsymbol{x}_{k+p} = 0$*,* $k = 1, 2, ..., p$*.*

To represent the result as a graph, the non-zero elements of $\boldsymbol{\beta}^{(1)}$ indicate the neighbors and edges of node X_j under the first condition and the non-zero elements of $\boldsymbol{\beta}^{(2)}$ indicate the neighbors and edges of node X_j under the second condition.The non-zero elements of $\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}$ provide the changed edges (both structural and parametric difference) of node X_j between two conditions. We repeat this procedure to each node X_j, $j = 1, 2, ..., p$, and then we obtain the graph under two conditions. In gene regulatory network modeling, we are particularly interested in where and how the gene regulatory network exhibits different network topology between two conditions. To highlight such changes, we extract the sub-network in which nodes have different connections between two conditions.

3.4 Algorithm

In the realm of computational biology and data mining, vast amount of data and high dimensionality require efficient algorithms. Although the optimization problems with ℓ_1-regularization can be solved readily by existing convex optimization techniques, a lot of efforts have been made to solve the problems efficiently by exploiting the special structures of the problems. A well-known approach is the least angle regression (LARS), which can be modified to solve lasso problems [32]. Recently, coordinate-wise descent algorithms have been studied in lasso related problems, such as lasso, garotte and elastic net [79]. Friedman *et al.* showed with experiments that a coordinate descent procedure for lasso, graphical lasso, is 30-4000 times faster than competing methods, making it a computationally attractive method [77].

3.4.1 Block Coordinate Descent Algorithm

In this chapter, we adopt this idea and propose a block coordinate descent algorithm to solve the optimization problem (3.7) for each node X_j, $j = 1, 2, ..., p$. The essence of the block coordinate descent algorithm is "one-block-at-a-time". At iteration $r + 1$, only one coordinate block, $(\beta_k^{(1)}, \beta_k^{(2)})$, is updated, with the remaining $(\beta_l^{(1)}, \beta_l^{(2)})$, $l \neq k$, fixed at their values at iteration r. Given

$$\boldsymbol{\beta}^r = [\beta_1^{(1),r}, \beta_2^{(1),r}, ..., \beta_p^{(1),r}, \beta_1^{(2),r}, \beta_2^{(2),r}, ..., \beta_p^{(2),r}]^T, \tag{3.9}$$

at iteration $r+1$, the estimation is updated according to the following sub-problem

$$\begin{aligned}
&\boldsymbol{\beta}^{r+1} = \arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) \\
&\text{s.t. } \beta_l^{(1)} = \beta_l^{(1),r}, \\
&\qquad \beta_l^{(2)} = \beta_l^{(2),r}, \\
&\qquad \text{for } l = 1, 2, ..., p, l \neq k.
\end{aligned} \tag{3.10}$$

We use a *cyclic rule* to update parameter estimation iteratively, *i.e.*, update parameter pair $(\beta_k^{(1)}, \beta_k^{(2)})$ at iteration $r+1$, and $k = ((r+1) \mod p) + 1$.

3.4.2 Closed-form Solution to the Sub-problem

Thus the problem is reduced to solving the sub-problem (3.10). Since $\beta_l^{(1)}$ and $\beta_l^{(2)}$, $l = 1, 2, ..., p, l \neq k$, are fixed during iteration $r+1$, we rewrite the objective function of (3.10) as

$$\begin{aligned}
&\tilde{f}(\boldsymbol{\beta}) \\
=&\frac{1}{2}\|\boldsymbol{y}_j - \sum_{l\neq j,k} \boldsymbol{x}_l \beta_l^{(1),r} - \sum_{l\neq j,k} \boldsymbol{x}_{(p+l)} \beta_l^{(2),r} \\
&\qquad\qquad - \boldsymbol{x}_k \beta_k^{(1)} - \boldsymbol{x}_{p+k} \beta_k^{(2)}\|_2^2 \\
&+ \lambda_1 \sum_{l\neq j,k} (|\beta_l^{(1),r}| + |\beta_l^{(2),r}|) + \lambda_2 \sum_{l\neq j,k} (|\beta_l^{(1),r} - \beta_l^{(2),r}|) \\
&+ \lambda_1 (|\beta_k^{(1)}| + |\beta_k^{(2)}|) + \lambda_2 (|\beta_k^{(1)} - \beta_k^{(2)}|)
\end{aligned} \tag{3.11}$$

Let

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{y}_j - \sum_{l\neq j,k} \boldsymbol{x}_l \beta_l^{(1),r} - \sum_{l\neq j,k} \boldsymbol{x}_{(p+l)} \beta_l^{(2),r}. \tag{3.12}$$

Therefore, updating $(\beta_k^{(1)}, \beta_k^{(2)})$ is equivalent to

$$\begin{aligned}
&(\beta_k^{(1),r+1}, \beta_k^{(2),r+1}) \\
=&\arg\min_{\beta_k^{(1)},\beta_k^{(2)}} \tilde{f}(\boldsymbol{\beta}) \\
=&\arg\min_{\beta_k^{(1)},\beta_k^{(2)}} \frac{1}{2}\|\tilde{\boldsymbol{y}}_j - \boldsymbol{x}_k\beta_k^{(1)} - \boldsymbol{x}_{p+k}\beta_k^{(2)}\|_2^2 \\
&+ \lambda_1(|\beta_k^{(1)}| + |\beta_k^{(2)}|) + \lambda_2(|\beta_k^{(1)} - \beta_k^{(2)}|)
\end{aligned} \tag{3.13}$$

Denote

$$\rho_1 = \tilde{\boldsymbol{y}}_j^{\,T} \cdot \boldsymbol{x}_k, \tag{3.14}$$

$$\rho_2 = \tilde{\boldsymbol{y}}_j^{\,T} \cdot \boldsymbol{x}_{p+k}. \tag{3.15}$$

First, we examine a simple case, the solution, $(\beta_k^{(1)}, \beta_k^{(2)})$, satisfies

$$\begin{cases}
\beta_k^{(1)} > 0, \\
\beta_k^{(2)} > 0, \\
\beta_k^{(1)} < \beta_k^{(2)}.
\end{cases} \tag{3.16}$$

Take derivative of objective function (3.11), and we have

$$\frac{\partial \tilde{f}}{\partial \beta_k^{(1)}} = \beta_k^{(1)} - \rho_1 + \lambda_1 \text{sgn}(\beta_k^{(1)}) + \lambda_2 \text{sgn}(\beta_k^{(1)} - \beta_k^{(2)}), \tag{3.17}$$

$$\frac{\partial \tilde{f}}{\partial \beta_k^{(2)}} = \beta_k^{(2)} - \rho_2 + \lambda_1 \text{sgn}(\beta_k^{(2)}) - \lambda_2 \text{sgn}(\beta_k^{(1)} - \beta_k^{(2)}), \tag{3.18}$$

where $\text{sgn}(\cdot)$ is the sign function.

When $\rho_1 > \lambda_1 - \lambda_2$ and $\rho_2 > \rho_1 + 2\lambda_2$, we have

$$\begin{cases}
\beta_k^{(1)} = \rho_1 - \lambda_1 + \lambda_2, \\
\beta_k^{(2)} = \rho_2 - \lambda_1 - \lambda_2.
\end{cases} \tag{3.19}$$

Similarly, we derive all closed-form solutions to problem (3.10), depending on the values of ρ_1, ρ_2 with respect to λ_1, λ_2. The plane (ρ_1, ρ_2) is divided into 13 regions, as shown in Figure 3.1.

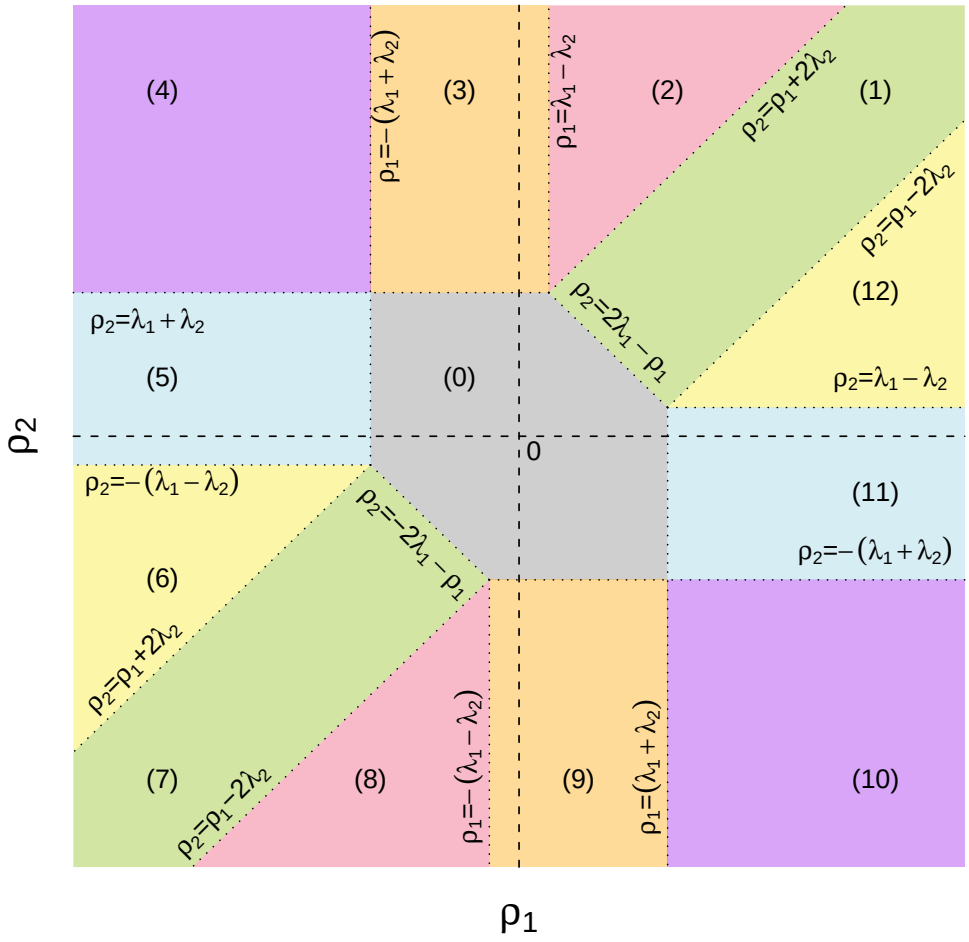


Figure 3.1: Solution regions of the sub-problem.

Depending on the location of (ρ_1, ρ_2) in the plane, the solutions to problem (3.10) are as follows.

If (ρ_1, ρ_2) is in region (0), then

$$\beta_k^{(1)} = \beta_k^{(2)} = 0. \tag{3.20}$$

If (ρ_1, ρ_2) is in region (1), then

$$\beta_k^{(1)} = \beta_k^{(2)} = \frac{1}{2}(\rho_1 + \rho_2) - \lambda_1 \tag{3.21}$$

If (ρ_1, ρ_2) is in region (2), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 - \lambda_1 + \lambda_2, \\ \beta_k^{(2)} = \rho_2 - \lambda_1 - \lambda_2. \end{cases} \tag{3.22}$$

If (ρ_1, ρ_2) is in region (3), then

$$\begin{cases} \beta_k^{(1)} = 0, \\ \beta_k^{(2)} = \rho_2 - \lambda_1 - \lambda_2. \end{cases} \tag{3.23}$$

If (ρ_1, ρ_2) is in region (4), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 + \lambda_1 + \lambda_2, \\ \beta_k^{(2)} = \rho_2 - \lambda_1 - \lambda_2. \end{cases} \tag{3.24}$$

If (ρ_1, ρ_2) is in region (5), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 + \lambda_1 + \lambda_2, \\ \beta_k^{(2)} = 0. \end{cases} \tag{3.25}$$

If (ρ_1, ρ_2) is in region (6), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 + \lambda_1 + \lambda_2, \\ \beta_k^{(2)} = \rho_2 + \lambda_1 - \lambda_2. \end{cases} \tag{3.26}$$

If (ρ_1, ρ_2) in region (7), then

$$\beta_k^{(1)} = \beta_k^{(2)} = \frac{1}{2}(\rho_1 + \rho_2) + \lambda_1. \tag{3.27}$$

If (ρ_1, ρ_2) is in region (8), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 + \lambda_1 - \lambda_2, \\ \beta_k^{(2)} = \rho_2 + \lambda_1 + \lambda_2. \end{cases} \tag{3.28}$$

If (ρ_1, ρ_2) is in region (9), then

$$\begin{cases} \beta_k^{(1)} = 0, \\ \beta_k^{(2)} = \rho_2 + \lambda_1 + \lambda_2. \end{cases} \tag{3.29}$$

If (ρ_1, ρ_2) is in region (10), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 - \lambda_1 - \lambda_2 \\ \beta_k^{(2)} = \rho_2 + \lambda_1 + \lambda_2. \end{cases} \tag{3.30}$$

If (ρ_1, ρ_2) is in region (11), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 - \lambda_1 - \lambda_2 \\ \beta_k^{(2)} = 0. \end{cases} \tag{3.31}$$

If (ρ_1, ρ_2) is in region (12), then

$$\begin{cases} \beta_k^{(1)} = \rho_1 - \lambda_1 - \lambda_2 \\ \beta_k^{(2)} = \rho_2 - \lambda_1 + \lambda_2. \end{cases} \tag{3.32}$$

3.4.3 Convergence Analysis

Finally, we summarize the optimization procedure to solve problem (3.7) in Algorithm 1.

The convergence of Algorithm 1 is stated in the following theorem.

Theorem 1. *The solution sequence generated by Algorithm 1 is bounded and every cluster point is a solution of problem* (3.7).

Algorithm 1 Block coordinate descent algorithm to solve problem (3.7).

Initialization: $\boldsymbol{\beta}^0 = [0, 0, ..., 0]$, $r = 0$

while $\boldsymbol{\beta}^r$ is not converged **do**

$k \leftarrow (r \mod p) + 1$

if $k \neq j$ **then**

Let $\beta_l^{(1),r+1} = \beta_l^{(1),r}$, $\beta_l^{(2),r+1} = \beta_l^{(2),r}$, $l \neq k$

Calculate $\tilde{\boldsymbol{y}}_j$ according to (3.12).

Calculate ρ_1 and ρ_2 using (3.14) and (3.15).

Update $\beta_k^{(1),r+1}$ and $\beta_k^{(2),r+1}$, according to (20)-(32).

end if

$r \leftarrow r + 1$

end while

Proof. We have shown in Property 1 and Property 2 that the objective function (3.6) is continuous and convex, and the non-differential part of the objective function is block-wise separable. By applying Theorem 4.1 proposed by Tseng *et al.* [81], we have that the solution sequence generated by Algorithm 1 is bounded and every cluster point is a solution of problem (3.7). □

3.4.4 Determining Parameters λ_1 and λ_2

As we discussed previously, the first ℓ_1-regularization term, $\lambda_1\|\boldsymbol{\beta}\|_1$, leads to the identification of sparse graph structures, and the second ℓ_1-regularization term, $\lambda_2\|\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}^{(2)}\|_1$, suppresses the inconsistencies of the network structures and parameters between two conditions, due to the noise in the data and limited samples.

First, we consider the case $\lambda_2 = 0$. In this case, the problem (3.7) is equivalent to applying lasso to the data under two conditions separately. The λ_1 controls the sparsity of the learned

graph, and Algorithm 1 is reduced to a coordinate descent algorithm, in which each sub-problem is lasso with two orthogonal predictors. The value of λ_1 can be determined easily via cross-validation. In our experiments, we used 10-fold cross-validation, following steps specified in [82].

Then we consider the second parameter λ_2. The parameter λ_2 controls the sparsity of structural and parametric changes between two conditions. From regions (1) and (7) of Figure 3.1, we can see that if $|\rho_1 - \rho_2| \leq 2\lambda_2$, then $\beta_k^{(1)}$ and $\beta_k^{(2)}$ will be set equal (Equations (3.21) and (3.27)) as the solution of the sub-problem (3.10). Therefore, the remaining question is when $|\rho_1 - \rho_2|$ is large enough to be considered significant, at a given significance level α. We present here a heuristic approach to determine λ_2.

Applying Fisher transform to both ρ_1 and ρ_2, we have

$$z_1 = \frac{1}{2}\ln\frac{1+\rho_1}{1-\rho_1}, \quad z_2 = \frac{1}{2}\ln\frac{1+\rho_2}{1-\rho_2}. \tag{3.33}$$

Since data matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ are drawn from Gaussian distributions, we know z_1 and z_2 are approximately normally distributed with standard deviation $\frac{1}{\sqrt{N-3}}$ and means $\frac{1}{2}\ln\frac{1+\rho_1}{1-\rho_1}$ and $\frac{1}{2}\ln\frac{1+\rho_2}{1-\rho_2}$, respectively.

Further, under the null hypothesis that $\rho_1 = \rho_2$ (and therefore $z_1 = z_2$), define

$$z = z_1 - z_2, \tag{3.34}$$

which approximately follows normal distribution with zero mean and standard deviation $\frac{1}{\sqrt{(N-3)/2}}$.

At a given significance level α (*e.g.*, $\alpha = 0.01$ is used in the experiments), if $|z| = |z_1 - z_2| \geq s$, it will be considered significant, where $s = \Phi^{-1}(1-\alpha/2)/\sqrt{(N-3)/2}$. Through simple derivation, we have

$$\begin{aligned} &|z| = |z_1 - z_2| \geq s \\ \Rightarrow &|\rho_1 - \rho_2| \geq \frac{e^{2s}-1}{e^{2s}+1}(1-\rho_1\rho_2) = 2\lambda_2 \end{aligned} \tag{3.35}$$

To further simplify (3.35) with some approximation, we estimate overall $\rho_1\rho_2$ by $\overline{\rho_1\rho_2} = 2\sum_{j<l} \boldsymbol{y}_j^T \boldsymbol{x}_l \cdot \boldsymbol{y}_j^T \boldsymbol{x}_{p+l}/p(p-1)$. Substituting $\overline{\rho_1\rho_2}$ in (3.35), we have

$$\lambda_2 = \frac{e^{2s}-1}{2e^{2s}+2}(1-\overline{\rho_1\rho_2}). \tag{3.36}$$

3.5 Discussions on Biological Prior Knowledge Incorporation

3.5.1 Motivation

In the past decade, a lot of efforts have also been made to manually curate molecular interactions in cells, such as protein-protein interactions and biological pathways [83], which can now be conveniently retrieved from relevant biological databases. These biological databases summarize existing knowledge and experimental evidence from multiple sources under diverse conditions, and attempt to delineate a more detailed picture of the interactome in the cells. Such biological prior knowledge provides rich domain knowledge to the biological network inference problem [84]. Compared with computational inference purely based on data, proper incorporation of biological prior knowledge into network learning algorithms can effectively leverage domain knowledge and make the inference more biologically meaningful. However, as biological prior knowledge is usually aggregated from multiple sources and under diverse experimental settings, direct incorporation of prior knowledge in specific problems is prone to errors or may even lead to biased results.

3.5.2 Problem Statement

We represent the condition-specific biological networks as graphs. Here we focus on condition-specific biological network structure and their corresponding structural changes under two

conditions, which is a common experiment setting in biomedical research, such as controlled experiments and comparisons between two populations. Suppose there are p nodes (genes) in the network of interest, and we denote the vertex set as V. Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be the two undirected graphs under the two conditions. G_1 and G_2 have the same vertex set V, and condition-specific edge sets E_1 and E_2. E_1 and E_2 are expected to have considerable overlap, with only a small amount of edges being different. Such edge changes are of particular interest, since such rewiring may reveal pivotal information on how the organisms response to different conditions.

Biological prior knowledge is collected from biological databases such as KEGG pathway database and Human Protein Reference Database. We denote the biological prior knowledge as a knowledge graph $G_{\mathbf{W}} = (V, E_{\mathbf{W}})$, where the vertex set V is the same set of nodes (genes) and edge set $E_{\mathbf{W}}$ over V is retrieved from biological databases as supported by the existing knowledge or other experimental evidence.

We use a symmetric matrix $\mathbf{W} \in \mathbb{R}^{p \times p}$ to represent the prior knowledge, which is the adjacency matrix of $G_{\mathbf{W}}$. The elements of $\mathbf{W}$ are either 1 or 0, with $W_{ij} = 1$ indicating the existence of an edge between the i^{th} gene and j^{th} gene (or their gene products) in the databases, where $i, j = 1, 2, \cdots, p, i \neq j$.

The main task here is to infer from data and prior knowledge $G_{\mathbf{W}}$ the condition-specific edge sets E_1 and E_2.

3.5.3 Convex Optimization Formulation

To take advantage of the prior knowledge in the structural learning and avoid the potential bias introduced by knowledge, we formulate the problem into a convex optimization problem with sparsity constraints, and set the proper weights to achieve both the effectiveness of utilizing the domain knowledge and the robustness to the false positives in the knowledge.

Biological prior knowledge is incorporated into the network learning algorithm through re-weighting the penalties for the potential connections in the network. If supporting evidence for a connection between two genes is available in the prior knowledge, the algorithm will reduce the penalty for that edge (connection) parameter, making it more likely be detected.

To minimize the adverse effects of false positive edges induced by directly incorporating imperfect and non-specific prior knowledge in specific problems, the prior knowledge incorporation scheme carefully evaluates and controls the impact of false positives in the prior knowledge on the network inference results, and automatically selects the "optimal" degree of information fusion between the evidence in knowledge and the evidence in the data. More specifically, the robustness of the method is achieved by estimating and controlling the expected network inference deviation incurred by "random" knowledge via a sampling method. Since "random" knowledge has the maximum entropy distribution over the edges and introduces minimal information into the network inference, this implies that even under the worst scenario the network learning algorithm ably handles high degree of false positives in the prior knowledge. On the other hand, the algorithm is able to identify novel connections between genes without prior knowledge if there is strong evidence in the data supportive of these connections, making it capable of gaining new biological knowledge and insights from experimental data.

Again, we consider the p nodes in V as p random variables, and denote them as $X_1, X_2, \cdots, X_p$. Suppose there are N_1 samples under condition 1 and N_2 samples under condition 2. Without loss of generality, we assume $N_1 = N_2 = N$. Under the first condition, for variable X_i, we have observations $\boldsymbol{x}_i^{(1)} = [x_{1i}^{(1)}, x_{2i}^{(1)}, \cdots, x_{Ni}^{(1)}]^T$, $i = 1, 2, \cdots, p$, while under the second condition, we have $\boldsymbol{x}_i^{(2)} = [x_{1i}^{(2)}, x_{2i}^{(2)}, \cdots, x_{Ni}^{(2)}]^T$, $i = 1, 2, \cdots, p$. Further, let $\mathbf{X}^{(1)} = [\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_p^{(1)}]$ be the data matrix under condition 1 and $\mathbf{X}^{(2)} = [\boldsymbol{x}_1^{(2)}, \boldsymbol{x}_2^{(2)}, \cdots, \boldsymbol{x}_p^{(2)}]$ be the data matrix under condition 2.

Denote

$$\boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{x}_i^{(1)} \\ \boldsymbol{x}_i^{(2)} \end{bmatrix}, \tag{3.37}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}^{(2)} \end{bmatrix}, \tag{3.38}$$

and

$$\begin{aligned} \boldsymbol{\beta}_i &= \begin{bmatrix} \boldsymbol{\beta}_i^{(1)} \\ \boldsymbol{\beta}_i^{(2)} \end{bmatrix} \\ &= [\beta_{1i}^{(1)}, \beta_{2i}^{(1)}, \cdots, \beta_{pi}^{(1)}, \beta_{1i}^{(2)}, \beta_{2i}^{(2)}, \cdots, \beta_{pi}^{(2)}]^T. \end{aligned} \tag{3.39}$$

We formulate the problem of learning structural changes between two conditions as a convex optimization problem. Network structures under two conditions as well as their changes are simultaneously obtained by solving the optimization problem for each node (variable) X_i, $i = 1, 2, \cdots, p$, with an objective function

$$\begin{aligned} f(\boldsymbol{\beta}_i) = \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 + \lambda_1 \sum_{j=1}^{p} (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) \\ + \lambda_2 \|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1. \end{aligned} \tag{3.40}$$

The solution is acquired by minimizing (3.40),

$$\begin{aligned} \hat{\boldsymbol{\beta}}_i &= \arg\min_{\boldsymbol{\beta}_i} f(\boldsymbol{\beta}_i) \\ &= \arg\min_{\boldsymbol{\beta}_i^{(1)}, \boldsymbol{\beta}_i^{(2)}} \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 \\ &+ \lambda_1 \sum_{j=1}^{p} (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2 \|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1 \\ &\text{s.t. } \beta_{ii}^{(1)} = 0, \beta_{ii}^{(2)} = 0. \end{aligned} \tag{3.41}$$

Remark 1: The prior knowledge is explicitly incorporated into the formulation by W_{ji} and θ in the weighted ℓ_1-regulari-zation term, $\lambda_1 \sum_{j=1}^{p} (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|)$. θ is a ℓ_1 penalty

relaxation parameter taking value in $[0, 1)$, which reduces the penalty on the edges with supporting evidence in the prior knowledge while having no effects on edges with no knowledge.

Remark 2: From a Bayesian perspective, as pointed out in [27], the ℓ_1 penalty term $\|\boldsymbol{\beta}_i^{(c)}\|_1$ is equivalent to independent Laplace priors for the $\beta_{ji}^{(c)}$, $j = 1, 2, ..., p$, $c = 1, 2$, which follow

$$pdf(\beta_{ji}^{(c)}) = \frac{1}{2b} \exp\big(- \frac{-|\beta_{ji}^{(c)}|}{b}\big) \tag{3.42}$$

where $b = \frac{1}{\lambda_1(1-W_{ji}\theta)}$. Note that the Laplace distribution with a larger b has a larger variance $(2b^2)$ and thus distributes less mass around zero and more mass in the two tails. When supporting evidence for the edge between node i and node j is present in the prior knowledge, a non-zero θ adjusts the prior distribution for this edge $(\beta_{ji}^{(c)})$ with a larger b, and thereby makes this edge more likely be detected.

Remark 3: It is straightforward to show that the objective function (3.6) is non-differentiable, continuous, and convex.

The problem (3.41) can be solved efficiently by the block coordinate descent algorithm proposed previously. We repeat this procedure to each node X_i, $i = 1, 2, \cdots, p$. The non-zero elements of $\boldsymbol{\beta}_i^{(1)}$ indicate the neighbors of the i^{th} node under the first condition and the non-zero elements of $\boldsymbol{\beta}_i^{(2)}$ indicate the neighbors of the i^{th} node under the second condition. We use two condition-specific adjacency matrices $A^{(1)}$ and $A^{(2)}$ to represent the edge sets E_1 and E_2 under condition 1 and condition 2, respectively. Moreover, the non-zero elements of $A^{(1)} - A^{(2)}$ pinpoint the changed edge sets between two conditions.

3.5.4 Degree of Prior Knowledge Incorporation

The non-zero elements in $\mathbf{W}$ introduce knowledge to the objective function (3.40), and θ determines to what degree the knowledge will affect the inference. If there is no knowledge supporting the connection between X_j and X_i, the ℓ_1 penalty for $\beta_{ji}^{(c)}$ will remain unchanged

and the inference of the edge between X_j and X_i will be totally based on data. On the other hand, if the connection between X_j and X_i is present in the prior knowledge, θ will reduce the penalty applied to the corresponding $\beta_{ji}^{(c)}$. As a result, the connection between X_j and X_i will more likely be detected.

We hope to limit the adverse effects caused by the spurious edges in the prior knowledge, but we are unable to assess such effects in real applications, since we do not know the ground-truth. Instead, we control such adverse effects incurred in the worst-case scenario. The worst-case scenario of prior knowledge is that the knowledge is totally random. In this case, the entropy of the knowledge distribution over the edges is maximized and the information introduced to the inference algorithm is minimal. Incorporated with such random knowledge, the inference results will deviate from the purely data driven result. Then, θ is carefully chosen so that the expected deviation is controlled within acceptable range in the worst-case scenario.

To estimate the deviation of network inference results incurred by incorporating prior knowledge, we use graph edit distance as a measurement for the dissimilarity of two graphs [85]. Let $G_{\mathbf{X}} = (V, E_{\mathbf{X}})$ denote the graph learned purely from data, *i.e.* $\mathbf{W} = \mathbf{0}$, and $G_{\mathbf{X},\mathbf{W}_R,\theta}(V, E_{\mathbf{X},\mathbf{W}_R,\theta})$ denote the graph learned with prior knowledge. $\mathbf{W}_R$ indicates that the prior knowledge is "random". Let $d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})$ denote the graph edit distance between the original learned graph and the one learned with knowledge. Further, let $|E_{\mathbf{X}}|$ be the number of edges in the graph $G_{\mathbf{X}}$.

We determine the degree of prior knowledge incorporation by selecting the largest θ that controls the expected normalized deviation, $\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})/|E_{\mathbf{X}}|]$, under an acceptable threshold $\delta > 0$:

$$\begin{aligned} &\hat{\theta} = \max \theta \\ &\text{s.t.} \quad \frac{\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]}{|E_{\mathbf{X}}|} \leq \delta. \end{aligned} \tag{3.43}$$

We use a sampling-based algorithm to find the empirical distribution of $d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})$ and estimate the solution to problem (3.43).

When the prior knowledge is very dense, the prior knowledge matrix $\mathbf{W}$ is not a sparse matrix. In this case, at a given θ, more additional edges will likely be detected compared to the sparse $\mathbf{W}$ case, since we reduce the penalties for more potential edges. Subsequently, it will lead to a larger $\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]$. Therefore, at a given δ, the solution to problem (3.43) will be a smaller θ in the dense $\mathbf{W}$ case than the sparse $\mathbf{W}$ case, which means this method automatically reduces the degree of prior knowledge incorporation since the knowledge is not specific. If the knowledge is more specific, the algorithm will select a larger θ to give more weight on the information in the prior knowledge in the network learning.

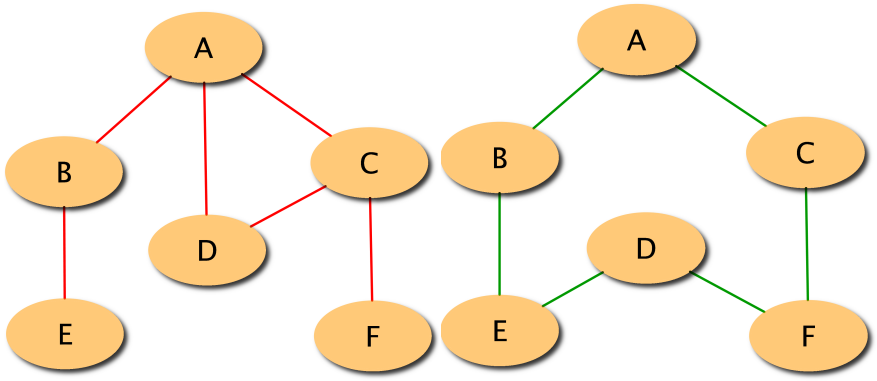
3.6 Experiments

3.6.1 A Synthetic Experiment

We first use a synthetic example to illustrate the principle and test the proposed method. Assume there are six nodes in the Gaussian graphical model, A, B, C, D, E, F. Under condition 1, their relationships are represented by Figure 3.2a. Under condition 2, their relationships are altered, as shown in Figure 3.2b. We generated 200 samples from the joint Gaussian distribution according to the Gaussian graphical model with the structure specified by Figure 3.2a, and 200 samples from the joint Gaussian distribution according to Gaussian graphical model with the structure specified by Figure 3.2b.

The penalty parameters are set to $\lambda_1 = 0.22$ and $\lambda_2 = 0.062$, calculated according to Section 4.4. Figure 3.3a is the composite network under two conditions inferred by the proposed algorithm, where the black lines are the edges that exist under both conditions, the red lines are the edges that exist only under condition 1 and the green lines are the edges that exist

only under condition 2. Since we are more interested in the changed part of the graph, we extracted the edges and nodes involved in the changes to highlight these structural changes. We term it differential sub-network, as shown in Figure 3.3b. We can see the proposed algorithm accurately captured the structural changes of the graphical model between two conditions.

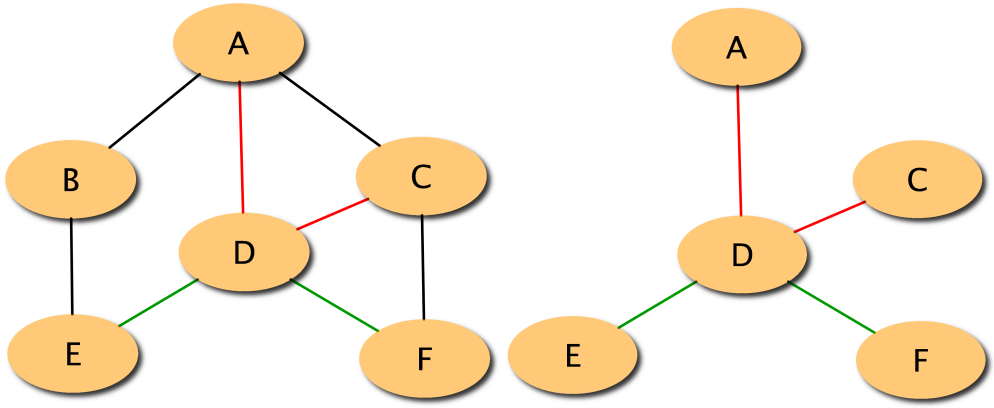


(a) Condition 1. (b) Condition 2.

Figure 3.2: The structures of the Gaussian graphical model under two conditions.

3.6.2 Algorithm Speed Comparison

We compared the running time between the proposed algorithm and a widely used, standard optimization package, CVXMOD. CVXMOD is a Python-based tool for expressing and solving convex optimization problems, developed by Jacob Mattingley, as PhD work under Stephen Boyd at Stanford University. It uses CVXOPT as its solver [86]. Our algorithm is implemented in Java. The core of the CVXOPT solver is implemented in C. We compared the running time between the two algorithms for the cases $n = 20, 40, 60, 80$ and $p = 100, 200, 500, 100$. The CPU of the computer used in this experiment is Intel Xeon Dual Core 3.00 GHz. As we can see from Table 3.1, under these 16 scenarios, the proposed algorithm is 106 to 19192 times faster than the CVXOPT solver. And on average, our algorithm



(a) Composite network. (b) Differential sub-network.

Figure 3.3: The network structure learned by the proposed method. The black lines are the edges that exist under both conditions. The red lines are the edges that exist only under condition 1. The green lines are the edges that exist only under condition 2.

is 5800 times faster than the standard convex optimization solver. Our algorithm also scales very nicely as n and p grow.

3.6.3 Algorithm Assessment by Precision and Recall Curves

In order to assess the effectiveness of our method, we construct a network with $p = 100$ nodes using the approach similar to the one described in [33], and then randomly modify 10% of the edges to create two condition-specific networks with sparse changes. Under each conditions, there are $n = 200$ samples. The parameter λ_2 is set using the heuristic approach proposed earlier. The parameter λ_1 controls the sparsity of learned networks. When λ_1 is smaller, the learned network is very dense; and when λ_1 is large, the learned network is very sparse. We increase the parameter λ_1 from 0.06 to 0.5 to examine the performance of the algorithm at different λ_1. Precision measures the fraction of edges in the learned network that are consistent with the ground-truth. Precision curve shows the accuracy of the our method in

Table 3.1: Running time comparison between the proposed algorithm and the CVXOPT solver.

p	n	Proposed algorithm (in seconds)	CVXOPT (in seconds)
20	100	0.009	0.9596
20	200	0.01	7.415
20	500	0.015	101.51
20	1000	0.02	795.76
40	100	0.015	2.576
40	200	0.019	8.559
40	500	0.03	148.4
40	1000	0.052	998.0
60	100	0.027	6.072
60	200	0.039	22.81
60	500	0.07	191.7
60	1000	0.108	1107.1
80	100	0.038	9.934
80	200	0.087	23.92
80	500	0.141	275.5
80	1000	0.23	852.1

learning condition-specific network structures. Recall measures the fraction of ground-truth edges that are successfully detected. The recall cure shows the sensitivity of our method to detect the network structures from the data. We repeat the simulation experiments for 200 times for each λ_1. The experiment results are shown in Figures 3.4 and 3.5. Each blue point in Figure 3.4 is the mean of the precision of 200 independent simulations. The

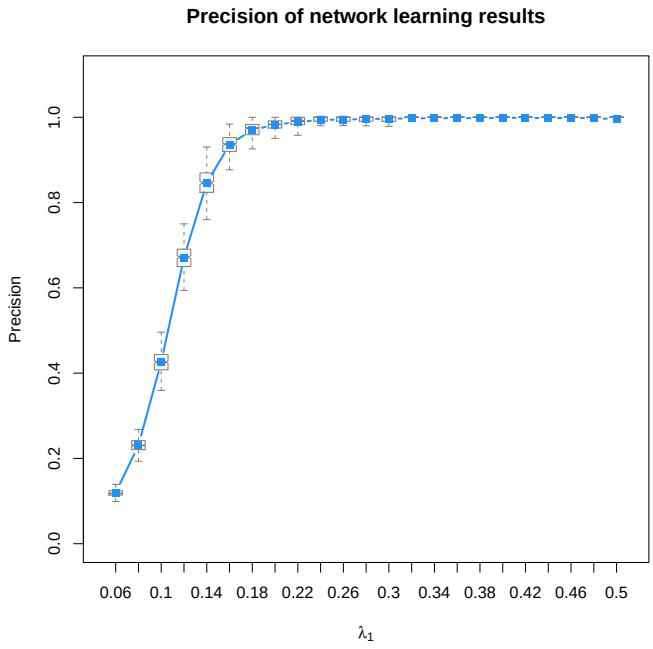


Figure 3.4: The precision curve of the proposed algorithm when $p = 100$, $n = 200$.

box plot at each point shows the five-number summaries of the precision results of these 200 simulations: the lowest datum still within 1.5 interquartile range (IQR) of the lower quartile, lower quartile (25%), median (50%), upper quartile (75%), and the highest datum still within 1.5 IQR of the upper quartile. Each blue point in Figure 3.5 is the mean of the recall of 200 independent simulations. The box plot in Figure 3.5 at each point shows the five-number summaries of recall results of these 200 simulations.

3.6.4 Experiment on Modeling Gene Regulatory Networks under Two Conditions

Inference of the structures of gene regulatory networks from expression data is a fundamental problem in computational biology. Our goal here is to infer and extract the structural changes of a gene regulatory network between two conditions using gene expression data. SynTReN is

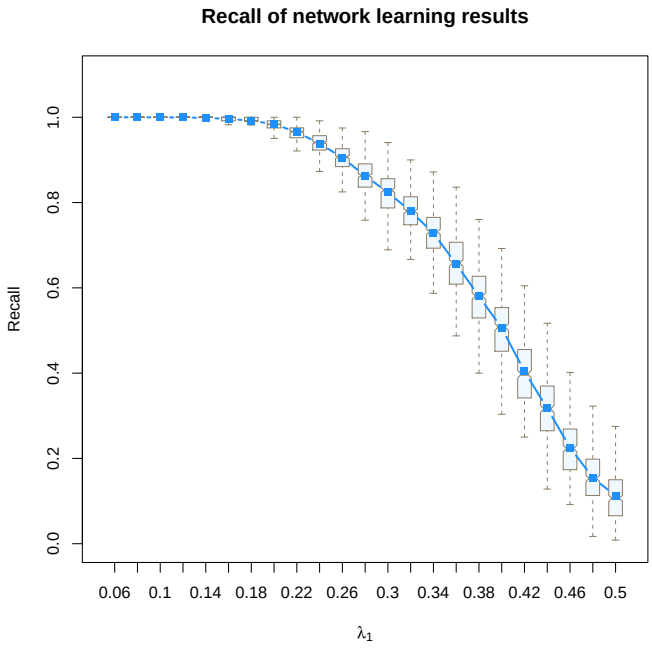
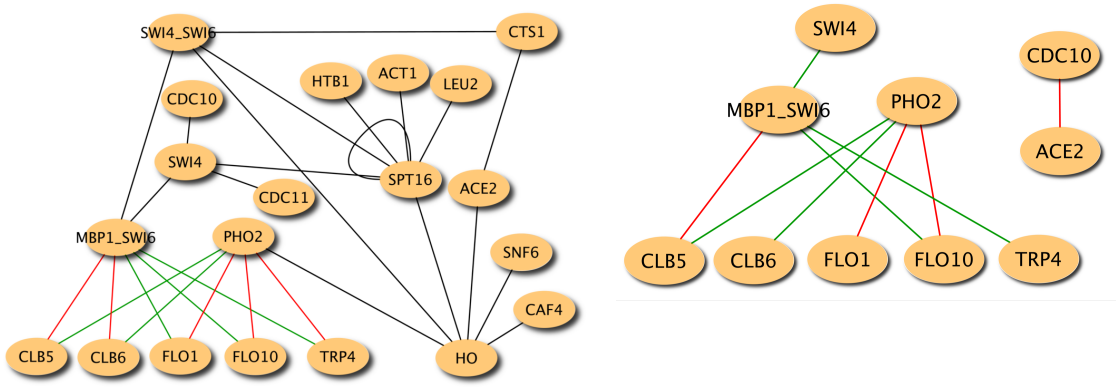


Figure 3.5: The recall curve of the proposed algorithm when $p = 100$, $n = 200$.

a network generator that creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximate experimental data, used as benchmarks for the validation of bioinformatics algorithms [42].

To test the applicability of the proposed framework in gene regulatory network modeling, we used the software SynTReN to generate one simulation dataset of 50 samples of a sub-network drawn from an existing signaling network in *Saccharomyces cerevisiae*. Then we changed part of network and used SynTReN to generate another dataset of 50 samples according to this modified network. The networks under two conditions is shown in Figure 3.6a. The network contains 20 nodes that represent 20 genes. The black lines indicate the regulatory relationships that exist under both conditions. The red and green lines are the regulatory relationships that exist only under condition 1 and condition 2, respectively. The sub-network comprised of nodes MBP1_SWI6, CLB5, CLB6, PHO2, FLO1, FLO10 and TRP4 and the green and red lines is the focus of our study that our algorithm tries to identify

from expression data.



(a) (b)

Figure 3.6: (a) The gene regulatory network under two conditions. Nodes in the network represent genes. Lines in the network indicate regulatory relationships between genes. The black lines are the regulatory relationships that exist under both conditions. The red and green lines represent the regulatory relationships that exist only under condition 1 and under condition 2, respectively. (b) The sub-network extracted by the proposed algorithm.

Figure 3.6b shows the differential sub-network between the two conditions extracted by the proposed algorithm. The penalty parameters are set to $\lambda_1 = 0.28$ and $\lambda_2 = 0.123$, calculated according to Section 4.4. Compared with the known network topology shown in Figure 3.6a, the proposed algorithm correctly identified all the nodes with structural changes and 7 of 10 differential edges. The edge between CDC10 and ACE2 was falsely detected. This indicates that our algorithm can successfully detect these interesting genes using their network structure information, even though the means of their expressions did not change substantially between the two conditions. Therefore, this method is able to identify biomarkers that cannot be detected by traditional gene ranking methods, providing a complimentary approach for biomarker identification problem.

3.7 Conclusions

In this chapter, we reported an effective learning strategy to extract structural changes in Gaussian graphical models in controlled experiments. We presented a convex optimization framework using ℓ_1-regularization to formulate this problem, and introduced an efficient block coordinate descent algorithm to solve it. We demonstrated the effectiveness of the approach on a numerical simulation experiment, and then we applied the algorithm to detecting gene regulatory network structural changes under two conditions and obtained very promising results. Additionally, this method can be extended to incorporating biological prior knowledge, which can efficiently utilize prior knowledge in the network inference while remaining robust to the false positive edges in the knowledge.

Chapter 4

Theoretical Analysis on Echo State Networks and Application to Modeling Gene Expression Time Course Data

4.1 Introduction

Recurrent neural networks (RNNs) are widely used to model nonlinear dynamical systems. Recently, a new framework for RNNs, namely echo state networks (ESNs), was proposed by H. Jaeger *et al.* [87, 88]. ESNs (and closely-related liquid state machines, independently proposed by Maass *et al.* [89]) share some features characteristic of models for learning in biological brains and they exhibit superior performance when used as "black-box" time-series models. In an ESN, neurons in a fixed (non-trainable) recurrent layer, known as "the reservoir", are driven by the input signals, and the trainable output neurons combine the

output of the excited reservoir state to generate task-specific temporal patterns. This new RNN paradigm is also known as "reservoir computing".

ESNs have drawn great interest from the research community and have been successfully applied to various tasks, *e.g.* chaotic time series prediction [90], communications channel equalization [87], dynamical pattern recognition [91, 92], and gene regulatory network modeling [93]. Various ESN schemes have been explored, including a small-world recurrent neural system with scale-free distribution [94], decoupled ESNs with lateral inhibition [95], and ESNs with uniformly distributed poles and adaptive bias [96]. Lukoševičius and Jaeger presented a comprehensive review on the theoretical results and applications of ESNs in [97].

The salient difference from traditional RNNs [98,99] is that an ESN employs a large number of *randomly* connected neurons (usually on the order of 50 to 1000), namely the "reservoir", *i.e.*, unlike traditional RNNs, the connection weights between neurons in the recurrent (reservoir) layer do not require any supervised training – only connection weights to output neurons are optimized. Thus, training is greatly simplified compared to traditional RNNs and well-known RNN training problems of slow convergence, even lack of convergence, and local minima are avoided. In fact, if the ESN employs a linear activation function in the output layer, ESN training reduces to a simple linear regression problem.

The working principle of an ESN derives from an important algebraic property of the reservoir, namely the *echo state property* (ESP). A recurrent reservoir driven by an external input signal has the echo state property if the reservoir states are systematic variations of the input driving signal. Essentially, satisfying the ESP means that the effect of both previous states and previous inputs on a future state will gradually vanish (*i.e.* neither persist nor become amplified) as time passes [88]. If the ESP holds, the reservoir network state will asymptotically (in time) depend only on the input history and the nonlinear system will be well-approximated through a linear combination of the reservoir's "echo state" signals. Metaphorically, under the ESP, the reservoir state signal can be thought of as an "echo" of

the input history.

Jaeger presented both a necessary condition (under the assumption that the input space includes the zero sequence) and a sufficient condition for the ESP [88]. Buehner and Young proposed a less restrictive sufficient condition based on minimizing the matrix operator **D**-norm over the set of diagonal matrices [100]. However, these papers did not consider the unique characteristic of the reservoir, *i.e.* that it is *randomly* generated. Here, by exploiting this fact and applying results from random matrix theory, we will show that the sufficient conditions in [88] and [100] are rather conservative.

The topology of the reservoir in ESNs has been of great research interest, with the classical form a randomly generated and sparsely connected network [87, 88]. Several attempts have been made to search for a better topology – the small-world, scale-free, and biologically inspired reservoir topologies. However, quoting [97], "none of the investigated network topologies was able to perform significantly better than simple random networks, both in terms of eigenvalue spread as well as testing error".

The novel contributions of this work are threefold. First, motivated by the above quotation, we analytically examine the essential characteristics of random reservoirs. We apply recent results from random matrix theory to demonstrate the asymptotic distributions of eigenvalues and singular values of reservoir weight matrices. We then show that randomly generated reservoirs, either sparsely or fully connected, either with Bernoulli or Gaussian connection weights (or, in fact, with weights distributed according to other density families), are all expected to behave similarly. These results thus explain the above quoted observation from [97]. Second, we quantify the gap between the scaling factor bounds used to define the ESP necessary and sufficient conditions proposed in previous works. We show that, asymptotic in the size of the reservoir, this gap becomes quite large, with the necessary condition bound twice as large as the sufficient condition bound. Finally, we show that when the spectral radius of the reservoir weight matrix is smaller than 1 (the *necessary* condition for

the ESP when the input space contains the zero sequence), the state transition mapping is in fact contractive with high probability, given a sufficiently large reservoir. This result corroborates the observation in [88] that the necessary condition for the echo state property is often good enough in practice, such that violations of the ESP are not practically observed. This result, together with the factor of two asymptotic gap between the scaling factor bounds, indicates the conservativeness of the sufficient conditions from [88] and [100]. The practical implication of these results is that standard ESN design approaches, based on use of the sufficient conditions, are suboptimal – use of a conservative scaling factor compromises the amount of memory in the RNN, and thus the ability to accurately model a given target dynamical system.

The remainder of this chapter is organized as follows. In Section 4.2, we revisit the ESN model, random reservoirs, and the ESP. This is followed by detailed discussion in Section 4.3 on relevant results from random matrix theory, the properties of random reservoirs, and the gap between the sufficient and necessary conditions previously proposed for the ESP. In Section 4.4, we prove that the necessary condition for the ESP ensures the state transition mapping is contractive with high probability. We briefly conclude our work in Section 4.7.

4.2 The Echo State Network Formulation

4.2.1 Basic ESN Formulation

A typical ESN is shown in Figure 4.1. It can be represented by state update and output equations. While enhanced representation power for an RNN may be achieved by the use of output feedback, this can also introduce instability problems [97, 101]. To avoid these issues and also to simplify the mathematical analysis, we will focus in this work on ESNs without output feedback, as also adopted by others [88, 100]. Thus, the activation of internal units

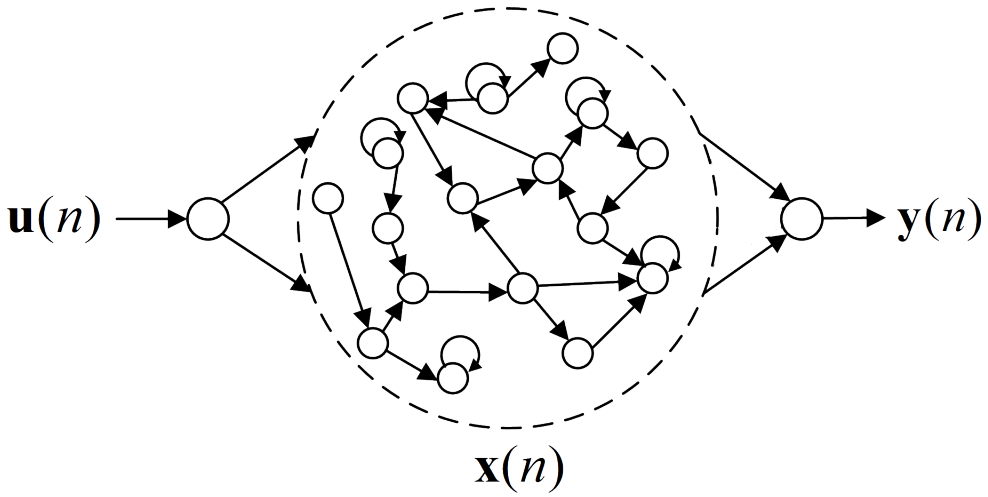


Figure 4.1: Illustration of an echo state network.

is updated according to

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1)), \tag{4.1}$$

where $\mathbf{x}$ is a $N \times 1$ vector of the reservoir state, $\mathbf{W}$ is a $N \times N$ reservoir weight matrix, $\mathbf{W}_{\text{in}}$ is an $N \times N_{\text{in}}$ input weight matrix, $\mathbf{u}$ is a $N_{\text{in}} \times 1$ vector of system inputs, $\mathbf{y}$ is a $N_{\text{out}} \times 1$ vector of system outputs, and f is the neuron activation function (usually a tanh sigmoid function), applied component-wise.

For notational convenience, we denote the state transition equation by

$$\begin{aligned} \mathbf{x}(n+1) &= T(\mathbf{x}(n), \mathbf{u}(n+1)) \\ &= f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1)), \end{aligned} \tag{4.2}$$

and the output equation by

$$\mathbf{y}(n) = g(\mathbf{W}_{\text{out}} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{u}(n) \end{bmatrix}), \tag{4.3}$$

where $\mathbf{W}_{\text{out}}$ is the $N_{\text{out}} \times (N + N_{\text{in}})$ output weight matrix, and g is usually a tanh sigmoid or an identity function, applied component-wise.

4.2.2 Random Reservoirs in ESNs

A salient feature that distinguishes ESNs from conventional RNNs is the use of large, fixed random reservoirs. The classical ESN reservoir topology is a randomly generated and sparsely connected network [87]. It was thought that "this condition lets the reservoir decompose into many loosely coupled subsystems, establishing a richly structured reservoir of excitable dynamics" [87]. Nevertheless, this is not generally true and it has in fact been reported that fully connected reservoirs work just as well as sparsely connected ones [101]. Such observation leads to inquiry of the essential characteristics of random reservoirs and their role in approximating nonlinear dynamical systems.

The types of random reservoirs are characterized by the structure of the reservoir weight matrix. Assume the matrix $\mathbf{W} = \alpha \mathbf{W}_N$, where α is a properly chosen global scaling factor (whose utility will be discussed later), and where the elements of the matrix $\mathbf{W}_N$ are random variables that are independent and identically distributed (i.i.d.). Here we consider the following three types of reservoir weight matrices.

Sparse random reservoir: This is the most common type of random reservoir in ESNs [87,88]. The random variable w (which characterizes each element of $\mathbf{W}_N$) follows the modified Bernoulli probability mass function (pmf)

$$\begin{cases} \Pr(w = 0) = 1 - c \\ \Pr(w = \pm 1) = c/2 \end{cases}, \tag{4.4}$$

where $\Pr(\cdot)$ denotes probability of an event and c is "the connectivity" of the reservoir. Note that if $\mathbf{W}_N[i, j] = 0$, there is no connection from reservoir neuron i to reservoir neuron j. Thus, using the modified Bernoulli pmf leads to a realization of $\mathbf{W}$ that is sparsely connected.

Fully-connected Gaussian random reservoir: w follows a standard normal distribution

$$w \sim N(0, 1). \tag{4.5}$$

Fully-connected Bernoulli random reservoir: w follows the Bernoulli distribution

$$\Pr(w = \pm 1) = 1/2. \tag{4.6}$$

These three types of reservoir weight matrices exhibit different network topologies, *i.e.*, either sparsely connected or fully connected neurons in the reservoir, and different types of weights, *i.e.*, either continuous-valued or discrete-valued. All three types have been used as random reservoirs in ESNs and have been successfully applied.

4.2.3 Definition of Echo State Property

In order to work properly, an echo state network should possess the ESP, as defined in [88].

Definition 1 (Jaeger [88])**.** *Assume standard compactness conditions, i.e. inputs drawn from a compact input space U and network states restricted to a compact set A. Assume that the network has no output feedback connections. Then, the network has echo states if the network state $\mathbf{x}(n)$ is uniquely determined by any left-infinite input sequence $\bar{\mathbf{u}}^{-\infty}$. More precisely, this means that for every input sequence, $\cdots, \mathbf{u}(n-1), \mathbf{u}(n) \in U^{-\mathbb{N}}$, for all state sequence pairs $\cdots, \mathbf{x}(n-1), \mathbf{x}(n) \in A^{-\mathbb{N}}$ and $\cdots, \mathbf{x}'(n-1), \mathbf{x}'(n) \in A^{-\mathbb{N}}$, where $\mathbf{x}(k) = T(\mathbf{x}(k-1), \mathbf{u}(k))$, $\mathbf{x}'(k) = T(\mathbf{x}'(k-1), \mathbf{u}(k))$, and $\mathbb{N}$ is the set of natural numbers, it holds that $\mathbf{x}(n) = \mathbf{x}'(n)$.*

The definition of the ESP implies that similar echo state sequences must represent similar input histories. In [88], Jaeger also provided several equivalent characterizations of echo states, *e.g.* the properties of being state contracting, state forgetting, and input forgetting. However, the ESP definition is hard to check in practice. A known *sufficient* algebraic condition for the ESP is that the largest singular value of $\mathbf{W}$ (defined as the square root of the largest eigenvalue of $\mathbf{W}\mathbf{W}^T$) is smaller than 1. On the other hand, the ESP is violated (for input space containing the zero sequence) when the spectral radius of $\mathbf{W}$ (defined as

its largest magnitude eigenvalue) is greater than 1. Therefore, the spectral radius of $\mathbf{W}$ restricted to being less than or equal to 1 serves as a *necessary* condition for the ESP. The following theorem formally presents these two conditions for the network to possess the ESP.

Theorem 2 (Jaeger [88])**.** *Assume a sigmoid network, i.e. with* $f = \tanh$*, applied component-wise. (a) Let the weight matrix* $\mathbf{W}$ *satisfy* $\sigma_{\max} < 1$*, where* $\sigma_{\max}$ *is its largest singular value. Then* $d(T(\mathbf{x}, \mathbf{u}), T(\mathbf{x}', \mathbf{u})) < d(\mathbf{x}, \mathbf{x}')$ *for all inputs* $\mathbf{u} \in U$*, for all states* $\mathbf{x}, \mathbf{x}' \in [-1, 1]^N$*, where* $d(\cdot, \cdot)$ *is any distance metric. This implies the ESP holds. (b) Let the weight matrix have spectral radius* $|\lambda_{\max}| > 1$*, where* $\lambda_{\max}$ *is the eigenvalue of* $\mathbf{W}$ *with the largest absolute value. Then the network has an asymptotically unstable null state. This implies that it does not satisfy the ESP for input space* U *containing* $\mathbf{0}$ *and admissible state space* $A = [-1, 1]^N$*.*

As suggested in [88], a convenient strategy to obtain ESNs is to start with some weight matrix $\mathbf{W}_N$ and then select a global scaling factor α to suitably define $\mathbf{W} = \alpha \mathbf{W}_N$. Let $\sigma_{\max}(\mathbf{W}_N)$ and $|\lambda_{\max}(\mathbf{W}_N)|$ denote the largest singular value and the spectral radius of $\mathbf{W}_N$, respectively. Then, according to [88], for the ESP to hold, the sufficient condition is $\alpha < \sigma_{\max}^{-1}(\mathbf{W}_N)$ and the necessary condition is $\alpha < |\lambda_{\max}^{-1}(\mathbf{W}_N)|$.

Furthermore, although the existence of the ESP for $\alpha \in [\sigma_{\max}^{-1}(\mathbf{W}_N), |\lambda_{\max}^{-1}(\mathbf{W}_N)|]$ has not been theoretically proved, it has been observed, albeit without analytical justification, that "one obtains echo states even when α is only marginally smaller than $|\lambda_{\max}^{-1}(\mathbf{W}_N)|$" and "the sufficient condition is very restrictive" [88].

Buehner and Young proposed a tighter sufficient condition for the ESP. The main idea is to minimize the matrix operator $\mathbf{D}$-norm over the set of diagonal matrices [100]. The $\mathbf{D}$-norm of a vector $\mathbf{x} \in \mathbb{R}^N$ is defined to be $\|\mathbf{x}\|_{\mathbf{D}} = \|\mathbf{D}\mathbf{x}\|$, where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is nonsingular. Then, the matrix operator $\mathbf{D}$-norm (the induced $\mathbf{D}$-norm) of a matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is given by

$$\|\mathbf{W}\|_{\mathbf{D}} = \sigma_{\max}(\mathbf{D}\mathbf{W}\mathbf{D}^{-1}).$$

However, because the matrix $\mathbf{D}$ does not have full structure (and in fact was restricted to being diagonal), the sufficient condition derived in [100] is still in general conservative. Pertinent to the sequel, we observe that the derivations of the existing results on the sufficient condition [88, 100] have not taken into account the primary unique characteristic of an ESN, *i.e.* that the reservoir matrix is a *random* matrix.

4.3 Random Matrix Theory and Random Reservoirs

In this section, we first introduce some recent results in random matrix theory, and then apply them to characterize some relevant properties of random reservoirs in ESNs.

4.3.1 The Empirical Spectral Distribution of Random Matrices

Let

$$\mu_{\mathbf{W}_N}(s,t) := \frac{1}{N}|\{i|1 \leq i \leq N, \mathrm{Re}(\lambda_i) \leq s, \mathrm{Im}(\lambda_i) \leq t\}| \tag{4.7}$$

be the *empirical spectral distribution* (ESD) of $\mathbf{W}_N$'s eigenvalues $\lambda_i \in \mathbb{C}$, $i = 1, ..., N$, where $|\cdot|$ denotes the cardinality of the set and $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ are the real and imaginary parts of the complex number, respectively. A well-known conjecture is *the circular law of random matrices*, which states that asymptotically, as N gets large, the eigenvalues of a properly normalized random matrix $\mathbf{W}_N$ are uniformly distributed on the unit disk in the complex plane. After many pioneering efforts in proving the circular law for various scenarios, including sparse random matrices [102–106], it was proved in full generality, in both weak and strong forms, quite recently [107].

Theorem 3 (Circular Law [107])**.** *Let $\mathbf{W}_N$ be the $N \times N$ random matrix whose entries are i.i.d. complex random variables with mean 0 and variance 1. Define $\mathbf{W} = \frac{1}{\sqrt{N}}\mathbf{W}_N$. Then*

the ESD of $\mathbf{W}$ *converges (in both the strong and weak senses) to the uniform distribution on the unit disk, as* $N \to \infty$.

Corollary 1. *The ESDs of reservoir weight matrices* $\mathbf{W}$ *as defined in* (4.4) *with the scaling factor* $\alpha = \frac{1}{\sqrt{cN}}$, (4.5) *with the scaling factor* $\alpha = \frac{1}{\sqrt{N}}$, *and* (4.6) *with the scaling factor* $\alpha = \frac{1}{\sqrt{N}}$ *all have the same limit distribution and, more specifically, converge (in both the strong and weak senses) to the uniform distribution on the unit disk.*

The circular law implies that when N is sufficiently large (as is typical for ESNs), the eigenvalues of $\mathbf{W}$ spread out evenly over the unit disk in the complex plane, independent of the specific distribution of w, as we illustrate in Figure 4.2. It is also important to note that when a sparse reservoir is used in ESNs, the connectivity c of the sparse reservoir weight matrix must satisfy the inequality $c > N^{-1+\epsilon_1}$, where $\epsilon_1 > 0$ is a small positive constant, because otherwise, with non-negligible probability, the sparse reservoir weight matrix would lose its rank-efficiency as N gets large (Theorem 1.3 in [106]).

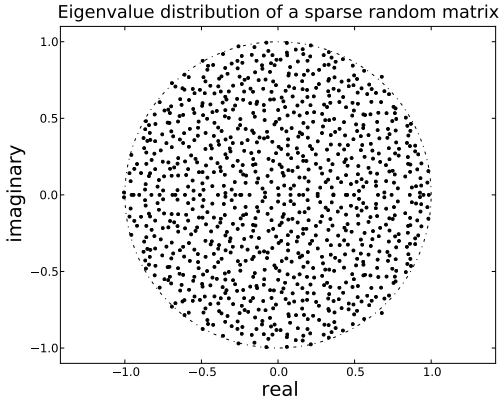
4.3.2 Singular Values of Random Matrices

Similarly, let $\sigma_1, \sigma_2, ..., \sigma_N$ be the singular values of $\mathbf{W}$. The empirical distribution of the squares of the singular values of $\mathbf{W}$ is defined by

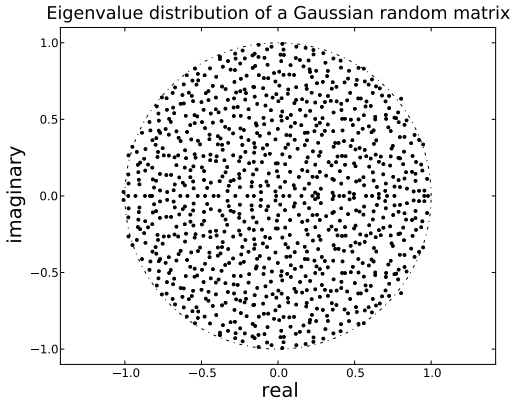
$$\nu_{\mathbf{W}}(t) := \frac{1}{N}|\{i|1 \leq i \leq N, \sigma_i^2 \leq t\}| \tag{4.8}$$

It has been shown that $\nu_{\mathbf{W}}$ is governed by the *Marchenko-Pastur law* [108–110].

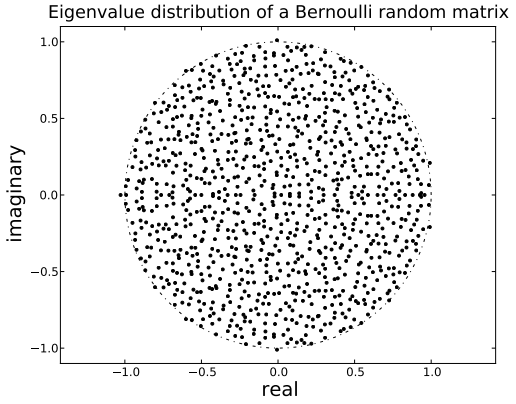
Theorem 4 (Marchenko-Pastur Law)**.** *Let* $\mathbf{W}_N$ *be the* $N \times N$ *random matrix whose entries are i.i.d. complex random variables with mean 0 and variance 1. Define* $\mathbf{W} = \frac{1}{\sqrt{N}}\mathbf{W}_N$. *Then the empirical distribution of the squares of the singular values of* $\mathbf{W}$, $\nu_{\mathbf{W}}(t)$, *converges (both in the sense of probability and in the almost sure sense) to* $\frac{1}{2\pi}\int_0^{\min(t,4)} \sqrt{\frac{4}{x} - 1}dx$, *as* $N \to +\infty$.



(a) A sparse random matrix



(b) A Gaussian random matrix



(c) A Bernoulli random matrix

Figure 4.2: The empirical eigenvalue distributions of three types of random matrices ($N = 1000$).

Remark: Supported by rigorous mathematical proofs, the circular and Marchenko-Pastur laws reveal an important, fundamental property of random matrices, *i.e.* that both the eigenvalues and the singular values of random reservoir weight matrices have unique limit distributions, independent of the distribution and connectivity of w, as $N \to \infty$.

4.3.3 The Gap between the Sufficient and Necessary Conditions

As well-discussed in [88] and as aforementioned in Subsection II.C, the global rescaling factor α must be properly chosen to ensure the ESP for $\mathbf{W} = \alpha \mathbf{W}_N$. Specifically, when $\alpha < |\lambda_{\max}^{-1}(\mathbf{W}_N)|$, the system is stable, which serves as the necessary condition (assuming the input space contains the zero sequence); when $\alpha < \sigma_{\max}^{-1}(\mathbf{W}_N)$, the ESP is guaranteed, *i.e.* this serves as the sufficient condition. However, the sufficient condition $\alpha < \sigma_{\max}^{-1}(\mathbf{W}_N)$ is considered conservative, with the practical implication being that the associated ESN design will be suboptimal, with the amount of memory in the dynamical system compromised (the smaller α, the shorter the system memory). In fact, it has been observed that "one obtains echo states even when α is only marginally smaller than $|\lambda_{\max}^{-1}(\mathbf{W}_N)|$" [88].

The discrepancy between the theoretical sufficient condition for the ESP and the empirical observation that the necessary condition often works well in practice raises a natural question: how big is the gap between $\sigma_{\max}^{-1}(\mathbf{W}_N)$ and $|\lambda_{\max}^{-1}(\mathbf{W}_N)|$]? Let the ratio $r = \frac{\sigma_{\max}(\mathbf{W}_N)}{|\lambda_{\max}(\mathbf{W}_N)|}$ quantify the gap between the sufficient and necessary condition bounds. It turns out that this gap is quite large: the asymptotic value of r is 2 as $N \to \infty$.

Before we give the proof of this result, we first introduce two theorems from the random matrix theory literature.

Theorem 5 (Bai [111])**.** *Let* $\{w_{ij} : i = 1, 2, ..., N, j = 1, 2, ..., N\}$ *be i.i.d. random variables, and* $\mathbf{W}_N$ *be the* $N \times N$ *matrix* $(w_{ij})_{N \times N}$*,* $i, j = 1, 2, ..., N$*. Suppose (a)* $E[w_{11}] = 0$*, (b)* $E[w_{11}^2] = \sigma^2$*, and (c)* $E[|w_{11}|^4] < \infty$*. Then* $\limsup_{N \to \infty} \max_{1 \le i \le N} |\lambda_i(\mathbf{W}_N / \sqrt{N})| \le \sigma$ *a.s.,*

where $\lambda_i(\mathbf{W}_N/\sqrt{N})$, $i = 1, 2, ..., N$, *are eigenvalues of* $\mathbf{W}_N/\sqrt{N}$.

Theorem 6 (Yin [112]). *Let* $\{w_{ij} : i = 1, 2, ..., N, j = 1, 2, ..., N\}$ *be i.i.d. random variables, and* $\mathbf{W}_N$ *be the* $N \times N$ *matrix* $(w_{ij})_{N\times N}$, $i, j = 1, 2, ..., N$. *Suppose (a)* $E[w_{11}] = 0$, *(b)* $E[w_{11}^2] = \sigma^2$, *and (c)* $E[|w_{11}|^4] < \infty$. *Let* $\overline{\sigma}_N^2$ *be the largest singular value of* $\mathbf{W}_N/\sqrt{N}$. *Then* $\lim_{N\to\infty} \overline{\sigma}_N^2 = 4\sigma^2$ *a.s.*

Theorem 7 (Gap between the Sufficient and Necessary Conditions). *If the random reservoir weight matrix is generated according to* (4.4), (4.5), *or* (4.6), *then* $r \xrightarrow{a.s.} 2$, *as* $N \to +\infty$.

Proof. First, it is straightforward to verify that the three distributions specified by (4.4), (4.5) and (4.6) all have zero mean and finite fourth-moment, and their variances are $c, 1$, and 1, respectively.

We consider random reservoir weight matrices generated according to (4.5) and (4.6). From Theorem 5, we have

$$|\lambda_{\max}(\frac{1}{\sqrt{N}}\mathbf{W}_N)| \leq 1, \quad \text{almost surely, as } N \to +\infty. \tag{4.9}$$

Then, combining (4.9) with the conclusion of the circular law, we have

$$|\lambda_{\max}(\frac{1}{\sqrt{N}}\mathbf{W}_N)| \xrightarrow{a.s.} 1, \quad \text{as } N \to +\infty \tag{4.10}$$

Next, from Theorem 6, we have

$$\sigma_{\max}(\frac{1}{\sqrt{N}}\mathbf{W}_N) \xrightarrow{a.s.} 2, \quad \text{as } N \to +\infty \tag{4.11}$$

Therefore, we have

$$\begin{aligned} r =& \frac{\sigma_{\max}(\mathbf{W}_N)}{|\lambda_{\max}(\mathbf{W}_N)|} = \frac{\sigma_{\max}(\frac{1}{\sqrt{N}}\mathbf{W}_N)}{|\lambda_{\max}(\frac{1}{\sqrt{N}}\mathbf{W}_N)|} \\ & \xrightarrow{a.s.} 2, \quad \text{as } N \to +\infty \end{aligned} \tag{4.12}$$

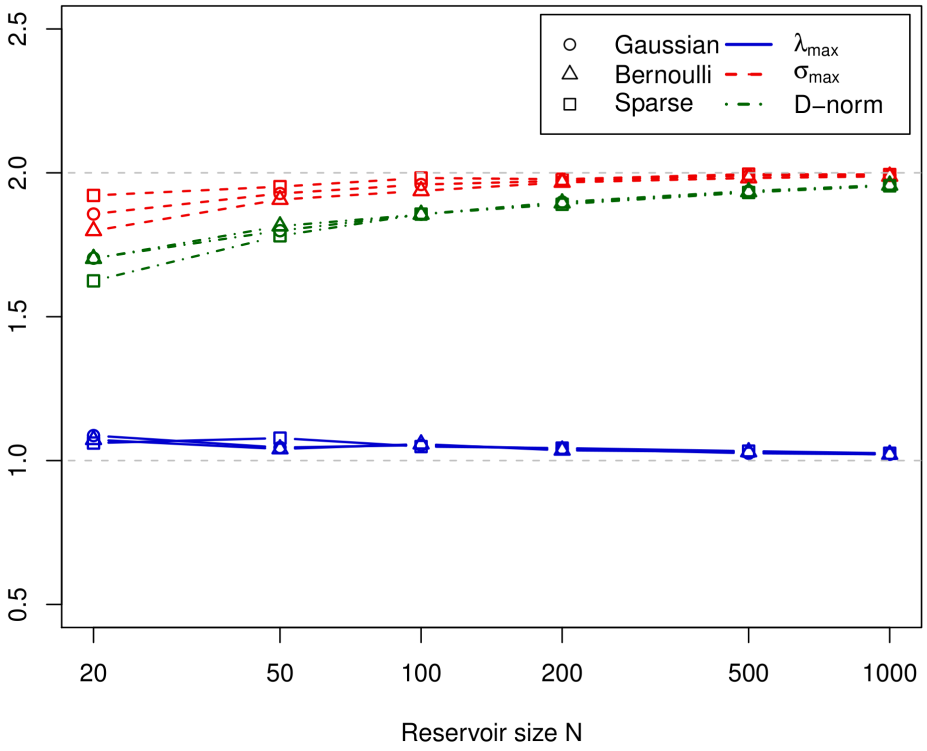


Figure 4.3: A simulation study on $\sigma_{\max}(\mathbf{W})$, $\lambda_{\max}(\mathbf{W})$, and $\|\mathbf{W}\|_D$ for Gaussian, Bernoulli, and sparse reservoirs, respectively, as N increases.

For the case of random reservoir weight matrices generated according to (4.4), if we replace $\frac{1}{\sqrt{N}}$ by $\frac{1}{\sqrt{cN}}$ in the above equations, it is straightforward to show the same conclusion stated in (4.12).

□

Figure 4.3 illustrates the asymptotic trend of $\sigma_{\max}(\mathbf{W})$, $\lambda_{\max}(\mathbf{W})$, and $\|\mathbf{W}\|_{\mathbf{D}}$ for Gaussian, Bernoulli, and sparse reservoir weight matrices as N increases. Each point in Figure 4.3 is the average of 20 independent simulations, and $\|\mathbf{W}\|_{\mathbf{D}}$ is calculated using MATLAB's μ-analysis Toolbox as suggested in [100]. First, we can see in Figure 4.3 that when N is large, Gaussian, Bernoulli, and sparse reservoirs all have similar respective values for $\sigma_{\max}(\mathbf{W})$, $\lambda_{\max}(\mathbf{W})$, and $\|\mathbf{W}\|_{\mathbf{D}}$. Second, as N increases, $\sigma_{\max}(\mathbf{W})$ tends to 2, and $\lambda_{\max}(\mathbf{W})$ tends to

1. Thus, consistent with Theorem 4, the bound for the necessary condition is about twice the bound for the sufficient condition for an ESN to possess the ESP as N gets large. Also, although we do not have theoretical results suggesting this, we observe in Figure 4.3 that $\lambda_{\max}(\mathbf{W})$ is approaching its asymptote from above, while $\sigma_{\max}(\mathbf{W})$ approaches its asymptote from below. That is, the gap, and thus the level of conservativeness (and the associated degree of potential suboptimality in using the sufficient condition in ESN design, relative to a design based on the necessary condition), is empirically observed to increase with N. Third, for the sufficient bound proposed in [100], $\|\mathbf{W}\|_{\mathbf{D}}$ is indeed tighter than $\sigma_{\max}(\mathbf{W})$ when N is small, for example for $N = 20$, but $\|\mathbf{W}\|_{\mathbf{D}}$ approaches very close to $\sigma_{\max}(\mathbf{W})$ as N gets large. Thus, empirically from Figure 3, there appears to be little to gain in using the sufficient condition from [100], rather than the sufficient condition from [88], as N gets large.

4.4 Why the Necessary Condition for Echo States Is Often "Sufficient in Practice"

To establish the sufficient condition for the ESP, Jaeger in [88] and Buehner and Young in [100] showed that, with $\mathbf{W}$ scaled to have its largest singular value less than one, the distance between two states $\mathbf{x}(n)$ and $\tilde{\mathbf{x}}(n)$ shrinks at every time step, *i.e.*, $d(T(\mathbf{x}(n), \mathbf{u}(n+1)), T(\tilde{\mathbf{x}}(n), \mathbf{u}(n+1))) < d(\mathbf{x}(n), \tilde{\mathbf{x}}(n))$, regardless of the input. This Lipschitz condition results in echo states.

In this section, alternatively, we will show that, asymptotically, as the size of the reservoir grows, for a much *less* conservative scaling of $\mathbf{W}$ that is *essentially* equivalent to scaling $\mathbf{W}$ just enough so that the necessary condition for the ESP is satisfied, the state transition mapping $T(\cdot, \cdot)$ is contractive with high probability, regardless of the input. In essence, we

will thus show that the necessary condition is "sufficient in practice". In order to make our mathematical analysis tractable and, thus, to establish our results, we consider a slightly unorthodox (albeit a still reasonable) procedure for scaling of the matrix, $\mathbf{W}$. Normally, and as considered in [88], one first randomly generates the matrix $\mathbf{W}_N$ and then sets $\mathbf{W} = \alpha\mathbf{W}_N$, where α is specifically chosen to satisfy an ESP condition – choosing $\alpha \leq |\lambda_{\max}^{-1}(\mathbf{W}_N)|$ meets the necessary condition, while setting $\alpha \leq \sigma_{\max}^{-1}(\mathbf{W}_N)$ ensures sufficiency. While choosing α in this way strictly ensures one (or both) of these ESP conditions, it also makes α a function of the random matrix, – *i.e.*, α is itself a random variable, with, moreover, a distribution that is dependent on N. Choosing α in this way will complicate our analysis. Alternatively, from (4.10), we know that if we choose $\mathbf{W} = (\rho/\sqrt{N})\mathbf{W}_N$, the spectral radius of $\mathbf{W}$ converges to ρ as $N \to \infty$. That is, picking a *constant* scaling factor $\rho < 1$, *independent* of both the dimension N and the particular realization of the random matrix $\mathbf{W}_N/\sqrt{N}$, satisfies the necessary condition for the ESP almost surely as N gets large. From this standpoint, choosing $\mathbf{W}$ in this "unconventional" way – one that is more amenable to analysis – is reasonable. More significantly, in the following, we will show that, by choosing $\mathbf{W}$ in this unconventional way, the state transition mapping $T(\cdot,\cdot)$ is contractive with high probability, regardless of the input. More specifically, for $\mathbf{x}(n), \tilde{\mathbf{x}}(n) \in [-1,1]^N$ and a random reservoir weight matrix $\mathbf{W} = (\rho/\sqrt{N})\mathbf{W}_N$, $\rho < 1$, the inequality $d(T(\mathbf{x}(n),\mathbf{u}(n+1)), T(\tilde{\mathbf{x}}(n),\mathbf{u}(n+1))) < d(\mathbf{x}(n),\tilde{\mathbf{x}}(n))$ holds with probability $1 - O(e^{-C_\rho N})$, where the constant C_ρ depends on ρ. In this sense, we show that asymptotically, for large N, the necessary condition is "*sufficient in practice*". Finally, although our theoretical results will assume an unconventional procedure for scaling $\mathbf{W}$, we will subsequently demonstrate at least *empirically* that "sufficiency of the necessary condition in practice" also applies if one uses the more standard procedure for scaling $\mathbf{W}$.

A key ingredient for establishing our results is the *concentration of measure phenomenon* [113] – *i.e.*, the fact that when projecting a state vector $\mathbf{x}$ onto the properly normalized

random reservoir weight matrix $\mathbf{W}$, the ℓ_2 norm of $\mathbf{Wx}$ is approximately equal to the ℓ_2 norm of $\mathbf{x}$, when N is sufficiently large.

Let $\mathbf{W}_N = (w_{ij})_{N\times N}$, $\mathbf{W} = \alpha\mathbf{W}_N$, and $\mathbf{x} = [x_1, x_2, ..., x_N]^T$. Suppose $\mathbf{W}_N$ follows (4.4), (4.5), or (4.6), with α set to $\frac{1}{\sqrt{cN}}$ under (4.4) or $\frac{1}{\sqrt{N}}$ under (4.5) and (4.6). We have

$$\mathbf{Wx} = [\sum_j \alpha w_{1j}x_j, \sum_j \alpha w_{2j}x_j, \cdots, \sum_j \alpha w_{Nj}x_j]^T. \tag{4.13}$$

For the i^{th}-element, we have

$$E[\sum_j \alpha w_{ij}x_j] = \sum_j \alpha E[w_{ij}]x_j = 0, \tag{4.14}$$

$$\begin{aligned} Var[\sum_j \alpha w_{ij}x_j] &= E[\sum_j \sum_k \alpha^2 w_{ij}x_j w_{ik}x_k] \\ &= \frac{1}{N}\sum_j x_j^2, \end{aligned} \tag{4.15}$$

where $E[\cdot]$ denotes expectation and $Var[\cdot]$ denotes variance.

Thus, we have:

$$\begin{aligned} E[\|\mathbf{Wx}\|^2] &= E[\sum_i (\sum_j \alpha w_{ij}x_j)^2] \\ &= \sum_i \frac{1}{N}\sum_j x_j^2 = \|\mathbf{x}\|^2. \end{aligned} \tag{4.16}$$

where $\|\cdot\|$ denotes the ℓ_2 norm, *i.e.* the expected squared length of $\mathbf{Wx}$ is the same as the squared length of $\mathbf{x}$. Now we need to investigate how the distribution of $\|\mathbf{Wx}\|$ concentrates around $\|\mathbf{x}\|$. We first develop the following lemma.

Lemma 1. *Assume the random matrix* $\mathbf{W}$ *follows* (4.4), (4.5), *or* (4.6), *with the scaling factor set to* $\frac{1}{\sqrt{cN}}$ *or* $\frac{1}{\sqrt{N}}$, *as appropriate. Let* $\hat{\mathbf{x}} \in \mathbb{R}^N$ *be a unit vector; then,* $\|\mathbf{W}\hat{\mathbf{x}}\|$ *converges to 1 in probability, as* $N \to \infty$.

Proof. Lemma 4 and Lemma 5 in [114] state that for $\mathbf{W}$ as in (4.5) (Lemma 4) and for $\mathbf{W}$ as in (4.4) and (4.6) (Lemma 5), the following two inequalities hold for all N and for all $0 < \epsilon_2 < 1$:

$$\Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \geq (1+\epsilon_2)) < \exp(-\frac{N}{2}(\epsilon_2^2/2 - \epsilon_2^3/2)) \tag{4.17}$$

$$\Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \leq (1-\epsilon_2)) < \exp(-\frac{N}{2}(\epsilon_2^2/2 - \epsilon_2^3/2)) \tag{4.18}$$

for small positive constant $\epsilon_2 > 0$.

Then, for $0 < \epsilon_3 < 1$,

$$\begin{aligned}
&\Pr(|\|\mathbf{W}\hat{\mathbf{x}}\| - 1| \geq \epsilon_3) \\
=&\Pr(\|\mathbf{W}\hat{\mathbf{x}}\| \leq 1-\epsilon_3) + \Pr(\|\mathbf{W}\hat{\mathbf{x}}\| \geq 1+\epsilon_3) \\
=&\Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \leq (1-\epsilon_3)^2) + \Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \geq (1+\epsilon_3)^2) \\
<&\Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \leq 1-\epsilon_3) + \Pr(\|\mathbf{W}\hat{\mathbf{x}}\|^2 \geq 1+\epsilon_3) \\
<&2\exp(-\frac{N}{2}(\epsilon_3^2/2 - \epsilon_3^3/2))
\end{aligned} \tag{4.19}$$

Therefore, as $N \to +\infty$, $\Pr(|\|\mathbf{W}\hat{\mathbf{x}}\| - 1| \geq \epsilon_3) \to 0$. □

Given the Lemma, we can now state and prove our contraction mapping main result.

Theorem 8. *Assume the network defined in* (4.2) *and* (4.3) *with neuron activation function* $f = \tanh$*, applied component-wise. Suppose that* $\mathbf{x}(n), \tilde{\mathbf{x}}(n) \in [-1,1]^N$ *and* $\mathbf{W}$ *is a random reservoir weight matrix defined by* $\mathbf{W} = \alpha\mathbf{W}_N$*, according to* (4.4)*,* (4.5)*, or* (4.6)*, with* $\alpha = \rho/\sqrt{N}$ *under* (4.5)*,* (4.6)*, and* $\alpha = \rho/\sqrt{cN}$ *under* (4.4)*, where* $0 < \rho < 1$*. Then,*

$$\begin{aligned}
&\Pr(\|\mathbf{x}(n+1) - \tilde{\mathbf{x}}(n+1)\| \leq \|\mathbf{x}(n) - \tilde{\mathbf{x}}(n)\|) \\
>&1 - \exp(-\frac{N}{2}((1-\rho)^2/2 - (1-\rho)^3/2)),
\end{aligned} \tag{4.20}$$

where $\mathbf{x}(n+1) = T(\mathbf{x}(n), \mathbf{u}(n+1))$ *and* $\tilde{\mathbf{x}}(n+1) = T(\tilde{\mathbf{x}}(n), \mathbf{u}(n+1))$.

Proof. Let $\mathbf{z}(n) = \mathbf{x}(n) - \tilde{\mathbf{x}}(n)$. We start by writing:

$$\begin{aligned}
\|\mathbf{z}(n+1)\| &= \|T(\mathbf{x}(n), \mathbf{u}(n+1)) - T(\tilde{\mathbf{x}}(n), \mathbf{u}(n+1))\| \\
&= \|f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1)) \\
&\qquad - f(\mathbf{W}\tilde{\mathbf{x}}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1))\| \\
&\leq \|(\mathbf{W}\mathbf{x}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1)) \\
&\qquad - (\mathbf{W}\tilde{\mathbf{x}}(n) + \mathbf{W}_{\text{in}}\mathbf{u}(n+1))\| \\
&= \|\mathbf{W}\mathbf{x}(n) - \mathbf{W}\tilde{\mathbf{x}}(n)\| \\
&= \|\mathbf{W}(\mathbf{x}(n) - \tilde{\mathbf{x}}(n))\| \\
&= \|\mathbf{W}\mathbf{z}(n)\| \qquad (4.21)
\end{aligned}$$

where the inequality four lines above follows because the $\tanh(\cdot)$ function satisfies the (element-wise) Lipschitz condition $|\tanh(v) - \tanh(z)| \leq |v - z|$, $\forall v, z \in \mathbb{R}$. Let $\hat{\mathbf{z}}(n) = \mathbf{z}(n)/\|\mathbf{z}(n)\|$. Then rewrite (4.21) as

$$\|\mathbf{z}(n+1)\| \leq \|\mathbf{W}\mathbf{z}(n)\| = \|\mathbf{W}\hat{\mathbf{z}}(n)\| \cdot \|\mathbf{z}(n)\|.$$

We have $\mathbf{W} = \alpha\mathbf{W}_N$, and $\mathbf{W}_N$ generated according to (4.4), (4.5), or (4.6), with α equaling $\frac{\rho}{\sqrt{cN}}$, $\frac{\rho}{\sqrt{N}}$, or $\frac{\rho}{\sqrt{N}}$, respectively. From the circular law and Theorem 4, we know that the spectral radius of $\mathbf{W}$ converges to ρ as $N \to \infty$.

Applying Lemma 1, we thus have

$$\begin{aligned}
\|\mathbf{z}(n+1)\| &\leq \|\mathbf{W}\hat{\mathbf{z}}(n)\| \cdot \|\mathbf{z}(n)\| \\
&= \rho\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\| \cdot \|\mathbf{z}(n)\| \\
&\xrightarrow{p} \rho\|\mathbf{z}(n)\| \qquad \text{as } N \to \infty.
\end{aligned}$$

Further, let us characterize the probability that $\|\mathbf{z}(n+1)\| \geq \|\mathbf{z}(n)\|$, *i.e.* that the contractive

property is *not* satisfied, when N is finite. First, define $\epsilon = 1 - \rho$. Then, we have

$$\begin{aligned}
&\Pr(\|\mathbf{z}(n+1)\| \geq \|\mathbf{z}(n)\|) \\
\leq &\Pr(\|\mathbf{W}\mathbf{z}(n)\| \geq \|\mathbf{z}(n)\|) \\
= &\Pr(\|\mathbf{W}\hat{\mathbf{z}}(n)\| \geq 1) \\
= &\Pr(\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\| \geq \frac{1}{\rho}) \\
= &\Pr(\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\| \geq \frac{1}{1-\epsilon}) \\
< &\Pr(\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\| \geq 1+\epsilon) \quad (\because 1+\epsilon < \frac{1}{1-\epsilon}) \\
= &\Pr(\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\|^2 \geq (1+\epsilon)^2) \\
\leq &\Pr(\|\frac{1}{\rho}\mathbf{W}\hat{\mathbf{z}}(n)\|^2 \geq 1+\epsilon) \\
< &\exp(-\frac{N}{2}((1-\rho)^2/2 - (1-\rho)^3/2))
\end{aligned}$$

where the first inequality above follows from (4.21) and the final inequality follows from Lemmas 4 and 5 from [114], specified earlier.

We thus see that the probability that $\|\mathbf{z}(n+1)\| > \|\mathbf{z}(n)\|$ is exponentially decreasing with N. Moreover, $\|\mathbf{z}(n+1)\| \leq \|\mathbf{z}(n)\|$ with probability $1 - O(e^{-C_\rho N})$, where $C_\rho = \frac{1}{2}((1-\rho)^2/2 - (1-\rho)^3/2)$.

□

Theorem 7 shows that when $\rho < 1$, for $\mathbf{x}(n), \tilde{\mathbf{x}}(n) \in [-1, 1]^N$ and a random reservoir weight matrix $\mathbf{W}$, $T(\cdot, \cdot)$ is contractive with probability $1 - O(e^{-C_\rho N})$. This result supports and provides theoretical grounding for previous observations in echo state network research: "extensive experience with this scaling game indicates that one obtains echo states when α is only marginally smaller than $\alpha_{\max}$" [88] ($\alpha_{\max} = |\lambda_{\max}^{-1}(\mathbf{W}_N)|$). To give a caveat on this result, we also note that while we have shown that there is a contractive property with

high probability for large N, Theorem 7 is not definitive on whether the *strict* ESP given in Definition 1 holds with high probability for large N. This remains an open question.

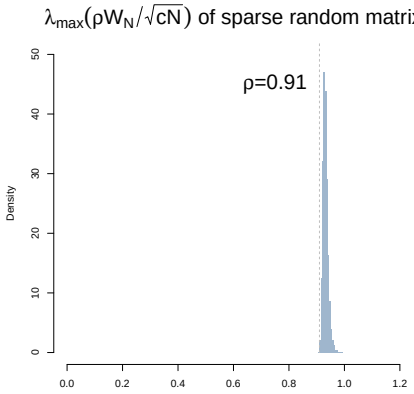
Finally, let us address the reader's possible concern that, in Theorem 7, we have assumed an unorthodox way of selecting the scaling factor on the weight matrix, in order to achieve our proof result, *i.e.* the reader may think our result is not relevant to the more conventional matrix scaling procedure. To address this concern, we next show that, from a practical standpoint, the result and insights obtained from Theorem 7 *also* apply if one considers the more conventional scheme for scaling $\mathbf{W}$. The logical argument goes as follows. There are two choices for the scaling factor – the conventional choice $\alpha = \rho|\lambda_{\max}^{-1}(\mathbf{W}_N)|$, and our unorthodox choice $\alpha = \rho/\sqrt{N}$ (for simplicity of discussion, we only consider the Gaussian and Bernoulli cases). Suppose that we could show that the spectral radius resulting from conventional scaling (which is the constant value, ρ) is *always* (for every realization of $\mathbf{W}_N$) less than or equal to the spectral radius resulting from our unorthodox scaling procedure. If this were true, one can see (from inspection of the proof of Theorem 7) that the Theorem 7 statement would directly apply not *only* for our unorthodox scaling procedure, but also for the more conventional scaling scheme. Likewise, if ρ is larger than the unorthodox scheme's spectral radius with vanishing probability as N gets large, we could say that Theorem 7 "practically applies" to conventional scaling, for large N. Let us consider two cases: i) asymptotically large N; ii) relatively large, finite (but increasing) N. For the asymptotic case, we simply note that, from the proof steps of Theorem 6, we know that the spectral radius obtained using these two different scaling methods converges to the *same* value (ρ) as $N \to \infty$. Thus, Theorem 7 is certainly relevant to the conventional scaling procedure in the limit of large N. Second, let us consider the case of finite (but increasing) N. There are two choices for the scaling factor – the conventional choice $\alpha = \rho|\lambda_{\max}^{-1}(\mathbf{W}_N)|$, and our unorthodox choice $\alpha = \rho/\sqrt{N}$. Now, it is not in fact true that ρ is *strictly* less (for all realizations $\mathbf{W}_N$) than the spectral radius obtained based on our unorthodox scaling. However, empirically, we will

next demonstrate the following results: 1) for large but finite N, the frequency with which conventional scaling leads to a larger spectral radius than unorthodox scaling is quite small; moreover, the "spread" of the unorthodox scaling's spectral radius distribution (around ρ) is small; 2) This frequency is observed to *decrease* with increasing N.

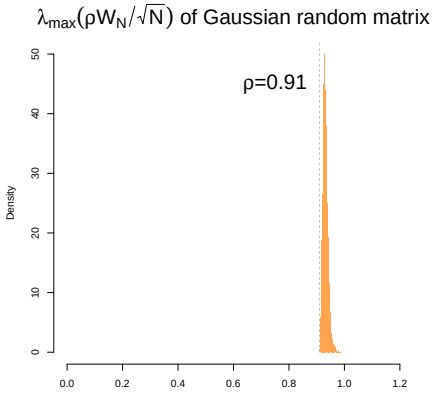
We simulated 10,000 trials for each of the three types of reservoirs, for $N = 500$, 1000, and 1500. We set $\rho = 0.91$ and observed that, using unorthodox scaling, for $N = 1000$ and $N = 1500$, the necessary ESP condition was met in every trial (with a small number of violations for $N = 500$). Our results, shown in Table 4.1, demonstrate that, very infrequently, ρ is greater than the spectral radius of the unconventional procedure. Furthermore, this frequency decreases for increasing N. Figure 4.4 shows the distribution of the unconventional procedure's spectral radius which, though skewed, is seen to have small spread about ρ. These experimental results suggest that Theorem 7 "practically applies" to conventional scaling as N gets large. The results also further corroborate our previous observation that for finite N, the mean of the spectral radius seems to converge from above to 1.

4.5 A Simulation Experiment

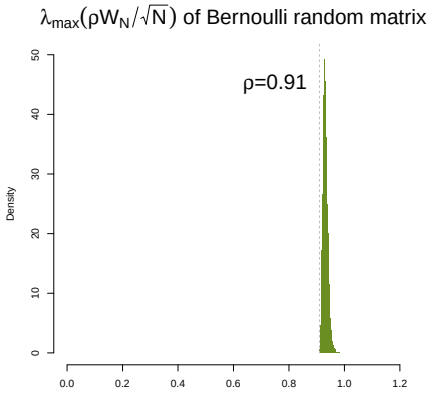
To demonstrate our results numerically, we applied ESNs to model the 10-th order NARMA as described in [115]. The random reservoir matrices were generated according to (4.4), (4.5), and (4.6), with the size $N = 300$. We used the previous sufficient condition ($\sigma_{\max}(\mathbf{W}) < 1$) and the new necessary and sufficient condition ($|\lambda_{\max}(\mathbf{W})| < 1$) for the echo state property to rescale $\mathbf{W}$. We repeated each experiment 20 times to calculate the mean and standard deviation of the normalized mean squared errors (NMSE) as shown in Table 4.2, Table 4.3, and Table 4.4. From the results, we have the following three observations: (1) the ratio $r = \frac{\sigma_{\max}(\mathbf{W})}{|\lambda_{\max}(\mathbf{W})|}$ is about 2 (Theorem 4); (2) ESNs with random reservoir matrices generated according to (4.4), (4.5), and (4.6) indeed behaved similarly; and (3) in all the experiments



(a) Sparse random matrices



(b) Gaussian random matrices



(c) Bernoulli random matrices

Figure 4.4: The histograms of the spectral radius of random matrices using the scaling factor in Theorem 7 with $\rho = 0.91$ and $N = 1000$.

Table 4.1: Simulation results on the spectral radius ($\hat{\rho}$) of random matrices using the scaling factor in Theorem 7 with $\rho = 0.91$ (10,000 trials).

	Sparse random reservoir		
N	$\Pr(\hat{\rho} \geq 1)$	$\Pr(\hat{\rho} < \rho)$	mean($\hat{\rho}$) (std)
500	0.09%	0.38%	0.938 ($\pm$0.014)
1000	0.00%	0.07%	0.932 ($\pm$0.009)
1500	0.00%	0.00%	0.929 ($\pm$0.007)
	Gaussian random reservoir		
N	$\Pr(\hat{\rho} \geq 1)$	$\Pr(\hat{\rho} < \rho)$	mean($\hat{\rho}$) (std)
500	0.13%	0.46%	0.938 ($\pm$0.014)
1000	0.00%	0.03%	0.932 ($\pm$0.009)
1500	0.00%	0.01%	0.928 ($\pm$0.007)
	Bernoulli random reservoir		
N	$\Pr(\hat{\rho} \geq 1)$	$\Pr(\hat{\rho} < \rho)$	mean($\hat{\rho}$) (std)
500	0.08%	0.29%	0.938 ($\pm$0.013)
1000	0.00%	0.01%	0.932 ($\pm$0.009)
1500	0.00%	0.01%	0.928 ($\pm$0.007)

where the new necessary and sufficient condition was applied, the echo state property was obtained, and the ESNs performed better than where the previous sufficient was applied.

4.6 Gene Expression Time Course Data Modeling

To understand the mechanism that orchestrate the genes and proteins in cells remains the key issue of systems biology studies. To construct computer models that can mimic the

Table 4.2: Results on ESNs with sparse random reservoirs and the new necessary and sufficient condition.

	Sparse random reservoir		
	$\bar{\sigma}_{\max}$	$\overline{\lvert\lambda_{\max}\rvert}$	NMSE (std)
$\sigma_{\max}(\mathbf{W}) < 1$	0.99	0.5032	0.1537 ($\pm$0.0015)
$\lvert\lambda_{\max}(\mathbf{W})\rvert < 1$	1.9404	0.99	0.1463 ($\pm$0.0002)

Table 4.3: Results on ESNs with Gaussian random reservoirs and the new necessary and sufficient condition.

	Sparse random reservoir		
	$\bar{\sigma}_{\max}$	$\overline{\lvert\lambda_{\max}\rvert}$	NMSE (std)
$\sigma_{\max}(\mathbf{W}) < 1$	0.99	0.5178	0.1515 ($\pm$0.0012)
$\lvert\lambda_{\max}(\mathbf{W})\rvert < 1$	1.8892	0.99	0.1464 ($\pm$0.0003)

Table 4.4: Results on ESNs with Bernoulli random reservoirs and the new necessary and sufficient condition.

	Sparse random reservoir		
	$\bar{\sigma}_{\max}$	$\overline{\lvert\lambda_{\max}\rvert}$	NMSE (std)
$\sigma_{\max}(\mathbf{W}) < 1$	0.99	0.5184	0.1526 ($\pm$0.0015)
$\lvert\lambda_{\max}(\mathbf{W})\rvert < 1$	1.8828	0.99	0.1463 ($\pm$0.0002)

behavior of cellular networks is very meaningful and instrumental to the understanding of gene regulation mechanism, which may lead to insights into many diseases. Since cellular networks (e. g. transcriptional networks, protein networks, and signaling pathways) are highly nonlinear and dynamic in nature, echo state networks are very suitable for modeling cellular networks based on time-course data. In [116], an in silico genetic regulatory network

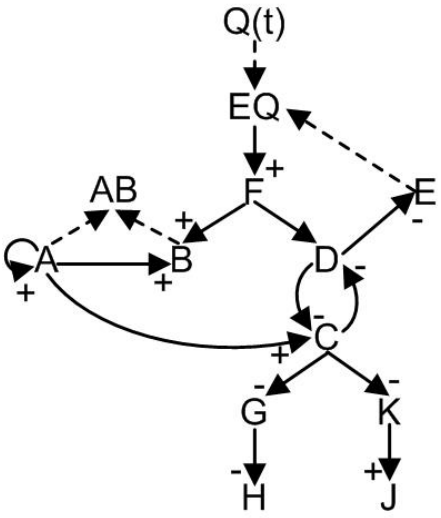


Figure 4.5: An *in silico* genetic regulatory network. Signed solid arrows indicate transcriptional regulatory interactions; dashed arrows indicate protein-protein interactions.

was constructed. The network structure is shown in Figure 4.5. The input to this system is the injection rate of the ligand Q, which triggers the response of gene expression in this network. In the simulation study, the input is pulse wave.

In this experiment, we want to model the gene expression time series of gene A, B, C, D, E, and F in response to the input ligand Q. We setup an ESN with 200 neurons in the reservoir with one input and six outputs. We test the learned model on a test data set (generated with different input sequence). The system output and ESN output are shown in Figure 4.6. We can see from Figure 4.6 that the dynamics of gene expression in this network in response to external inputs are well captured by the ESN model.

4.7 Conclusions

In this chapter, we applied random matrix theory to examine the properties of the random reservoirs used by ESNs, including different reservoir topologies (sparse or fully-connected)

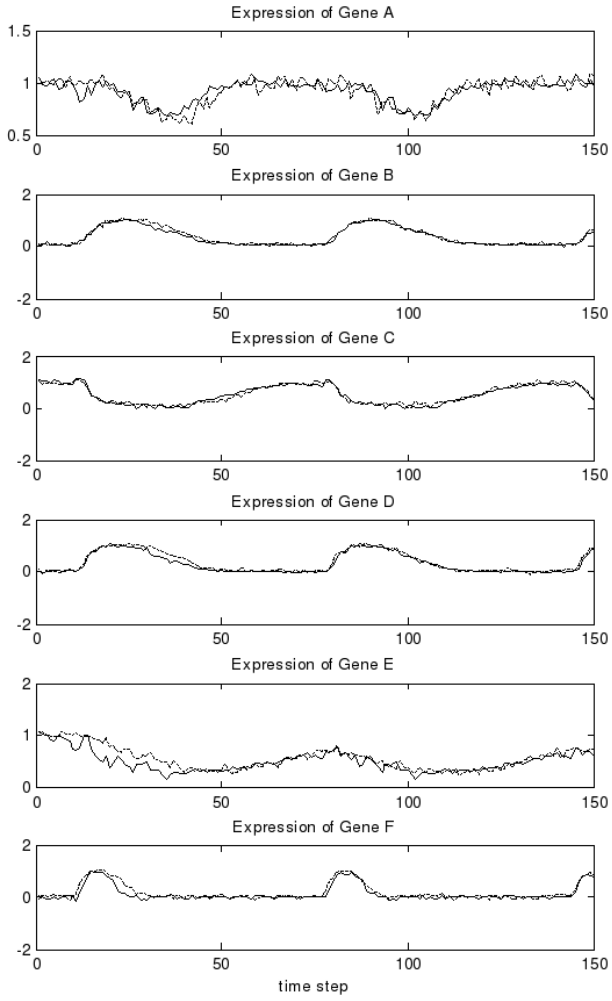


Figure 4.6: The expected output and the ESN output in gene expression time series modeling. The dotted line is the expected output and the solid line is the ESN output.

and different connection weights (Bernoulli or Gaussian). The asymptotic uniform distribution of the eigenvalues of the reservoir weight matrix ensures diverse dynamical patterns of the reservoir states. Moreover, this phenomenon does not depend on the topology of the reservoir or on the distribution of the weights of the connections. We showed that, asymptotic in the reservoir size, the bound for the necessary condition in [88] is about twice the bound for the sufficient condition in [88] for an ESN to possess the echo state property. Finally, we showed that when the the spectral radius $\rho < 1$, the state transition mapping $T(\cdot, \cdot)$ is contractive with high probability, which explains why the necessary condition has been found to be "sufficient in practice".

Chapter 5

Bayesian Analysis of Copy Number Mixtures to Correct Normal Cell Contamination and Characterize Tumor Evolution

5.1 Introduction

DNA copy number change is an important form of structural variation in the human genome. Somatic copy number alterations (CNAs) are key genetic events in the development and progression of human cancers, and frequently contribute to tumorigenesis [117]. The coverage of copy number changes varies from a few hundred to several million nucleotide bases, and somatic CNAs in tumors exhibit highly complex patterns. The advance of oligonucleotide-based single nucleotide polymorphism (SNP) arrays provides a high-density and allelic-specific genomic profile and enables researchers to study copy number changes on a genome-

wide scale. For instance, Affymetrix offers several DNA analysis arrays for SNP genotyping and copy number variation (CNV) analysis, and the newest Affymetrix Genome-Wide Human SNP Array 6.0 features 1.8 million genetic markers, including more than 906,600 SNPs and more than 946,000 probes for detecting CNVs or CNAs.

Quantitative analysis of somatic CNAs has found broad application in cancer research. Although molecular analysis of tumors in their native tissue environment provides the most accurate picture of their *in vivo* state, tissue samples often consist of mixed cancer and normal cells, and accordingly, the observed SNP intensity signals are the weighted sum of the copy numbers contributed from both cancer and normal cells. This tissue heterogeneity inherited in the measured copy number signals could significantly confound subsequent marker identification and molecular diagnosis rooted in cancer cells, *e.g.*, true copy number estimation, consensus region detection, CNA association studies, and detection of loss of heterozygosity and homozygous deletion. Experimental methods for minimizing normal cell contamination, such as cell enrichment or purification, are prohibitively expensive, inconvenient and prone to errors [1].

Here we ask whether it is possible to computationally correct normal tissue contamination by estimating the proportions of normal and cancer cells and recovering the true copy number profiles of cancer cells, based on the observed SNP intensity signals from cell mixtures. Albeit with limited success, some initial efforts have been recently made to address the impact of normal tissue contamination in copy number analysis [118–121], or to estimate the fraction of normal cells in tumor samples [122, 123]. Nancarrow, *et al.* [119] developed a visual inspection toolkit that allows users to determine the presence of stromal contamination. Yamamoto, *et al.* [122] and Goransson, *et al.* [123] proposed computational methods to estimate the proportion of normal cells by matching to the experimental or simulated histograms of different mixtures. However, given the fact that the noise level in the raw copy number data is often quite high and varies from sample to sample, neither visual inspection

nor simulated histogram matching will be able to produce an accurate and stable estimate of the fraction of normal cells in the tumor sample. An additional limitation associated with these methods is the lack of rigorous statistical principles in driving algorithm development.

In this study, we report a statistically-principled *in silico* approach to accurately detect genomic deletion type, estimate normal tissue contamination, and accordingly recover the true copy number profile in cancer cells. By exploiting the allele-specific information provided by SNP arrays, we introduce a series of definitions and theorems to illustrate the detectability and its conditions, and propose a Bayesian Analysis of COpy number Mixtures (BACOM) method. The BACOM algorithm is based on a statistical mixture model for copy number deletion segments in heterogeneous tumor samples, whose parameters are estimated using Bayesian differentiation between hemizygous deletion (hemi-deletion, where one allele is absent) and homozygous deletion (homo-deletion, where both alleles are absent) and plug-in sample averaging. Subsequently, the weighted average of estimated normal tissue fraction coefficients across multiple segments is used to estimate the true copy numbers rooted in cancer cells across all loci on the genome. As shown in the result section, this method not only produces cancer-specific copy number profiles but also substantially improves significant consensus events (SCEs) detection power.

To better serve the research community, we have developed a cross-platform Java application, which implements the whole pipeline of copy number analysis of heterogeneous cancer tissues. The BACOM software instantiates the algorithms described in this report and other necessary processing steps. To take advantage of many widely used packages in R to perform DNA copy number analysis and R's powerful and versatile visualization capabilities, we also provide an R interface, bacomR, that enables users to smoothly incorporate BACOM into their specific copy number analysis, or to integrate BACOM with other R or Bioconductor packages. We expect this newly developed software to be a useful tool in routine copy number analysis of heterogeneous tissues.

5.2 Problem Formulation

We first discuss a deletion-focused latent variable model for the copy number signal in heterogeneous tumor samples. Then, we propose a Bayesian approach to stochastically characterize distinctive copy number signals due to homo-deletion or hemi-deletion, supported by a novel summary statistic derived from allele-specific information. Next, we estimate the fraction of normal cells in the sample based on the deletion-type-specific segments, and subsequently recover the cancer-specific DNA copy number profile. Figure 5.1 gives the flowchart of BACOM consisting of three major steps: (1) inference of deletion types, (2) estimation of the normal tissue fraction, and (3) recovery of the copy number profile in cancer cells.

5.2.1 Copy Number Signal Model

Figure 5.2 shows SNP array intensity signals that serve as the raw data to study copy number changes, where observed non-integer copy numbers suggest the presence of normal cells in the tumor sample. In heterogeneous tumor samples, the measured array intensity is a mixture of DNA copy number signals from both normal and cancer cells, given mathematically by

$$X_i = \alpha \times X_{\text{normal},i} + (1-\alpha) \times X_{\text{cancer},i}. \tag{5.1}$$

where X_i is the observed DNA copy number signal at locus i, α is the unknown fraction of normal cell subpopulation in the sample, and $X_{\text{normal},i}$ and $X_{\text{cancer},i}$ are the unknown latent DNA copy number signals in normal and cancer cells at locus i, respectively. It should be noted that, in model (5.1), we have chosen not to consider CNVs in normal cells, because these are much rarer than CNAs in cancer cells.

Since human somatic cells are diploid, the expected DNA copy number at locus i in normal cells is two, *i.e.*, $E[X_{\text{normal},i}] = 2$. In contrast, if there is a homo-deletion or hemi-deletion at locus i in cancer cells, then the expected DNA copy number becomes zero or one, *i.e.*,

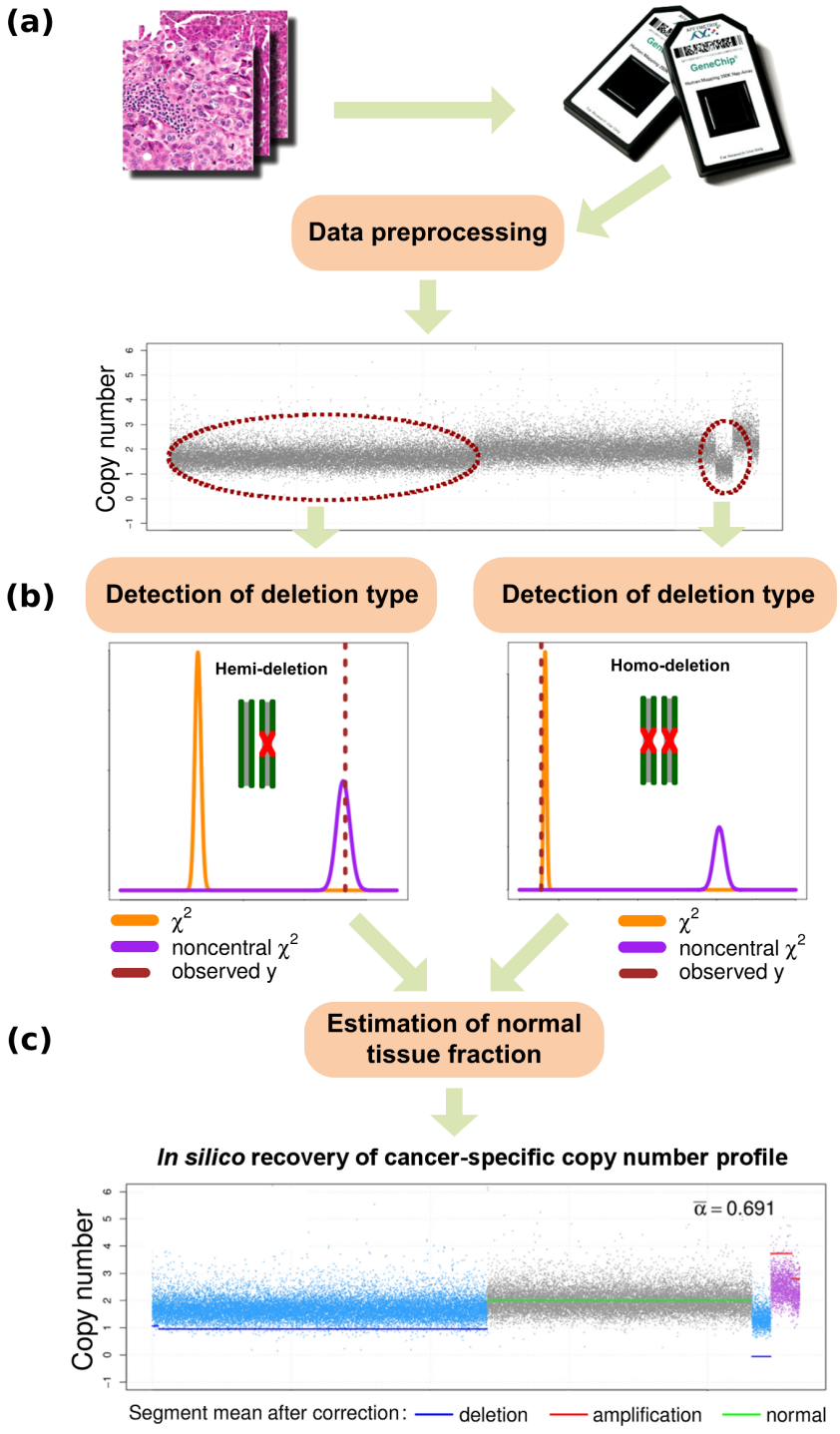


Figure 5.1: The flow chart of BACOM.

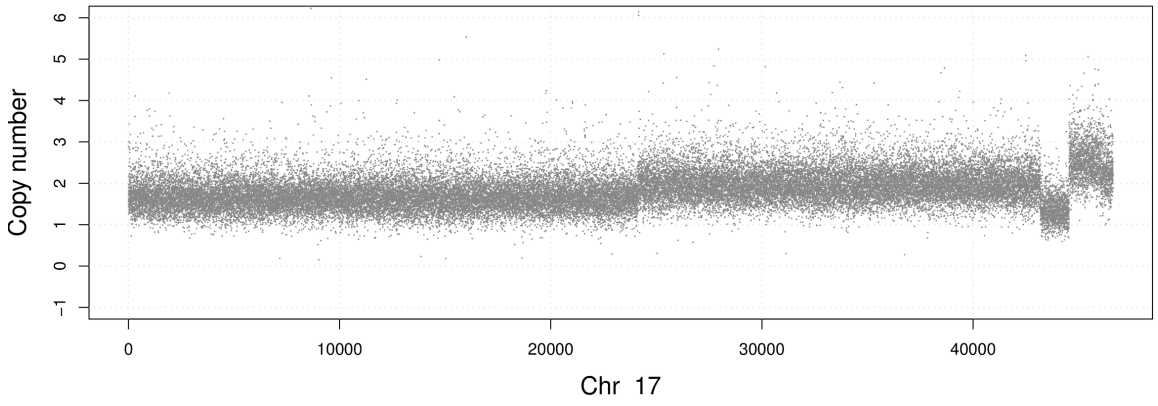


Figure 5.2: An illustration of a DNA copy number profile of Chromosome 17.

$E[X_{\text{cancer},i}] = 0$ or 1. By focusing on deletion-only CNA loci and taking the expectations on both sides of Equation (5.1), we have

$$\begin{cases} E[X_i] = \alpha \times 2 + (1-\alpha) \times 0 = 2\alpha, & \text{if homo-deletion,} \\ E[X_i] = \alpha \times 2 + (1-\alpha) \times 1 = 1 + \alpha, & \text{if hemi-deletion.} \end{cases} \tag{5.2}$$

Equation (5.2) indicates that, as a function of normal cell fraction α, the expected copy number at a deletion locus depends on the deletion type and is distinctive except when $\alpha = 1$. Inspired by this observation, we propose to explore a statistically-principled solution (detectability): if a Bayesian hypothesis test can be constructed to differentiate between homo-deletion and hemi-deletion segments based on allele-specific signals, we could, in principle, estimate α by the sample average over the deletion segments.

5.2.2 Inference of Deletion Type

Affymetrix SNP chips provide both allele-specific signals (A-allele and B-allele) and their summed intensity (observed DNA copy number signal). If we denote the signals of alleles A and B at locus i by $X_{A,i}$ and $X_{B,i}$, respectively, then, the observed DNA copy number signal

X_i in model (5.1) can be re-written as

$$X_i = X_{A,i} + X_{B,i}. \tag{5.3}$$

To fully exploit allele-specific information readily provided by the SNP arrays and associated genotype calling algorithms, our method will focus solely on AB genotype (not considering AA or BB genotypes). For a length-L homo/hemi-deletion segment $\{X_i | i = 1, 2, \cdots, L\}$, we make the following realistic assumption on the allele-specific signals.

Assumption 1. *For a length-L homo/hemi-deletion segment* $\{X_i | i = 1, 2, \cdots, L\}$*, each of the allele-specific signals* $X_{A,i}$ *and* X_{B_i} *are independently distributed Gaussian random variables with distinct means but common variance,* σ^2*, for* $i = 1, 2, \cdots, L$*.*

It should be noted that $X_{A,i}$ and $X_{B,i}$ are not statistically independent but, rather often correlated, referred to as the crosstalk between alleles A and B [124]). Thus, under Assumption 1, the observed copy number signals X_i are independent and identically distributed random variables following a normal distribution $N(\mu_{A+B}, \sigma^2_{A+B})$ whose mean μ_{A+B} and variance σ^2_{A+B} can be readily estimated by using the observed signals X_i for $i = 1, 2, \cdots, L$.

To statistically differentiate between hemi-deletion and homo-deletion, we define a novel summary statistic, given mathematically by the following newly defined random variable

$$Y = \sigma^{-2}_{A-B} \sum_{i=1}^{L} (X_{A,i} - X_{B,i})^2, \tag{5.4}$$

where σ^2_{A-B} is the variance of $X_{A,i} - X_{B,i}$. Under Assumption 1, it can be shown that Y follows either a noncentral or a standard χ^2 distribution, depending upon the deletion type. We therefore present the following two lemmas with proofs to show that the key parameter associated with these χ^2 distributions can be estimated using signals X_i, $X_{A,i}$ and $X_{B,i}$.

Lemma 2. *Suppose that, within a length-L hemi-deletion segment, each of the allele-specific signals* $X_{A,i}$ *and* $X_{B,i}$ *are independently distributed Gaussian random variables with distinct*

means and common variance. Then, the summary statistic random variable Y defined in (5.4) *follows an L degrees of freedom noncentral χ^2 distribution with non-centrality parameter $\lambda = L(2-\mu_{A+B})^2\sigma_{A+B}^{-2}(1+\rho)/(1-\rho)$, where ρ is the correlation coefficient between $X_{A,i}$ and $X_{B,i}$.*

Proof. Applying Equation (5.1) to the loci within a hemi-deletion segment, where one of the alleles (but not both) is deleted, we have, for $i = 1, 2, \cdots, L$

$$\begin{aligned}
\mu_{A-B} &= E[X_{A,i} - X_{B,i}] \\
&= E[\alpha \times (X_{\text{normal},A,i} - X_{\text{normal},B,i}) \\
&\qquad + (1-\alpha) \times (X_{\text{cancer},A,i} - X_{\text{cancer},B,i})] \\
&= \alpha \times E[X_{\text{normal},A,i} - X_{\text{normal},B,i}] \\
&\qquad + (1-\alpha) \times E[X_{\text{cancer},A,i} - X_{\text{cancer},B,i}] \\
&= \alpha \times (1-1) \pm (1-\alpha) \times (1-0) \\
&= \pm(1-\alpha), \quad i = 1, 2, \cdots, L.
\end{aligned}$$

While from Equation (5.2), we have $\mu_{A+B} = E[X_i] = 1 + \alpha$ which implies $\alpha = \mu_{A+B} - 1$. Thus, μ_{A-B} can be expressed in therms of μ_{A+B} as

$$\mu_{A-B} = \pm(1-\alpha) = \pm[1-(\mu_{A+B}-1)] = \pm(2-\mu_{A+B}).$$

Furthermore, Assumption 1 implies that

$$\sigma_{A+B}^2 = 2\sigma^2(1+\rho) \quad \text{and} \quad \sigma_{A-B}^2 = 2\sigma^2(1-\rho).$$

Although direct estimation of σ_{A-B}^2 is a nontrivial task, simple mathematical manipulation shows that σ_{A-B}^2 can be expressed in terms of σ_{A+B}^2 as

$$\sigma_{A-B}^2 = \sigma_{A+B}^2(1-\rho)/(1+\rho).$$

By the definition of the non-centrality parameter λ and Equation (5.4), we conclude

$$\begin{aligned}\lambda &= \sum_{i=1}^{L} (\frac{\mu_{A-B,i}}{\sigma_{A-B,i}})^2 \\ &= \sum_{i=1}^{L} \frac{[\pm(2-\mu_{A+B})]^2(1+\rho)}{\sigma_{A+B}(1-\rho)} \\ &= L(2-\mu_{A+B})^2\sigma_{A+B}^{-2}(1+\rho)/(1-\rho).\end{aligned}$$

Accordingly, the conditional L degrees of freedom noncentral χ^2 distribution of Y under hemi-deletion is given by

$$\chi^2(y; L, \lambda) = \begin{cases} \dfrac{e^{-(y+\lambda)/2}}{2^{L/2}} \displaystyle\sum_{k=0}^{\infty} \frac{y^{L/2+k-1}\lambda^k}{\Gamma(k+L/2)2^{2k}k!} & \text{for } y > 0, \\ 0 & \text{for } y \leq 0. \end{cases} \tag{5.5}$$

where Γ denotes the Gamma function.

Q.E.D. □

Lemma 3. *Suppose that, within a length-L homo-deletion segment, each of the allele-specific signals $X_{A,i}$ and $X_{B,i}$ are independently distributed Gaussian random variables with distinct means and common variance. Then, the summary statistic random variable Y defined in* (5.4) *follows an L degrees of freedom standard χ^2 distribution.*

Proof. Applying Equation (5.1) to the loci within a homo-deletion segment, where both

alleles are deleted, we have, for $i = 1, 2, \cdots, L$

$$\begin{aligned}
\mu_{A-B} &= E[X_{A,i} - X_{B,i}] \\
&= E[\alpha \times (X_{\text{normal},A,i} - X_{\text{normal},B,i}) \\
&\qquad + (1-\alpha) \times (X_{\text{cancer},A,i} - X_{\text{cancer},B,i})] \\
&= \alpha \times E[X_{\text{normal},A,i} - X_{\text{normal},B,i}] \\
&\qquad + (1-\alpha) \times E[X_{\text{cancer},A,i} - X_{\text{cancer},B,i}] \\
&= \alpha \times (1-1) + (1-\alpha) \times (0-0) \\
&= 0.
\end{aligned}$$

Thus, Equation (5.4) implies that, under homo-deletion, the summary statistic random variable Y defined in (5.4) follows an L degrees of freedom standard χ^2 distribution, given by

$$\chi^2(y; L) = \begin{cases} \dfrac{1}{2^{L/2}\Gamma(L/2)}\, y^{(L/2)-1} e^{-y/2} & \text{for } y > 0, \\ 0 & \text{for } y \leq 0, \end{cases} \tag{5.6}$$

where Γ denotes the Gamma function.

Q.E.D. □

Lemmas 2 and 3 suggest the possibility of constructing a Bayesian hypothesis testing strategy to differentiate between the two deletion-types (i.e., hemi-deletion and homo-deletion). The novel and powerful feature of this approach is that the parameter value of the underlying deletion-type-conditioned probability density function can be readily estimated using the available signals X_i, $X_{A,i}$ and $X_{B,i}$ without the knowledge of the deletion type associated with X_i, $X_{A,i}$ and $X_{B,i}$. Furthermore, having determined the deletion-type-conditioned probability density functions, we can then identify the deletion type of the segment using Bayesian hypothesis testing. The conclusion is summarized in the following theorem.

Theorem 9 (Deletion-type Identifiability)**.** *Suppose that, within a length-L deletion segment, each of the allele-specific signals $X_{A,i}$ and $X_{B,i}$ are independently distributed Gaussian random variables with distinct means and common variance. Then, the summary statistic random variable $Y = \sigma_{A-B}^{-2} \sum_{i=1}^{L} (X_{A,i} - X_{B,i})^2$ follows an L degrees of freedom χ^2 distribution under homo-deletion, and a noncentral L degrees of freedom χ^2 distribution under hemi-deletion, with a parameter that can be estimated based on signals $X_{A,i}$ and $X_{B,i}$. Accordingly, the segment deletion type can be optimally determined by Bayesian hypothesis testing.*

Proof. From Lemma 2, the summary statistic random variable Y under hemi-deletion follows an L degrees of freedom noncentral χ^2 distribution. From Lemma 3, the summary statistic random variable Y under homo-deletion follows an L degrees of freedom standard χ^2 distribution. Again, from Lemma 2, we have

$$\lambda = L(2 - \mu_{A+B})^2 \sigma_{A+B}^{-2} (1 + \rho)/(1 - \rho)$$

which can be estimated using readily-available signals.

Then, a straightforward application of Bayesian hypothesis testing implies that the deletion type of the segment can be optimally determined by

$$\begin{cases} \text{hemi-deletion}, & \text{if } P(\text{hemi-deletion}|y) \geq P(\text{homo-deletion}|y), \\ \text{homo-deletion}, & \text{if } P(\text{hemi-deletion}|y) < P(\text{homo-deletion}|y), \end{cases} \tag{5.7}$$

where $P(\cdot|\cdot)$ denotes the posterior probability of the segment deletion type given the observed segment signals.

Q.E.D. □

5.2.3 Implementation of BACOM Algorithm

We now complete the description of the BACOM algorithm by considering the estimation of the model parameters μ_{A+B}, σ_{A+B} and ρ. Note that μ_{A+B} and σ_{A+B} are segment-specific. For each segment, they can be readily estimated from the observed copy number signals by

$$\mu_{A+B} = \frac{1}{L}\sum_{i=1}^{L} X_i, \tag{5.8}$$

$$\sigma^2_{A+B} = \frac{1}{L-1}\sum_{i=1}^{L} (X_i - \mu_{A+B})^2. \tag{5.9}$$

Moreover, we assume that ρ is identical across all the loci within one subject profile, and hence we conveniently estimate its value based on only the signals at the N_{normal} loci within all normal segments, as given by

$$\mu_A = \sum_{i=1}^{N_{\text{normal}}} X_{A,i}, \qquad \mu_B = \sum_{i=1}^{N_{\text{normal}}} X_{B,i}, \tag{5.10}$$

$$\rho = \frac{\sum_{i=1}^{N_{\text{normal}}} (X_{A,i} - \mu_A)(X_{B,i} - \mu_B)}{\sqrt{\sum_{i=1}^{N_{\text{normal}}} (X_{A,i} - \mu_A)^2 \sum_{i=1}^{N_{\text{normal}}} (X_{B,i} - \mu_B)^2}}. \tag{5.11}$$

Having determined the parameters of the deletion-type conditional models we can infer the type of each deletion segment by applying Bayesian hypothesis testing based on (5.7). Subsequently, we can estimate the fraction of normal cells in the sample specified by (5.2), *i.e.*, $\alpha_j = \mu_j - 1$ for hemi-deletion and $\alpha_j = \mu_j/2$ for homo-deletion, where μ_j is the sample average of the copy number signals of the jth deletion segment. Moreover, assume that there are K deletion segments, we can calculate the ensemble estimate of the normal cell proportion via segment-length weighted average

$$\overline{\alpha} = \frac{\sum_{j=1}^{K} \alpha_j \times L_j}{\sum_{j=1}^{K} L_j}, \tag{5.12}$$

where L_j is the length of the jth deletion segment.

Finally, the estimated normal cell fraction can be used to recover the true copy numbers in cancer cells in the sample. Since $X_{\text{normal},i} = 2$ and based on (5.1), it is straightforward to estimate the DNA copy number of pure cancer cells by

$$\hat{X}_{\text{cancer},i} = \frac{X_i - 2\overline{\alpha}}{1 - \overline{\alpha}}. \tag{5.13}$$

5.2.4 Characterization of Tumor Evolution using BACOM

Normal tissue contamination correction using BACOM assumes that the tumor sample consists of normal cells and tumor cells and tumor cells are homogeneous. What will happen if the tumor cells are heterogeneous. Now let's consider the following illustrative example as shown in Figure 5.3. At time point 0, all the cells are normal, we observe the ideal copy number profile as illustrated in Figure 5.3(a). Then a genomic change takes place: one segment of the DNA has one allele deleted. The cells with the genomic change proliferate and become a tumor. At time point 1, we take a tumor sample, which consists of 50% normal cells and 50% tumor cells. The copy number profile of this tumor sample is illustrated in Figure 5.3(b). Later on, in some of the tumor cells, another genomic change takes place: another segment of the DNA has one allele deleted. This creates two subpopulations in the tumor cells: one subtype has one deletion, and the other subtype has two deletions. At time point 2, we take another tumor sample and we assume in the tumor sample, there are 50% normal cells, 25% tumor cells with one deletion, and 25%tumor cells with two deletions. The copy number profile is illustrated in Figure 5.3(c).

When we apply our BACOM method to the tumor sample at time point 2, we will have the estimated normal tissue fraction for the first deletion segment (segment in red in Figure 5.3(c)) $\hat{\alpha}_1 = 0.5$, and the estimated normal tissue fraction for the second deletion segment (segment in dark red in Figure 5.3(c)) $\hat{\alpha}_1 = 0.75$.

The discrepancy in the estimated normal tissue fractions actually indicates *the sequence of*

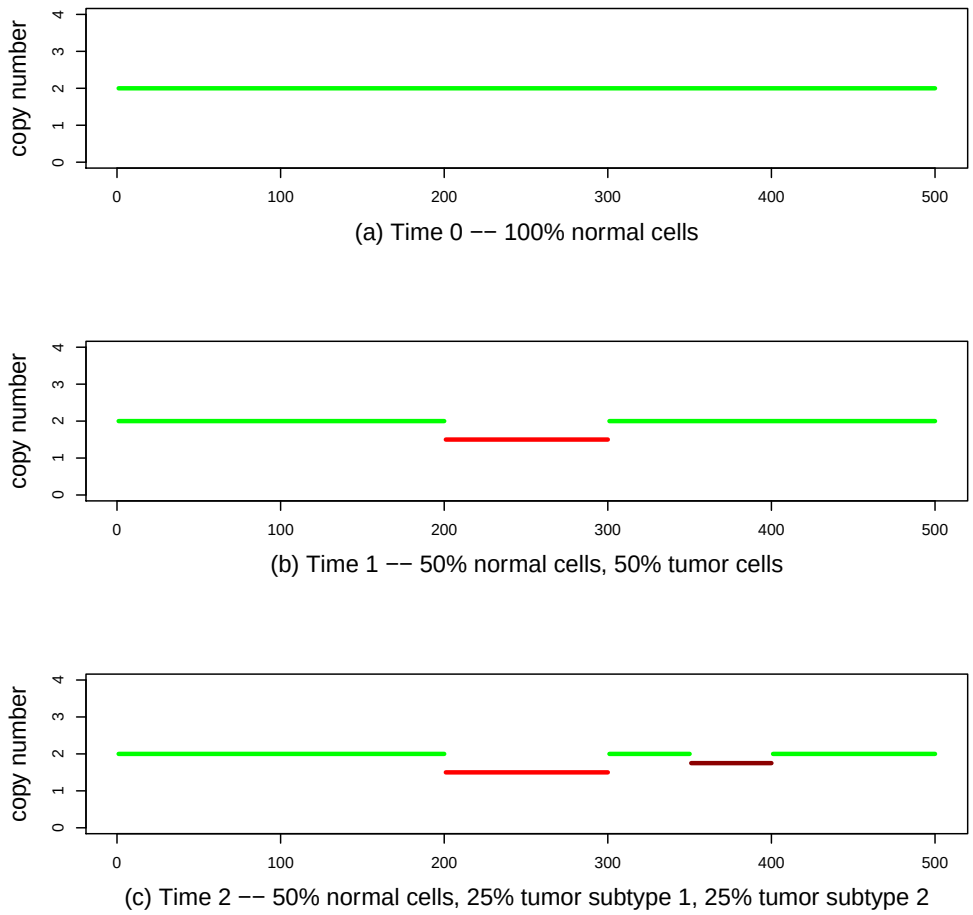


Figure 5.3: An illustration of the tumor evolution and the corresponding copy number profiles.

genomic change events. Therefore, in principle, we may infer the chronological order of genomic change events, utilizing the heterogeneity of tumor samples.

5.3 BACOM software

5.3.1 Standalone Java Application

To better serve the research community, we developed a cross-platform and open source BACOM Java application, which implements the entire pipeline of copy number change analysis for heterogeneous cancer tissues. The BACOM software instantiates not only the novel algorithms described here but also other relevant processing steps, including extraction of raw copy number signals from CEL files, iterative data normalization, identification of AB loci, copy number detection and segmentation, probe sets annotation, differentiation of deletion types, estimation of the normal tissue fraction, and correction of normal tissue contamination. Interested readers can download freely the software and source code at `http://www.cbil.ece.vt.edu/software.htm`. The screen shot of the BACOM Java application is shown in Figure 5.4.

5.3.2 Running BACOM in R Environment

To take advantage of many widely used packages in R and its associated powerful and versatile visualization capabilities, we also implemented an R interface, bacomR, that enables users to smoothly incorporate BACOM into their routine copy number analysis pipeline, or integrate BACOM with other R or Bioconductor packages. Users can use their preferred methods to perform routine tasks such as array normalization and DNA copy number segmentation and estimation, while using the newly added BACOM to estimate the normal cell fraction and subsequently recover the true copy number profiles in pure cancer cells.

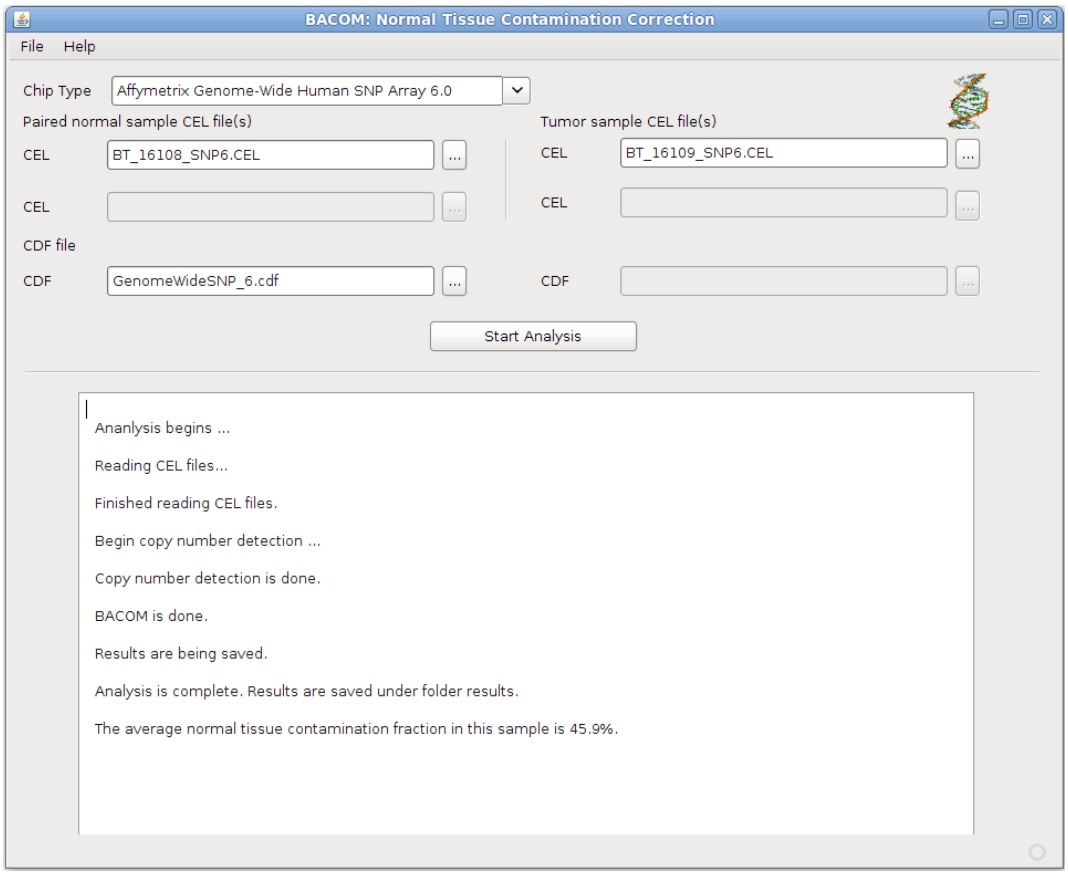


Figure 5.4: A screen shot of the BACOM software.

5.4 Results

5.4.1 Simulation Studies

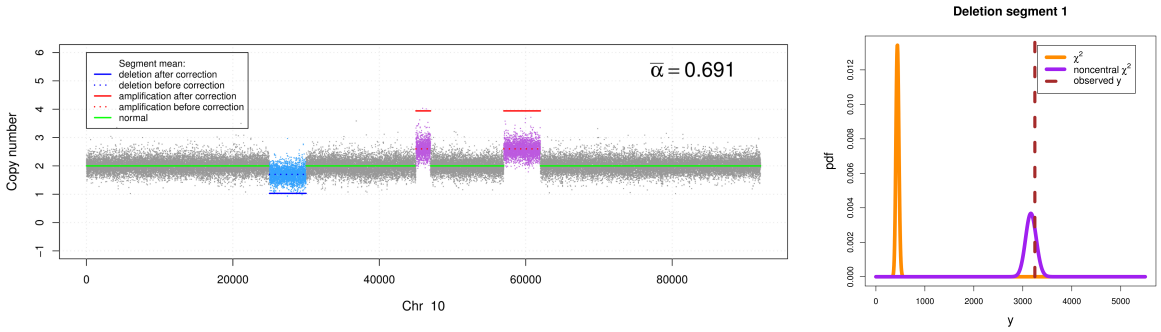
We first consider a realistic synthetic data set from a mixture of normal and simulated cancer copy number profiles, as shown in Figure 5.5a. The cancer copy number profile is simulated based on the real DNA copy number profile of a normal tissue sample assayed on the Affymetrix Genome-Wide 6.0 SNP array, consisting of two simulated 4-copy amplification segments and one simulated hemi-deletion segment. The normal and cancer copy number profiles are numerically mixed based on known proportions to produce the observed copy

number signal. Since there is only one deletion segment (loci 25k∼30k), it is theoretically impossible to tell the deletion type by examining the observed copy number signal, given the fact that the cancer copy number signal has been severely contaminated by a normal copy number signal. The single-deletion inclusion in this data set has been chosen in order to illustrate the unsupervised learning ability of BACOM in determining deletion types.

To determine the deletion type, we first estimate the posterior probability models of the summary statistic using allele-specific signals provided by SNP chips, and plot the observed value of the summary statistic associated with the deletion segment, shown in Figure 5.5b. The plot clearly suggests the hemi-deletion type of the deletion segment. We then estimate the normal tissue fraction in the sample based on the sample average of the deletion segment $\alpha = \mu_{A+B} - 1$. This leads to an estimate of $\alpha = 0.692$ and the accordingly corrected cancer copy number profile shown in Figure 5.5a. The results show the effectiveness of the BACOM approach in that the deletion type is correctly determined, the estimated normal tissue fraction is very close to the true value $\alpha = 0.7$, and the recovered amplification signals indicate the two expected 4-copy segments.

As an example of a more complex simulation, we consider a data set from a mixture of normal and simulated cancer copy number profiles, as shown in Figure 5.6a. The cancer copy number profile includes one homo-deletion, two hemi-deletions and three different amplification (copy numbers 3, 4 and 5) segments. The simulated cancer copy number signal, with a total of six altered copy number segments, not only retains the statistical characteristics of real SNP array intensity data but also provides a more complete picture of copy number alterations and genomic instability in cancer cells. Once again, the normal and cancer copy number profiles are numerically mixed based on known proportions to produce the observed copy number signal. The multiple-type-deletion inclusions in this data set have been chosen in order to illustrate the consistency and applicability of BACOM in estimating normal tissue fraction and cancer-associated copy number alterations.

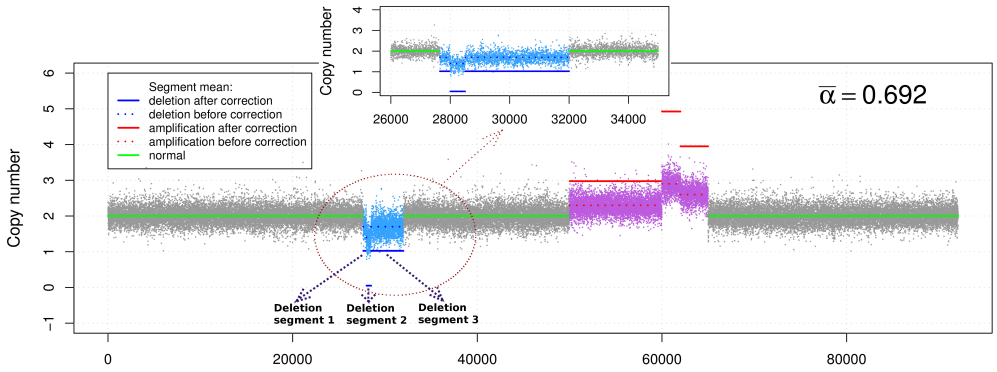
We first estimate separately the individual normal tissue fractions α_j from one homo-deletion and two hemi-deletion segments, where the posterior probability models and observed values of the summary statistic associated with the deletion segments are shown in Figure 5.6b. We then use the average value $\overline{\alpha}$ to recover the cancer-associated copy number profile, shown in Figure 5.6a, where the solid line segments are the recovered cancer-associated copy number changes. We tested BACOM on six simulation data sets with different α values, as given in Table 5.1. The BACOM approach again achieved very promising results in which the deletion types are correctly determined, the estimated normal tissue fractions from different deletion segments are highly consistent, with the average value very close to the true value, and the recovered signals of all six deletion and amplification segments indicate the expected integer-valued copy number changes. Table 5.1 summarizes the experimental results from all twelve simulation data sets.



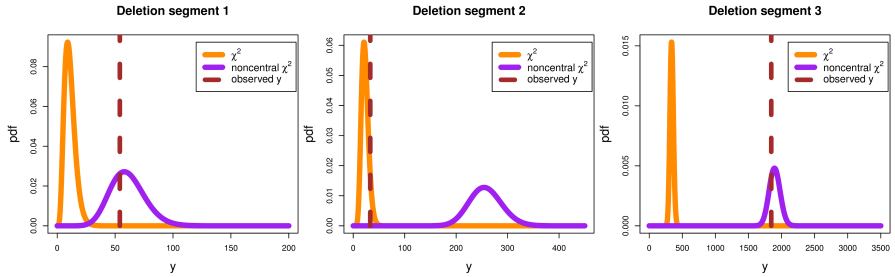
(a) Copy number profile and correction results.

(b) Bayesian analysis to determine the deletion type of the deletion segment.

Figure 5.5: The DNA copy number profile and Bayesian analysis of the deletion segment of the simulation dataset 1 when $\alpha = 0.7$.



(a) Copy number profile and correction results.



(b) Bayesian analysis to determine the deletion types of the three deletion segments.

Figure 5.6: The DNA copy number profile and Bayesian analysis of deletion segments of the simulation dataset 2 when $\alpha = 0.7$.

5.4.2 Analysis of Real DNA Copy Number Data

To test the applicability of our proposed method, we consider a real copy number profile for a prostate cancer sample assayed on the Affymetrix SNP 500K array. We first applied the BACOM algorithm to estimate the fraction of normal cell population in the sample, resulting in $\overline{\alpha} = 0.784$, which indicates significant normal tissue contamination. We then used the estimated $\overline{\alpha}$ value to recover cancer-specific copy number signal by Equation (5.13). The resulting corrected copy number profile for Chromosome 10 is shown in Figure 5.7, where dotted signals are the mixed copy number signals arising from the tumor sample with blue-colored regions being the detected deletion segments, green solid lines are the normal copy

Table 5.1: Estimation results on two simulation datasets.

	Dataset 1		Dataset 2			
α	$\overline{\alpha}$	α_1	$\overline{\alpha}$	α_1	α_1	α_3
0.291	0.291	0.3	0.293	0.285	0.286	0.293
0.391	0.391	0.4	0.393	0.385	0.386	0.393
0.491	0.491	0.5	0.493	0.485	0.486	0.493
0.591	0.591	0.6	0.592	0.585	0.585	0.593
0.693	0.693	0.7	0.692	0.691	0.685	0.685
0.793	0.793	0.8	0.792	0.785	0.785	0.793

number segments, and blue solid lines are the corrected cancer-specific deletion segments. In this experiment, our analysis readily reveals and distinguishes both deletion types and their occurred genomic locations. It is worth noting that BACOM algorithm identified a homo-deletion segment around locus 18,500 in Chromosome 10, that contains the well-known tumor suppressor gene *PTEN*.

As an example of a somewhat independent verification, we applied the BACOM algorithm to the copy number profile of another prostate cancer sample assayed on the Affymetrix Genome-Wide 6.0 platform [10]. The estimated fraction of normal cells in the sample is $\overline{\alpha} = 0.691$ and the results of similar analyses are given in Figure 5.8. Different from the previous example, this copy number profile contains two amplification segments that are purple-colored. Denoted by red solid lines, the corrected copy numbers of amplification segments are integer-valued, consistent with our theoretical expectation. This observation serves as a convincing validation of the proposed method, since the normal cell fraction $\overline{\alpha}$ was independently estimated from only deletion segments.

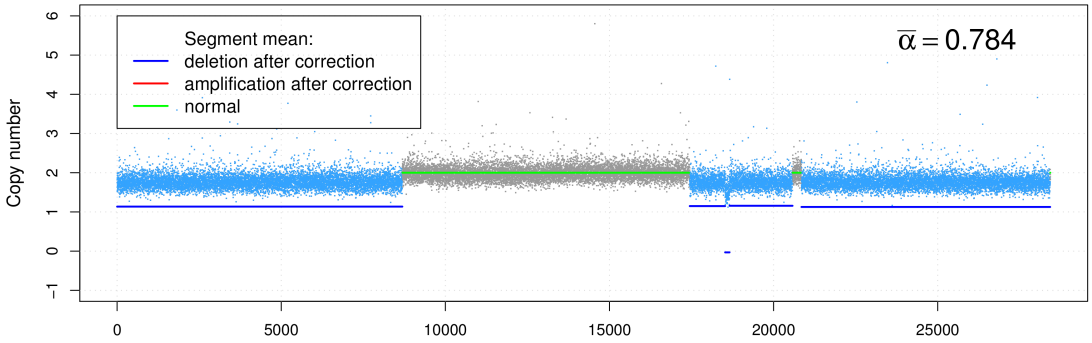


Figure 5.7: The DNA copy number profile of Chromosome 10 in a prostate cancer sample assayed on Affymetrix SNP 500K platform.

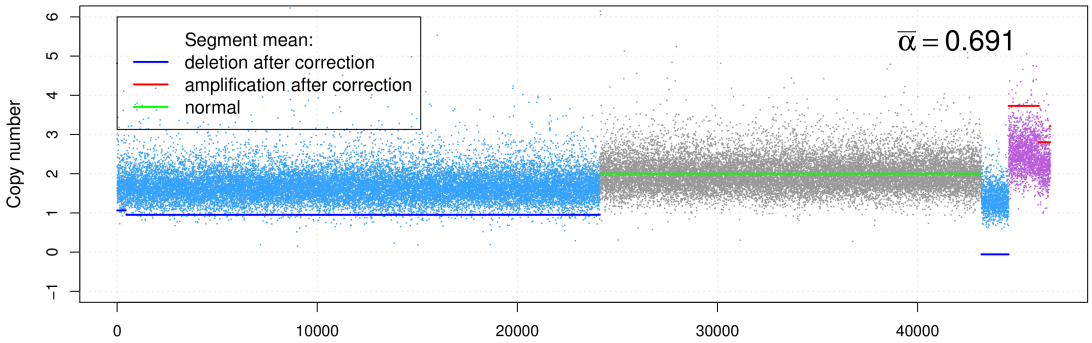


Figure 5.8: The DNA copy number profile of Chromosome 17 in a prostate cancer sample assayed on Affymetrix Genome-Wide 6.0.

5.4.3 A Case Study on a Prostate Cancer Data Set

We applied BACOM to a prostate cancer data set reported in [10]. Our study focused on the subjects that have genomic mutations in gene TP53. The DNA copy number profiles of Chromosome 17 associated with these samples are shown in Figures 5.9 $\sim$ 5.13. In these figures, dotted signals are the mixed copy number signals arising from tumor sample, where blue-colored regions are the detected deletion segments, purple-colored regions are the detected amplification segments, green solid lines are the normal copy number segments,

blue solid lines are the corrected cancer-specific deletion segments, and red solid lines are the corrected cancer-specific amplification segments. The vertical dashed orange line indicates the position of gene TP53 on the chromosome. Table 5.2 shows the estimated fraction $\overline{\alpha}$ of normal tissue contamination and the estimated copy number of gene TP53 before and after correction.

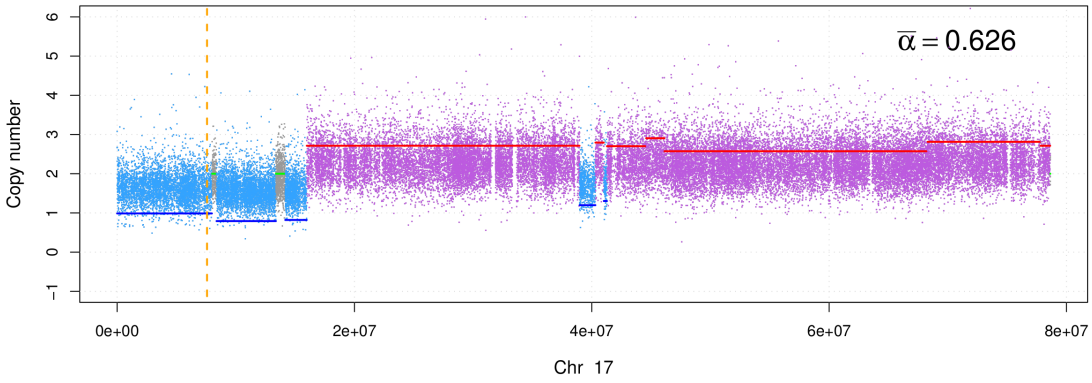


Figure 5.9: The DNA copy number profile of Chromosome 17 of Subject 3.

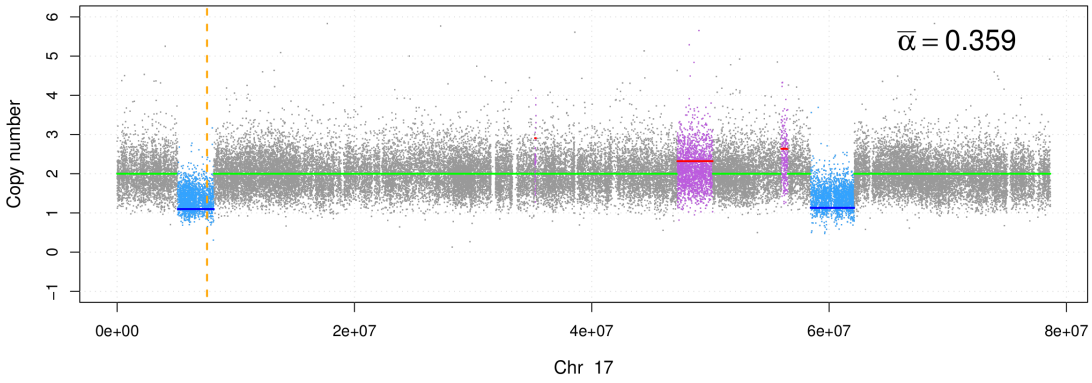


Figure 5.10: The DNA copy number profile of Chromosome 17 of Subject 16.

From Figures 5.9 $\sim$ 5.13 and Table 5.2, we can see that there is considerable variation in the normal cell contamination in these real prostate tumor samples (35.9% $\sim$ 69.1%). From

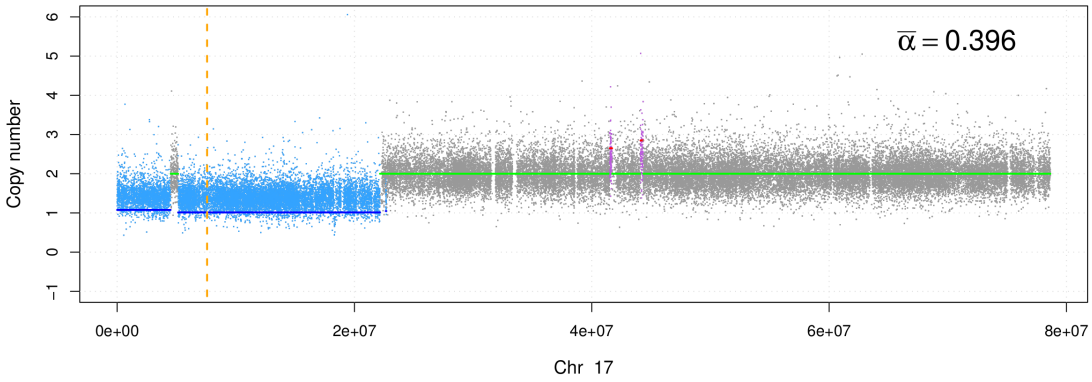


Figure 5.11: The DNA copy number profile of Chromosome 17 of Subject 24.

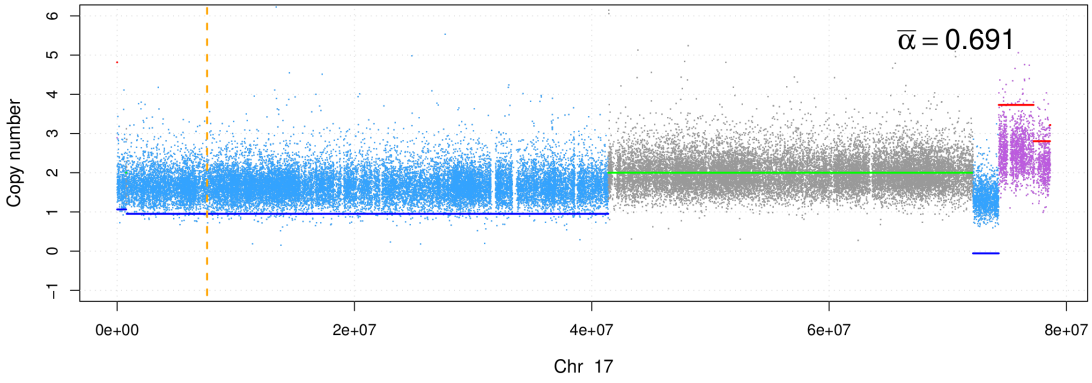


Figure 5.12: The DNA copy number profile of Chromosome 17 of Subject 30.

our simulation studies, such normal cell contaminations are expected to affect the accuracy of various follow-up analyses, such as the power to detect significant consensus regions. In Table 5.2, after correction, the copy numbers of TP53 gene are close to 1 in these samples, suggesting that hemizygous deletion has taken place in this region.

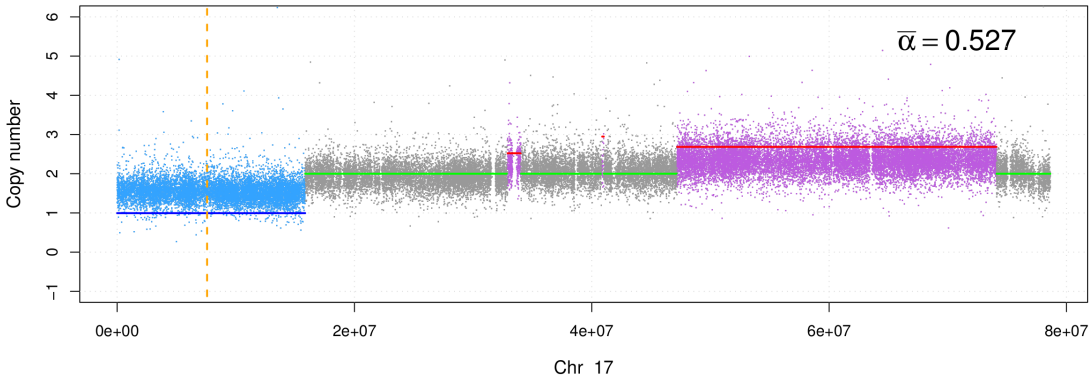


Figure 5.13: The DNA copy number profile of Chromosome 17 of Subject 32.

Table 5.2: TP53 copy number in a prostate cancer data set.

Subject	$\overline{\alpha}$	TP53 gene copy number	
		before correction	after correction
3	0.626	1.622	0.987
16	0.359	1.422	1.098
24	0.396	1.404	1.013
30	0.691	1.676	0.952
32	0.527	1.524	0.995

5.4.4 A Case Study on TCGA Ovarian Cancer Data Set

To further test the applicability of BACOM, we also applied BACOM software to The Cancer Genome Atlas (TCGA) ovarian cancer data set. Figures 5.14∼5.16 present some of the preliminary results showing that tumor suppressor genes TP53 and BRCA1 are jointly deleted in these ovarian cancer samples. In these figures, dotted signals are the mixed copy number signals arising from tumor samples, where blue-colored regions are the detected

deletion segments, purple-colored regions are the detected amplification segments, green solid lines are the normal copy number segments, blue solid lines are the corrected cancer-specific deletion segments, and red solid lines are the corrected cancer-specific amplification segments. The vertical dashed orange line indicates the position of gene TP53 on the chromosome, and the vertical dashed brown line indicates the position of gene BRCA1.

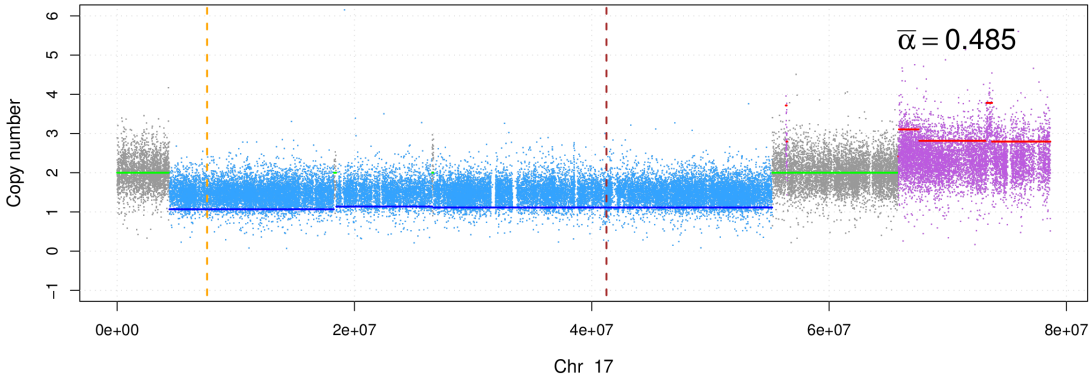


Figure 5.14: The DNA copy number profile of Chromosome 17 of Subject 1662.

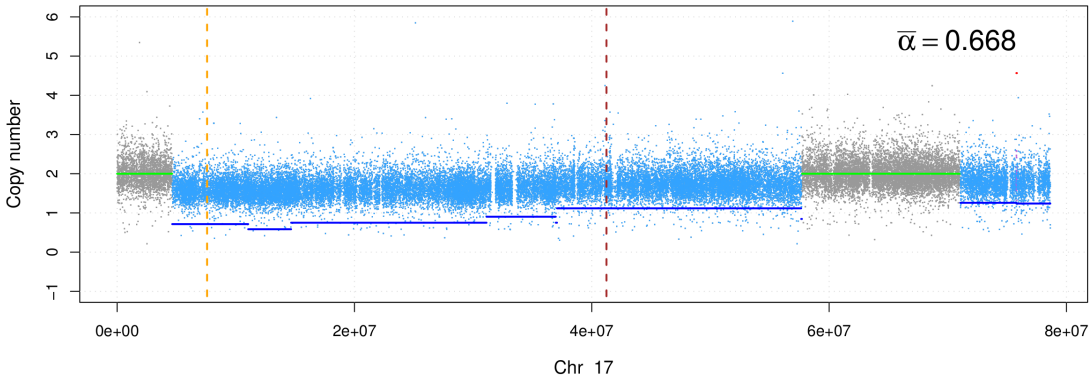


Figure 5.15: The DNA copy number profile of Chromosome 17 of Subject 1557.

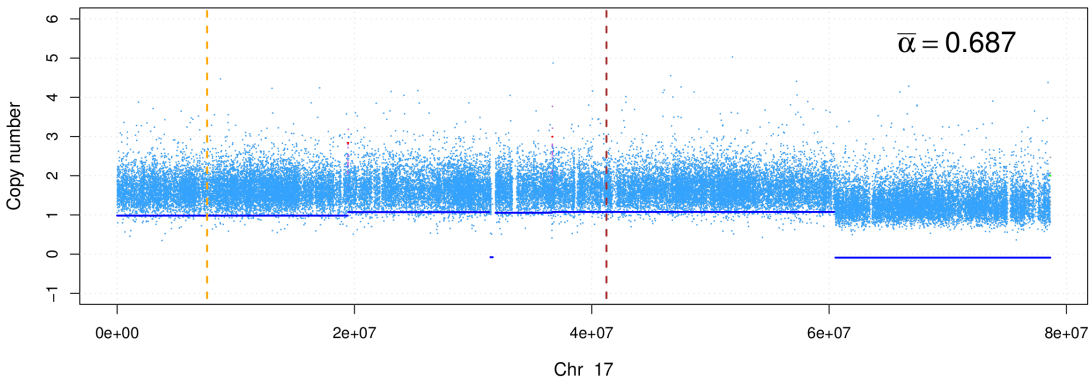


Figure 5.16: The DNA copy number profile of Chromosome 17 of Subject 1544.

5.4.5 Impact on Detecting Significant Consensus Events

Somatic copy number alterations in genomes underlie almost all human cancers. One of the systematic efforts to characterize cancer genomes is to identify significant consensus events (SCEs) from random background aberrations. To test the utility of our method to address an important biological question, we applied BACOM together with GISTIC (genomic identification of significant targets in cancer) [125] to specifically designed copy number simulation datasets. Each sample (3,000 loci) contains both normal copy number and various deletion/amplification segments (150∼250 loci). The consensus events at certain loci are inserted into the base profile according to a specified frequency, while random background aberrations are simulated with randomly assigned length and loci. Simulation parameters include sample size, consensus frequency, and normal cell fraction. We generated 1,000 simulation datasets for each combinatorial parameter setting, resulting in a total of 20,000 simulation datasets, each containing 30∼90 samples.

Next, for each of the mixed copy number profiles, we recovered cancer-specific copy numbers by BACOM. To detect SCEs from both mixed and deconvolved copy number profiles, we applied GISTIC, a statistical method that calculates a score that is based on both the ampli-

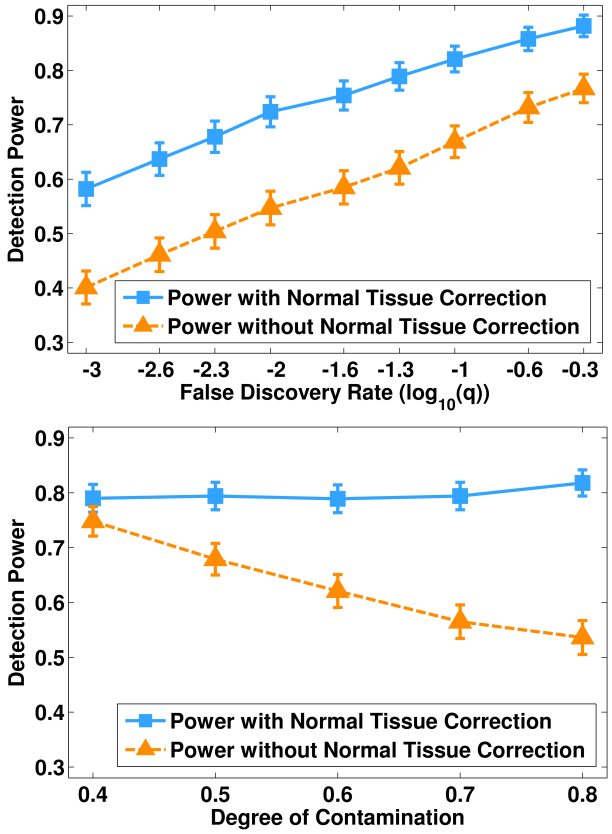


Figure 5.17: A comparison of the power to detect significant consensus events with- and without- correction of the normal tissue contamination, along different false discovery rates (FDR) and degree of contamination.

tude and frequency of copy number changes at each position, using a semi-exact approach to determine significance. To analyze the impact of correcting normal tissue contamination on detecting SCEs, we calculated power based on GISTIC outcomes and ground truth. Comparative experimental results, given in Figure 5.17, consistently show significantly improved power using deconvolved cancer-specific profiles.

5.4.6 Evidence of Tumor Evolution in Metastatic Prostate Cancer Copy Number Data

To search for the evidence of tumor evolution in metastatic prostate cancer copy number data using BACOM, we performed experiments on the prostate cancer copy number data described in [10]. In this dataset, there are six cancerous samples from Subject 33. We applied our BACOM algorithm to the copy number data. Figure 5.18 shows the histograms of the estimated normal tissue fractions using deletion segments. If the tumor cells are homogeneous, ideally, the histogram of the estimated normal tissue fractions should be unimodal (such as a Gaussian). However, we can clearly see multimodal distributions in some samples in Figure 5.18, which suggests the heterogeneity in the cancerous cells.

Now let's have a closer look at Sample 16010. As shown in the Figure 5.19, the estimated normal tissue fraction using the shaded region of the Chromosome 10 is 0.713, greater than the average estimated normal tissue fraction 0.577. We may hypothesize that the deletion of this segment took place rather late and only a fraction of the tumor cells went through such genomic change. To support this hypothesis, we examined the same region of the genome in other tumor samples (from the same Subject 33). As shown in Figure 5.19, other tumor samples did not show deletion in this segment, which may imply that at the time of metastasis, this genomic change had not occurred.

A more complicated story may be told about Chromosome 6 in Sample 16010, as shown in Figure 5.20. We shaded two small deletion segments in Chromosome 6 of Subject 33, which we will refer to as deletion segment 1 and deletion segment 2. These two segments show higher percentage of normal tissue in the sample, which are 0.789 and 0.686, respectively, larger than the average normal tissue fraction 0.577. This may imply that these two genomic changes took place later than other genomic changes on this chromosome. Samples 16029, 16030, 16031, 16035 and 16036 can be grouped into two categories: the first group is Samples

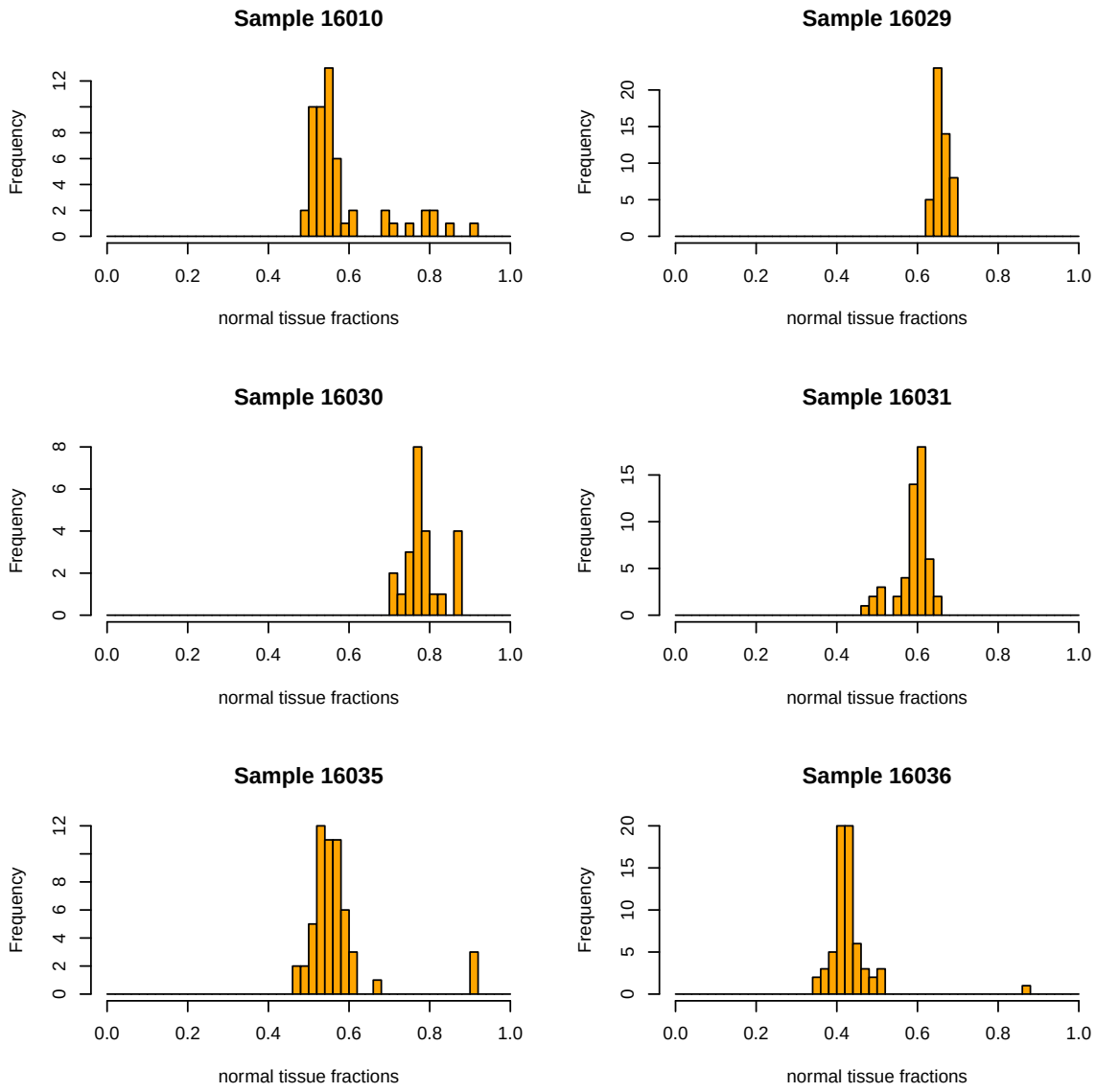


Figure 5.18: The histograms of the estimated normal tissue fractions (Samples 16010, 16029, 16030, 16031, 16035, 16036).

16029, 16031 and 16035, and the second group is Samples 16030 and 16036. Metastasis seems to have took place in the first group before the occurrence of deletion segment 2, while metastasis seems to have took place in the second group after both deletion segment 1 and deletion segment 2 had occurred.

5.5 Conclusions

In this chapter, we report a statistically-principled *in silico* approach to estimate copy number deletion types and normal tissue contamination, and to extract the true copy number profile in cancer cells. The BACOM algorithm utilizes the allele-specific information provided by SNP chips to differentiate between hemi-deletion and homo-deletion and subsequently estimates the fraction of normal cells in tissues. We tested the proposed method on twelve simulation datasets and two real datasets and obtained highly promising results. We expect the newly developed BACOM software to be a useful tool in copy number analysis of heterogeneous tissues.

There are some questions worth further exploration. Specifically, so far we have focused on normal tissue contamination by assuming a homogeneous cancer cell population, while in reality, cancer cells are often clonally heterogeneous leading to cancer subtypes. The ability to further dissect genomic heterogeneity of cancer cells is of great interest and will facilitate pathogenesis studies with far-reaching clinical implications.

In addition to heterogeneity of copy number, more mutations in cancer cells are expected and may have some unknown implications. However, since the summary statistic was defined on the whole deletion segment and the final normal tissue fraction was estimated using segment-length weighted average over multiple deletion segments, such mutations will only have negligible effects on the estimation accuracy as long as the mutations are sporadic compared with copy number alterations. In our experiments on real data sets, we have not observed any major effects caused by such mutations.

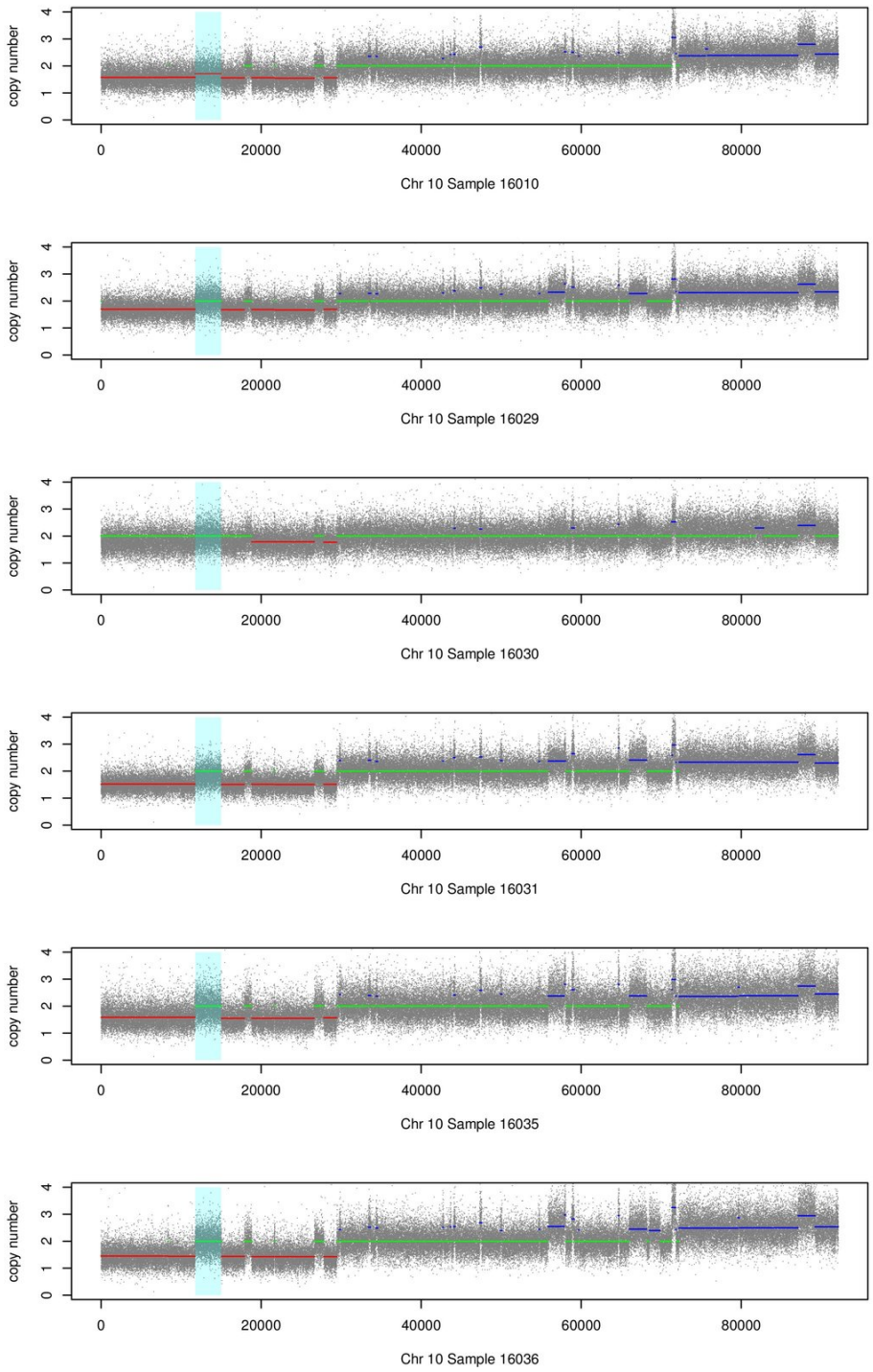


Figure 5.19: The copy number profiles of Chromosome 10 of Subject 33 (Samples 16010, 16029, 16030, 16031, 16035, 16036).

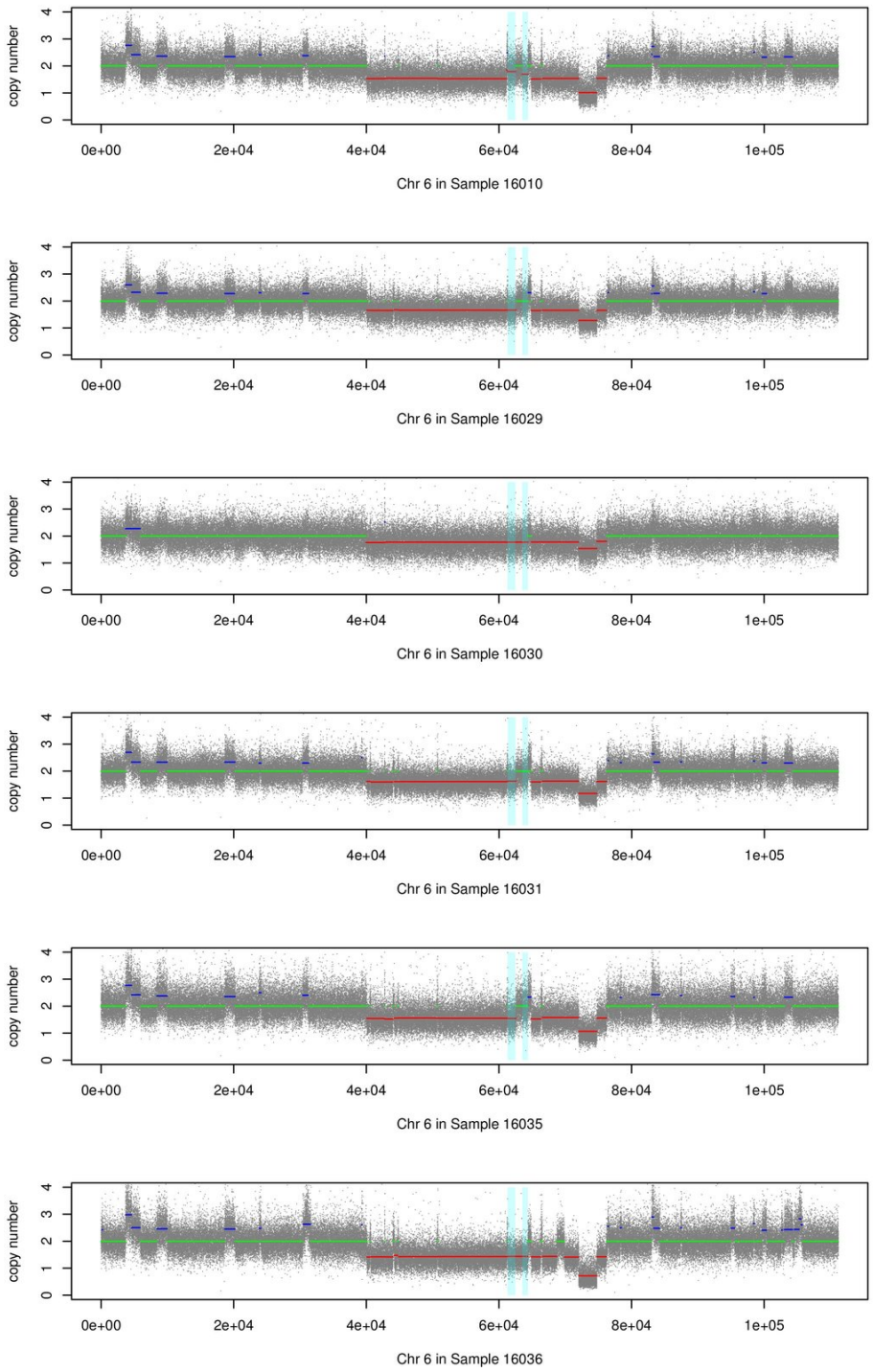


Figure 5.20: The copy number profiles of Chromosome 6 of Subject 33 (Samples 16010, 16029, 16030, 16031, 16035, 16036).

Chapter 6

Summary of Contributions and Future Work

6.1 Summary of Contributions

6.1.1 DDN Methodology and Its Applications in Condition-Specific Biological Networks

We propose a differential dependency network (DDN) analysis to detect statistically significant topological changes in the transcriptional networks between two biological conditions. The contributions of this work include:

- Differential dependency network (DDN) analysis was the first formal attempt to formulate the problem of detecting statistically significant network topological changes between two conditions. While most biological network modeling methods had been focused on constructing composite and static network models that could explain various regulation programs in the cells, and gene selection methods had been mainly look-

ing for differentially expressed genes, DDN analysis provides an alternative approach to identify key genes by emphasizing the dynamic nature of biological networks and utilizing the network structural information.

- DDN analysis decomposes the whole network into a set of local structures and utilizes Lasso method to learn their sparse structures; and subsequently, a permutation test scheme and a novel test statistic were proposed to detect statistically significant network topological changes.

- DDN has been successfully applied to several breast cancer, muscular dystrophy, and ovarian cancer studies, providing new biological insights and understandings.

- We implemented DDN algorithm as a standalone Java application, a Matlab package, an R package, and a Cytoscape plug-in, CytoDDN. DDN has been adopted by caBIG® (cancer Biomedical Informatics Grid) as an analytical tool for detecting and visualizing statistically significant topological changes in transcriptional networks representing two biological conditions.

6.1.2 A General Framework and an Efficient Algorithm of Learning Structural Changes in Graphical Models

We report an effective learning strategy to extract structural changes in Gaussian graphical models between two conditions. The contributions of this research work are threefold:

- We propose a novel formulation of graphical model structural change learning between two conditions as a convex optimization problem. This formulation has three useful properties which can be translated into efficient algorithms.

- We propose an efficient block coordinate descent algorithm to solve this problem. The

proposed algorithm for solving the optimization problem is several thousands times faster than the standard convex optimization as implemented in widely used packages such as CVXOPT.

- This framework is very flexible to incorporate biological prior knowledge. The prior knowledge incorporation scheme carefully evaluates and controls the impact of false positives in the prior knowledge on the network inference results, and automatically selects the "optimal" degree of information fusion between the evidence in knowledge and the evidence in the data.

6.1.3 Theoretical Analysis of Echo State Networks

We examine several important properties of the random reservoirs of echo state networks using random matrix theory and apply echo state networks to model gene expression time-course data. The contributions of this work include:

- We apply recent results from random matrix theory to demonstrate the asymptotic distributions of eigenvalues and singular values of reservoir weight matrices. We then show that randomly generated reservoirs, either sparsely or fully connected, either with Bernoulli or Gaussian connection weights (or, in fact, with weights distributed according to other density families), are all expected to behave similarly.

- We quantify the gap between the scaling factor bounds used to define the echo state property necessary and sufficient conditions proposed in previous works. We show that, asymptotic in the size of the reservoir, this gap becomes quite large, with the necessary condition bound twice as large as the sufficient condition bound.

- Finally, we show that when the spectral radius of the reservoir weight matrix is smaller than 1 (the *necessary* condition for the echo state property when the input space

contains the zero sequence), the state transition mapping is in fact contractive with high probability, given a sufficiently large reservoir. This result corroborates the observation in [88] that the necessary condition for the echo state property is often good enough in practice, such that violations of the ESP are not practically observed. This result, together with the factor of two asymptotic gap between the scaling factor bounds, indicates the conservativeness of the sufficient conditions from [88] and [100]. The practical implication of these results is that standard ESN design approaches, based on use of the sufficient conditions, are suboptimal – use of a conservative scaling factor compromises the amount of memory in the RNN, and thus the ability to accurately model a given target dynamical system.

6.1.4 BACOM Methodology and Its Applications

We propose a statistically-principled *in silico* approach, Bayesian Analysis of COpy number Mixtures, to accurately detect genomic deletion type, estimate normal tissue contamination, and accordingly recover the true copy number profile in cancer cells. The specific contributions of this work include:

- We design an innovative summary statistic, which utilizes the allelic-specific information in SNP arrays and also can be characterized by χ^2 and noncentral χ^2 distributions, to differentiate deletion types. On the contrary, traditional analysis of copy number data usually uses only the copy number amplitude information, but overlooks the allelic-specific information provided by Affymetrix SNP arrays.

- Based on these deletion segments, we estimate normal tissue fraction contamination and recover the true copy number profile in cancer cells. Correcting normal tissue contamination increases the power of follow-up copy number analysis such as consensus region detection.

- BACOM algorithm has been successfully applied to several prostate cancer and ovarian cancer data sets.
- We develop an open-source, cross-platform standalone Java application, which implements the whole pipeline of copy number analysis of heterogeneous cancer tissues, including extraction of raw copy number signals from CEL files, iterative data normalization, identification of AB loci, copy number detection and segmentation, probe sets annotation, differentiation of deletion types, estimation of the normal tissue fraction, and correction of normal tissue contamination.

6.2 Future Work

6.2.1 Condition-Specific Network Learning from Heterogeneous Biological Data

It is of great scientific significance to model condition-specific biological networks using data from multiple and heterogeneous data sources, including mRNA data, DNA copy number data, DNA methylation data, ChIP-chip data, protein-protein interaction data, and biological pathway databases. The differential dependency network analysis and its extension in Chapters 2 and 3 have demonstrated the possibility and effectiveness that integrate mRNA data and biological prior knowledge to learn condition-specific biological networks. There are several directions worth further explorations.

1. Integrative learning from multiple data types and bridging the gap between the genomic variations and biological network rewiring.
2. Inclusion of discreet variables in the network modeling, in addition to continuous variables.

3. Extension of this framework to multiple conditions and time-course data.

6.2.2 Intra-tumor Heterogeneity and Tumor Evolution

As discussed in Chapter 5, BACOM can also be applied to investigate tumor evolution in cancer cells and some evidence in the data partially supports our hypothesis. Here are some further thoughts along this direction.

1. Current BACOM extension for tumor evolution has the (implicit) assumption that there are two cancer subpopulations and they have multiple distinct genomic deletion segments in order to differentiate them (in this case, the distribution of α will have two peaks). Under this scenario, we can devise a model selection algorithm (*e.g.* Gaussian mixture model to fit the α distribution and select the number of mixtures) to determine whether a given tumor sample has intra-tumor heterogeneity. It will be of great interest to relax this assumption and be able to deal with multiple cancer subpopulations in the sample.

2. The next-generation sequencing technology has great potential applications in studying intra-tumor heterogeneity and tumor evolution. A recent progress [126] along this direction is refreshing and stimulates us to think beyond current strategies to characterize tumor evolution, in addition to DNA the copy number data based methods. Joint analysis of both SNP arrays and sequence data provides richer information about the tumor evolution history.

Bibliography

[1] R. Clarke, H. W. Ressom, A. T. Wang, J. H. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer*, vol. 8, no. 1, pp. 37–49, 2008.

[2] A. Beyer, S. Bandyopadhyay, and T. Ideker, "Integrating physical and genetic maps: from genomes to interaction networks," *Nature Reviews Genetics*, vol. 8, no. 9, pp. 699–710, 2007.

[3] N. M. Luscombe, M. M. Babu, H. Y. Yu, M. Snyder, S. A. Teichmann, and M. Gerstein, "Genomic analysis of regulatory network dynamics reveals large topological changes," *Nature*, vol. 431, no. 7006, pp. 308–312, 2004.

[4] L. Hood, J. R. Heath, M. E. Phelps, and B. Y. Lin, "Systems biology and new technologies enable predictive and preventative medicine," *Science*, vol. 306, no. 5696, pp. 640–643, 2004.

[5] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[6] 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," *Mol. Biol. Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[7] P. Zhao, J. Seo, Z. Wang, Y. Wang, B. Shneiderman, and E. P. Hoffman, "In vivo filtering of in vitro expression data reveals MyoD targets," *Comptes Rendus Biologies*, vol. 326, no. 10-11, pp. 1049 – 1065, 2003.

[8] S. B. P. Chargé and M. A. Rudnicki, "Cellular and Molecular Regulation of Muscle Regeneration," *Physiol. Rev.*, vol. 84, no. 1, pp. 209–238, 2004.

[9] L. Feuk, A. R. Carson, and S. W. Scherer, "Structural variation in the human genome," *Nat Rev Genetics*, vol. 7, pp. 85–97, February 2006.

[10] W. Liu, S. Laitinen, S. Khan, M. Vihinen, J. Kowalski, G. Yu, L. Chen, C. M. Ewing, M. A. Eisenberger, M. A. Carducci, W. G. Nelson, S. Yegnasubramanian, J. Luo, Y. Wang, J. Xu, W. B. Isaacs, T. Visakorpi, and S. G. Bova, "Copy number analysis indicates monoclonal origin of lethal metastatic prostate cancer," *Nat Med*, vol. 15, pp. 559–565, May 2009.

[11] H. Li, J. Xuan, Y. Wang, and M. Zhan, "Inferring regulatory networks," *Frontiers in Bioscience*, vol. 13, no. 1, pp. 263–275, 2008.

[12] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, no. 5659, pp. 799–805, 2004.

[13] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[14] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks," *Bioinformatics*, vol. 19, no. 17, pp. 2271–2282, 2003.

[15] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[16] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani, "Modeling T-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[17] J. C. Liao, R. Boscolo, Y. L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, "Network component analysis: Reconstruction of regulatory signals in biological systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 26, pp. 15522–15527, 2003.

[18] H. Kim, W. Hu, and Y. Kluger, "Unraveling condition specific gene transcriptional regulatory networks in Saccharomyces cerevisiae," *BMC Bioinformatics*, vol. 7, p. 165, 2006.

[19] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nat Genet*, vol. 34, no. 2, pp. 166–176, 2003. 10.1038/ng1165.

[20] C. C. Liu, W. S. E. Chen, C. C. Lin, H. C. Liu, H. Y. Chen, P. C. Yang, P. C. Chang, and J. J. W. Chen, "Topology-based cancer classification and related pathway mining using microarray data," *Nucleic Acids Research*, vol. 34, no. 14, pp. 4069–4080, 2006.

[21] T. F. Fuller, A. Ghazalpour, J. E. Aten, T. A. Drake, A. J. Lusis, and S. Horvath, "Weighted gene coexpression network analysis strategies applied to mouse weight," *Mammalian Genome*, vol. 18, no. 6-7, pp. 463–472, 2007.

[22] P. Qiu, Z. J. Wang, and K. J. Liu, "Ensemble dependence model for classification and prediction of cancer and normal gene expression data," *Bioinformatics*, vol. 21, no. 14, pp. 3114–21, 2005.

[23] P. Qiu, Z. J. Wang, K. J. Liu, Z. Z. Hu, and C. H. Wu, "Dependence network modeling for biomarker identification," *Bioinformatics*, vol. 23, no. 2, pp. 198–206, 2007.

[24] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–1182, 2008.

[25] Z. Wei and H. Li, "A markov random field model for network-based analysis of genomic data," *Bioinformatics*, vol. 23, no. 12, pp. 1537–44, 2007.

[26] B. Zhang, H. Li, R. B. Riggins, M. Zhan, J. Xuan, Z. Zhang, E. P. Hoffman, R. Clarke, and Y. Wang, "Differential dependency network analysis to identify condition-specific topological changes in biological networks," *Bioinformatics*, vol. 25, pp. 526–532, December 2009.

[27] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 58, no. 1, pp. 267–288, 1996.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing ed., October 2007.

[29] E. M. Airoldi, "Getting started in probabilistic graphical models," *PLoS Computational Biology*, vol. 3, pp. e252+, Nov 2007.

[30] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *Journal of Machine Learning Research*, vol. 1, no. 1, pp. 49–75, 2000.

[31] D. M. Chickering, *Learning Bayesian networks is NP-complete.* Learning from Data: Artificial Intelligence and Statistics V, New York: Springer-Verlag, 1996.

[32] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[33] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[34] M. Schmidt, A. Niculescu-Mizil, and K. Murphy, "Learning graphical model structure using L1-regularization paths," in *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pp. 1278–1283, AAAI Press, 2007.

[35] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B-Statistical Methodology*, vol. 67, pp. 301–320, 2005.

[37] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007* (J. N. Kok, J. Koronacki, R. L. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, eds.), vol. 4701 of *Lecture Notes in Computer Science*, ch. 28, pp. 286–297, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[38] S.-I. Lee, V. Ganapathi, and D. Koller, "Efficient structure learning of Markov networks using L1-regularization," in *Advances in Neural Information Processing Systems (NIPS 2006)*, 2006.

[39] E. R. Dougherty, S. Kim, and Y. D. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.

[40] B. Efron and R. Tibshirani, *An introduction to the bootstrap*. Monographs on statistics and applied probability; 57, New York: Chapman & Hall, 1993.

[41] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, pp. 2498–2504, November 2003.

[42] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal, "SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms," *BMC Bioinformatics*, vol. 7, no. 1, pp. 43+, 2006.

[43] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining (Adaptive Computation and Machine Learning)*. The MIT Press, August 2001.

[44] J. D. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 9440–9445, August 2003.

[45] C. Y. Lin, A. Strom, V. B. Vega, S. L. Kong, A. L. Yeo, J. S. Thomsen, W. C. Chan, B. Doray, D. K. Bangarusamy, A. Ramasamy, L. A. Vergara, S. Tang, A. Chong, V. B. Bajic, L. D. Miller, J. A. Gustafsson, and E. T. Liu, "Discovery of estrogen receptor alpha target genes and response elements in breast tumor cells," *Genome Biol*, vol. 5, no. 9, p. R66, 2004.

[46] A. Howell, "Pure oestrogen antagonists for the treatment of advanced breast cancer," *Endocrine-Related Cancer*, vol. 13, no. 3, pp. 689–706, 2006.

[47] M. T. Kuo, "Roles of multidrug resistance genes in breast cancer chemoresistance," in *Breast Cancer Chemosensitivity*, vol. 608 of *Advances in Experimental Medicine and Biology*, pp. 23–30, Berlin: Springer-Verlag Berlin, 2007.

[48] R. B. Riggins, A. H. Bouton, M. C. Liu, and R. Clarke, "Antiestrogens, aromatase inhibitors, and apoptosis in breast cancer," in *Vitamins and Hormones - Advances in Research and Applications, Vol 71*, vol. 71 of *Vitamins and Hormones-Advances in Research and Applications*, pp. 201–237, San Diego: Elsevier Academic Press Inc, 2005.

[49] R. B. Riggins, R. S. Schrecengost, M. S. Guerrero, and A. H. Bouton, "Pathways to tamoxifen resistance," *Cancer Letters*, vol. 256, no. 1, pp. 1–24, 2007.

[50] N. N. Iwakoshi, A. H. Lee, and L. H. Glimcher, "The X-box binding protein-1 transcription factor is required for plasma cell differentiation and the unfolded protein response," *Immunological Reviews*, vol. 194, no. 1, pp. 29–38, 2003.

[51] L. H. Ding, J. H. Yan, J. H. Zhu, H. J. Zhong, Q. J. Lu, Z. H. Wang, C. F. Huang, and Q. N. Ye, "Ligand-independent activation of estrogen receptor alpha by XBP-1," *Nucleic Acids Research*, vol. 31, no. 18, pp. 5266–5274, 2003.

[52] Y. Fang, J. H. Yan, L. H. Ding, Y. F. Liu, J. H. Zhu, C. F. Huang, H. Q. Zhao, Q. J. Lu, X. M. Zhang, X. Yang, and Q. N. Ye, "XBP-1 increases ER alpha transcriptional activity through regulation of large-scale chromatin unfolding," *Biochemical and Biophysical Research Communications*, vol. 323, no. 1, pp. 269–274, 2004.

[53] B. P. Gomez, R. B. Riggins, A. N. Shajahan, U. Klimach, A. Wang, A. C. Crawford, Y. Zhu, A. Zwart, M. Wang, and R. Clarke, "Human X-Box binding protein-1 confers both estrogen independence and antiestrogen resistance in breast cancer cell lines," *Faseb Journal*, vol. 21, no. 14, pp. 4013–4027, 2007.

[54] S. Somai, M. Chaouat, D. Jacob, J. Y. Perrot, W. Rostene, P. Forgez, and A. Gompel, "Antiestrogens are pro-apoptotic in normal human breast epithelial cells," *International Journal of Cancer*, vol. 105, no. 5, pp. 607–612, 2003.

[55] K. N. Felekkis, R. P. Narsimhan, R. Near, A. F. Castro, Y. Zheng, L. A. Quilliam, and A. Lerner, "AND-34 activates phosphatidylinositol 3-kinase and induces anti-estrogen resistance in a SH2 and GDP exchange factor-like domain-dependent manner," *Molecular Cancer Research*, vol. 3, no. 1, pp. 32–41, 2005.

[56] R. B. Riggins, L. A. Quilliam, and A. H. Bouton, "Synergistic promotion of c-Src activation and cell migration by Cas and AND-34/BCAR3," *Journal of Biological Chemistry*, vol. 278, no. 30, pp. 28264–28273, 2003.

[57] R. S. Schrecengost, R. B. Riggins, K. S. Thomas, N. S. Guerrero, and A. H. Bouton, "Breast cancer antiestrogen resistance-3 expression regulates breast cancer cell migration through promotion of p130(Cas) membrane localization and membrane ruffling," *Cancer Research*, vol. 67, no. 13, pp. 6174–6182, 2007.

[58] T. Van Agthoven, J. Veldscholte, M. Smid, T. Van Agthoven, and L. C. J. Dorssers, "Functional identification of genes causing estrogen independence.," *Breast Cancer Research and Treatment*, vol. 100, pp. S37–S37, 2006.

[59] M. A. C. Pratt, T. E. Bishop, D. White, G. Yasvinski, M. Menard, M. Y. Niu, and R. Clarke, "Estrogen withdrawal-induced NF-kappa B activity and Bcl-3 expression in breast cancer cells: Roles in growth and hormone independence," *Molecular and Cellular Biology*, vol. 23, no. 19, pp. 6887–6900, 2003.

[60] Y. Zhou, S. Eppenberger-Castori, C. Marx, C. Yau, G. K. Scott, U. Eppenberger, and C. C. Benz, "Activation of nuclear factor-κB (NFκB) identifies a high-risk subset of

hormone-dependent breast cancers," *The International Journal of Biochemistry & Cell Biology*, vol. 37, no. 5, pp. 1130 – 1144, 2005.

[61] M. G. Sanna, J. D. Correia, O. Ducrey, J. Lee, K. Nomoto, N. Schrantz, Q. L. Deveraux, and R. J. Ulevitch, "IAP suppression of apoptosis involves distinct mechanisms: The TAK1/JNK1 signaling cascade and caspase inhibition," *Molecular and Cellular Biology*, vol. 22, no. 6, pp. 1754–1766, 2002.

[62] P. Englebienne and K. Meirleir, *Chronic Fatigue Syndrome: A Biological Approach*. CRC Press, 2002.

[63] P. Viatour, M. Bentires-Alj, A. Chariot, V. Deregowski, L. de Leval, M. P. Merville, and V. Bours, "Nf-kappa b2/p100 induces bcl-2 expression," *Leukemia*, vol. 17, no. 7, pp. 1349–1356, 2003.

[64] A. T. Ferguson, R. G. Lapidus, S. B. Baylin, and N. E. Davidson, "Demethylation of the Estrogen Receptor Gene in Estrogen Receptor-negative Breast Cancer Cells Can Reactivate Estrogen Receptor Gene Expression," *Cancer Research*, vol. 55, no. 11, pp. 2279–2283, 1995.

[65] M. Widschwendter, K. D. Siegmund, H. M. Müller, H. Fiegl, C. Marth, E. Müller-Holzner, P. A. Jones, and P. W. Laird, "Association of breast cancer DNA methylation profiles with hormone receptor status and response to Tamoxifen," *Cancer Research*, vol. 64, no. 11, pp. 3807–3813, 2004.

[66] M. G. Friedrich, D. J. Weisenberger, J. C. Cheng, S. Chandrasoma, K. D. Siegmund, M. L. Gonzalgo, M. I. Toma, H. Huland, C. Yoo, Y. C. Tsai, P. W. Nichols, B. H. Bochner, P. A. Jones, and G. Liang, "Detection of Methylated Apoptosis-Associated Genes in Urine Sediments of Bladder Cancer Patients," *Clinical Cancer Research*, vol. 10, no. 22, pp. 7457–7465, 2004.

[67] A. Noer, A. Boquest, and P. Collas, "Dynamics of adipogenic promoter DNA methylation during clonal culture of human adipose stem cells to senescence," *BMC Cell Biology*, vol. 8, no. 1, p. 18, 2007.

[68] V. Seyfert, S. McMahon, W. Glenn, A. Yellen, V. Sukhatme, X. Cao, and J. Monroe, "Methylation of an immediate-early inducible gene as a mechanism for B cell tolerance induction," *Science*, vol. 250, no. 4982, pp. 797–800, 1990.

[69] T.-L. Cha, B. P. Zhou, W. Xia, Y. Wu, C.-C. Yang, C.-T. Chen, B. Ping, A. P. Otte, and M.-C. Hung, "Akt-mediated phosphorylation of EZH2 suppresses methylation of Lysine 27 in histone H3," *Science*, vol. 310, pp. 306–310, October 2005.

[70] S. Bogdanovich, T. O. B. Krag, E. R. Barton, L. D. Morris, L.-A. Whittemore, R. S. Ahima, and T. S. Khurana, "Functional improvement of dystrophic muscle by myostatin blockade," *Nature*, vol. 420, pp. 418–421, 2002.

[71] M. H. Parker, P. Seale, and M. A. Rudnicki, "Looking back to the embryo: defining transcriptional networks in adult myogenesis," *Nature Review Genetics*, vol. 4, no. 7, pp. 497–507, 2003.

[72] R. Grifone and R. G. Kelly, "Heartening news for head muscle development," *Trends in Genetics*, vol. 23, no. 8, pp. 365–369, 2007.

[73] S. L. Lauritzen, *Graphical Models (Oxford Statistical Science Series)*. Oxford University Press, USA, July 1996.

[74] M. Jordan, *Learning in Graphical Models*. The MIT Press, November 1998.

[75] M. J. Wainwright, P. Ravikumar, and J. D. Lafferty, "High-dimensional graphical model selection using L1-regularized logistic regression," in *Advances in Neural Information Processing Systems (NIPS 2006)*, MIT Press, 2006.

[76] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[77] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[78] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[79] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[80] M. Kolar, L. Song, and E. P. Xing, "Sparsistent learning of varying-coefficient models with structural changes," in *Advances in Neural Information Processing Systems (NIPS 2009)*, 2009.

[81] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[82] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics, Springer, 2nd ed. ed., September 2008.

[83] M. Kanehisa and S. Goto, "KEGG: Kyoto encyclopedia of genes and genomes.," *Nucleic Acids Research*, vol. 28, pp. 27–30, Jan. 2000.

[84] M. F. Ochs, "Knowledge-based data analysis comes of age," *Brief Bioinform*, vol. 11, pp. 30–39, Jan. 2010.

[85] H. Bunke and G. Allermann, "Inexact graph matching for structural pattern recognition," *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245 – 253, 1983.

[86] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.

[87] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78–80, April 2004.

[88] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," tech. rep., German National Research Center for Information Technology Tech. Rep. 148, 2001.

[89] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[90] Z. Shi and M. Han, "Support vector echo-state machine for chaotic time-series prediction," *Neural Networks, IEEE Transactions on*, vol. 18, no. 2, pp. 359–372, 2007.

[91] M. C. Ozturk and J. C. Píncipe, "An associative memory readout for ESNs with applications to dynamical pattern recognition.," *Neural Networks*, vol. 20, no. 3, pp. 377–390, 2007.

[92] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural Networks*, vol. 20, no. 3, pp. 414 – 423, 2007.

[93] B. Zhang and Y. Wang, "Echo state networks with decoupled reservoir states," in *Machine Learning for Signal Processing 2008, IEEE Workshop on*, pp. 444–449, IEEE, 2008.

[94] Z. Deng and Y. Zhang, "Collective behavior of a small-world recurrent neural system with scale-free distribution," *Neural Networks, IEEE Transactions on*, vol. 18, no. 5, pp. 1364–1375, 2007.

[95] Y. Xue, L. Yang, and S. Haykin, "Decoupled echo state networks with lateral inhibition," *Neural Networks*, vol. 20, no. 3, pp. 365 – 376, 2007.

[96] M. C. Ozturk, D. Xu, and J. C. Príncipe, "Analysis and design of echo state networks," *Neural Comput.*, vol. 19, no. 1, pp. 111–138, 2007.

[97] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127 – 149, 2009.

[98] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[99] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *IEEE Computer Society Neural Networks Technology Series*, pp. 112–127, 1990.

[100] M. Buehner and P. Young, "A tighter bound for the echo state property," *Neural Networks, IEEE Transactions on*, vol. 17, no. 3, pp. 820–824, 2006.

[101] H. Jaeger, "Echo state network," *Scholarpedia*, vol. 2, no. 9, 2007.

[102] M. Mehta, *Random matrices and the statistical theory of energy levels*. New York: Academic Press, 1967.

[103] A. Edelman, "The probability that a random real Gaussian matrix has k real eigenvalues, related distributions, and the circular law," *Journal of Multivariate Analysis*, vol. 60, no. 2, pp. 203–232, 1997.

[104] V. L. Girko, "Circular law," *Theory of Probability and Its Applications*, vol. 29, pp. 694–706, 1984.

[105] Z. D. Bai, "Circular law," *Annals of Probability*, vol. 25, no. 1, pp. 494–529, 1997.

[106] T. Tao and V. Vu, "Random matrices: the circular law," *Commun. Contemp. Math.*, vol. 10, no. 2, pp. 261–307, 2008.

[107] T. Tao, V. Vu, and M. Krishnapur, "Random matrices: Universality of esds and the circular law," `arXiv:0807.4898v4 [math.PR]`, 2008.

[108] T. Tao and V. Vu, "Random matrices: the distribution of the smallest singular values," *Geometric And Functional Analysis*, vol. 19, 2010.

[109] V. A. Marenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, p. 457, 1967.

[110] Y. Q. Yin, "Limiting spectral distribution for a class of random matrices," *J. Multivar. Anal.*, vol. 20, no. 1, pp. 50–68, 1986.

[111] Z. D. Bai and Y. Q. Yin, "Limiting behavior of the norm of products of random matrices and two problems of geman-hwang," *Probability Theory and Related Fields*, vol. 73, pp. 555–569, 1986.

[112] Y. Q. Yin, Z. D. Bai, and P. R. Krishnaiah, "On the limit of the largest eigenvalue of the large dimensional sample covariance matrix," *Probability Theory and Related Fields*, vol. 78, no. 4, pp. 509–521, 1988.

[113] M. Ledoux, *The Concentration of Measure Phenomenon*. American Mathematical Society, 2001.

[114] D. Achlioptas, "Database-friendly random projections," in *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (New York, NY, USA), pp. 274–281, ACM, 2001.

[115] S. T. S. Becker and K. Obermayer, eds., *Adaptive Nonlinear System Identification with Echo State Networks*, MIT Press Cambridge, MA, MIT Press Cambridge, MA, 2003.

[116] D. E. Zak, G. E. Gonye, J. S. Schwaber, and F. J. Doyle, "Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insights from an identifiability analysis of an in silico network," *Genome Research*, vol. 13, no. 11, pp. 2396–2405, 2003.

[117] J. R. Pollack, T. Sørlie, C. M. Perou, C. A. Rees, S. S. Jeffrey, P. E. Lonning, R. Tibshirani, D. Botstein, A.-L. L. Børresen-Dale, and P. O. Brown, "Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors.," *PNAS*, vol. 99, pp. 12963–12968, October 2002.

[118] D. A. Peiffer, J. M. Le, F. J. Steemers, W. Chang, T. Jenniges, F. Garcia, K. Haden, J. Li, C. A. Shaw, J. Belmont, S. W. Cheung, R. M. Shen, D. L. Barker, and K. L. Gunderson, "High-resolution genomic profiling of chromosomal aberrations using infinium whole-genome genotyping," *Genome Res*, vol. 16, pp. 1136–1148, September 2006.

[119] D. J. Nancarrow, H. Y. Handoko, M. S. Stark, D. C. Whiteman, and N. K. Hayward, "SiDCoN: A tool to aid scoring of DNA copy number changes in SNP chip data," *PLoS ONE*, vol. 2, p. e1093, October 2007.

[120] P. Lamy, C. L. Andersen, L. Dyrskjot, N. Torring, and C. Wiuf, "A hidden Markov model to estimate population mixture and allelic copy-numbers in cancers using Affymetrix SNP arrays," *BMC Bioinformatics*, vol. 8, p. 434, November 2007.

[121] G. Assie, T. LaFramboise, P. Platzer, J. Bertherat, C. A. Stratakis, and C. Eng, "SNP arrays in heterogeneous tissue: Highly accurate collection of both germline and somatic

genetic information from unpaired single tumor samples," *Am J Hum Genet*, vol. 82, pp. 903–915, April 2008.

[122] G. Yamamoto, Y. Nannya, M. Kato, M. Sanada, R. L. Levine, N. Kawamata, A. Hangaishi, M. Kurokawa, S. Chiba, D. G. Gilliland, H. P. Koeffler, and S. Ogawa, "Highly sensitive method for genomewide detection of allelic composition in nonpaired, primary tumor specimens by use of Affymetrix single-nucleotide-polymorphism genotyping microarrays," *Am J Hum Genet*, vol. 81, pp. 114–126, July 2007.

[123] H. Goransson, K. Edlund, M. Rydaker, M. Rasmussen, J. Winquist, S. Ekman, M. Bergqvist, A. Thomas, M. Lambe, R. Rosenquist, L. Holmberg, P. Micke, J. Botling, and A. Isaksson, "Quantification of normal cell fraction and copy number neutral LOH in clinical lung cancer samples using SNP array data," *PLoS ONE*, vol. 4, p. e6057, June 2009.

[124] H. Bengtsson, R. Irizarry, B. Carvalho, and T. P. Speed, "Estimation and assessment of raw copy numbers at the single locus level," *Bioinformatics*, vol. 24, pp. 759–767, March 2008.

[125] R. Beroukhim, G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J. C. Lee, J. H. Huang, S. Alexander, J. Du, T. Kau, R. K. Thomas, K. Shah, H. Soto, S. Perner, J. Prensner, R. M. Debiasi, F. Demichelis, C. Hatton, M. A. Rubin, L. A. Garraway, S. F. Nelson, L. Liau, P. S. Mischel, T. F. Cloughesy, M. Meyerson, T. A. Golub, E. S. Lander, I. K. Mellinghoff, and W. R. Sellers, "Assessing the significance of chromosomal aberrations in cancer: Methodology and application to glioma," *PNAS*, vol. 104, no. 50, pp. 20007–20012, 2007.

[126] S. Durinck, C. Ho, N. J. Wang, W. Liao, L. R. Jakkula, E. A. Collisson, J. Pons, S.-W. Chan, E. T. Lam, C. Chu, K. Park, S.-w. Hong, J. S. Hur, N. Huh, I. M. Neuhaus, S. S. Yu, R. C. Grekin, T. M. Mauro, J. E. Cleaver, P.-Y. Kwok, P. E. LeBoit, G. Getz,

K. Cibulskis, J. C. Aster, H. Huang, E. Purdom, J. Li, L. Bolund, S. T. Arron, J. W. Gray, P. T. Spellman, and R. J. Cho, "Temporal dissection of tumorigenesis in primary cancers," *Cancer Discovery*, 2011.

Appendix A

A.1 Biographical Sketch

Bai Zhang received his BS and MS degrees in the Department of Automation at Tsinghua University, Beijing, China, in 2004 and 2006, respectively. Since August 2006, he has been a doctoral student and graduate research assistant in the Bradley Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University (Virginia Tech), Virginia, United States of America, under the supervision of Dr. Yue Wang. His research interests include machine learning and its applications to bioinformatics and computational biology.

A.2 List of Publications Related to the Dissertation

Journal Papers

[1] G. Yu*, **B. Zhang***, J. Xu, G. S. Bova, I.-M. Shih, and Y. Wang, "BACOM: in silico detection of genomic deletion types and correction of normal cell contamination in copy number data", *Bioinformatics*, 27(11):1473-1480, 2011. (* Joint first author)

[2] **B. Zhang**, Y. Tian, L. Jin, H. Li, I.-M. Shih, S. Madhavan, R. Clarke, E. P. Hoffman, J. Xuan, L. Hilakivi-Clarke, and Y. Wang, "DDN: a caBIG® analytical tool for differential network analysis", *Bioinformatics*, 27(7):1036-1038, 2011.

[3] R. Clarke, A. N. Shajahan, Y. Wang, J. J. Tyson, R. B. Riggins, L. M. Weiner, W. T. Bauman, J. Xuan, **B. Zhang**, C. Facey, H. Aiyer, K. Cook, F. E. Hickman, I. Tavassoly, A. Verdugo, C. Chen, A. Zwart, A. Wärri, and L. A. Hilakivi-Clarke, "Endoplasmic reticulum stress, the unfolded protein response, and gene network modeling in antiestrogen resistant breast cancer", *Hormone Molecular Biology and Clinical Investigation*, 5(1):35-44, 2011.

[4] **B. Zhang**, H. Li, R. B. Riggins, M. Zhan, J. Xuan, Z. Zhang, E. P. Hoffman, R. Clarke, and Y. Wang, "Differential dependency network analysis to identify condition-specific topological changes in biological networks," *Bioinformatics*, 25(4):526-532, 2009.

Book chapters

[1] **B. Zhang**, H. Li, R. Clarke, L. Hilakivi-Clarke, and Y. Wang, "Differential dependency network analysis to identify topological changes in biological networks," in F. Emmert-Streib and M. Dehmer (Eds.) *Medical Biostatistics for Complex Diseases*, pp. 185-204, Wiley-VCH, Weinheim, 2010.

Conference Papers

[1] Y. Tian, **B. Zhang**, I.-M. Shih, Y. Wang, "Knowledge-guided differential dependency network learning for detecting structural changes in biological networks", in *Proc. of ACM Conference on Bioinformatics and Computational Biology (ACM BCB 2011)*, Aug. 2011.

[2] **B. Zhang**, Yue Wang, "Learning structural changes of Gaussian graphical models in controlled experiments", in *Proc. of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, Avalon, CA: 701-708, July 2010.

[3] G. Yu, **B. Zhang**, J. Xu, I.-M. Shih, and Y. Wang, "Accurate estimation of genomic deletions and normal cell contamination by Bayesian analysis of mixtures," in *Proc. IEEE Intl Conf. on Bioinformatics & Biomedicine*, Washington D.C., USA, Nov. 2009.

[4] **B. Zhang**, Y. Wang, "Echo state networks with decoupled reservoir states," in *Proc. IEEE Machine Learning for Signal Processing*, Cancún, Mexico, 2008.

[5] L. Xiong, C. Wang, **B. Zhang**, Y. Wang, E. P. Hoffman, R. Clarke, and J. Xuan, "Inferring condition-specific miRNA-gene modules from miRNA and mRNA profiling data," in *Proc. Intl Conf. on Bioinformatics, Computational Biology, Genomics and Chemoinformatics*, Orlando, USA, July 2008.

Manuscripts in Preparation / under Review

[1] **B. Zhang**, D. J. Miller, Y. Wang, "Nonlinear system modeling with random matrices: echo state networks revisited," under review, *IEEE Trans. Neural Networks*.

[2] **B. Zhang**, Yue Wang, "Learning structural changes of graphical models between two conditions", in preparation.