MODELING PERTURBATIONS IN GENE REGULATORY AND SIGNALING NETWORKS

By

NIRMALYA BANDYOPADHYAY

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011
To
my beloved wife Shruti
and
our baby whom we lost during our journey to Ph.D.
ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Sanjay Ranka and Dr. Tamer Kahveci, for all their guidance, support and the numerous opportunities they provided throughout my studies and research. I would also like to thank my committee members, Dr. Alin Dobra, Dr. Arunava Banerjee and Dr. Valérie de Crécy-Lagard, for all of their help and valuable suggestions.

I would like to thank Dr. Paul Gader for his inspiration and excellent teaching in his three courses of machine learning.

I express my gratitude to my beloved wife Shruti, without whose relentless support, inspiration and sacrifice my Ph.D. could not have been possible.

Finally, my special thanks goes to my family and friends for their continued support and guidance to overcome the obstacles during my journey to Ph.D.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1  INTRODUCTION</td>
<td>12</td>
</tr>
<tr>
<td>2  BACKGROUND</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Feature Selection in Gene Expression Data</td>
<td>19</td>
</tr>
<tr>
<td>2.2 Analysis of Perturbation in Gene Networks</td>
<td>20</td>
</tr>
<tr>
<td>2.3 Synthetic Sickness Lethality</td>
<td>22</td>
</tr>
<tr>
<td>3  PATHWAY BASED FEATURE SELECTION FOR CANCER MICROARRAY DATA</td>
<td>24</td>
</tr>
<tr>
<td>3.1 Algorithm</td>
<td>27</td>
</tr>
<tr>
<td>3.1.1 Picking the First Feature: Where to Start?</td>
<td>28</td>
</tr>
<tr>
<td>3.1.2 Selecting the Candidate Features</td>
<td>29</td>
</tr>
<tr>
<td>3.1.3 Selecting the Best Candidate Gene</td>
<td>30</td>
</tr>
<tr>
<td>3.1.4 Exploration of Training Space</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Data Set and Experiments</td>
<td>33</td>
</tr>
<tr>
<td>3.2.1 Experimental Setup</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Pathways with Unresolved Genes</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 Biological Validation of Selected Features</td>
<td>38</td>
</tr>
<tr>
<td>3.2.4 Comparison with Van ’t Veer’s Gene Signature</td>
<td>40</td>
</tr>
<tr>
<td>3.2.5 Comparison with I-RELIEF</td>
<td>42</td>
</tr>
<tr>
<td>3.2.6 Cross Validation Experiments</td>
<td>43</td>
</tr>
<tr>
<td>3.2.7 Experiments on Fully Connected Pathways</td>
<td>48</td>
</tr>
<tr>
<td>3.2.8 Contribution of Pathway Information</td>
<td>48</td>
</tr>
<tr>
<td>3.3 Discussion</td>
<td>49</td>
</tr>
<tr>
<td>4  MODELING PERTURBATIONS USING GENE NETWORKS</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Methods</td>
<td>53</td>
</tr>
<tr>
<td>4.1.1 Notation and Problem Formulation</td>
<td>53</td>
</tr>
<tr>
<td>4.1.2 Overview of Our Solution</td>
<td>55</td>
</tr>
<tr>
<td>4.1.3 Computation of the Prior Density Function</td>
<td>56</td>
</tr>
<tr>
<td>4.1.4 Approximation of the Objective Function</td>
<td>60</td>
</tr>
<tr>
<td>4.1.5 Calculation of Likelihood Density Function</td>
<td>61</td>
</tr>
</tbody>
</table>
4.1.6 Objective Function Optimization ........................................ 63
4.2 Experiments ................................................................. 64
  4.2.1 Evaluation of Biological Significance ............................... 65
  4.2.2 Evaluation of the Rankings of Neighbor Genes ...................... 66
  4.2.3 Comparison to Other Methods ....................................... 68
  4.2.4 Sensitivity to the Gap between Primary and Secondary Effects .. 72
4.3 Discussion ................................................................. 73

5 IDENTIFYING DIFFERENTIALLY REGULATED GENES ................. 74
  5.1 Methods ................................................................. 78
    5.1.1 Notation and Problem Formulation ............................... 78
    5.1.2 Overview of the Solution ....................................... 81
    5.1.3 Computation of the Prior Density Function ..................... 82
    5.1.4 Approximation of the Objective Function ....................... 89
    5.1.5 Computation of Likelihood Density Function ................... 91
    5.1.6 Objective Function Optimization ................................ 94
  5.2 Results and Discussion ................................................ 95
    5.2.1 Comparison to Other Methods .................................... 97
    5.2.2 Statistical Significance Experiment ............................. 100
  5.3 Discussion ............................................................. 104

6 SSLPRED: PREDICTING SYNTHETIC SICKNESS LETHALITY .......... 105
  6.1 Methods ............................................................... 109
    6.1.1 Problem Formulation and Notation ................................ 109
    6.1.2 Between Pathway Conjectures ................................... 110
    6.1.3 Regression Based Solution ...................................... 112
  6.2 Experiments ........................................................... 116
    6.2.1 Datasets .......................................................... 116
    6.2.2 Comparison with Hescott’s Method ................................ 117
  6.3 Discussion ............................................................. 122

7 CONCLUSION ...................................................................... 123

REFERENCES ....................................................................... 124

BIOGRAPHICAL SKETCH ...................................................... 136
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1 List of publications supporting the first twenty features obtained from BCR data set about their responsibility for cancer.</td>
<td>36</td>
</tr>
<tr>
<td>3-2 Accuracy of our algorithm obtained from Cross Validation experiments on real pathway.</td>
<td>44</td>
</tr>
<tr>
<td>4-1 The table enumerates the truth values for the two binary feature functions.</td>
<td>58</td>
</tr>
<tr>
<td>4-2 List of top 25 genes that are mostly affected by external perturbation.</td>
<td>65</td>
</tr>
<tr>
<td>5-1 Enumeration of the values of $Z_i$, $Z_j$ and $X_{ij}$ for different values of $S_{Ai}$, $S_{Bi}$, $S_{Aj}$ and $S_{Bj}$.</td>
<td>79</td>
</tr>
<tr>
<td>5-2 Enumeration of five different unary feature functions $F_1$, $F_2$, $F_3$, $F_6$ and $F_7$.</td>
<td>85</td>
</tr>
<tr>
<td>5-3 The table enumerates the truth values for the binary feature function left external equality $(f_4)$.</td>
<td>87</td>
</tr>
<tr>
<td>5-4 The table enumerates the truth values for the binary feature function right external equality $(f_5)$.</td>
<td>88</td>
</tr>
<tr>
<td>6-1 Summarization of the feature functions of the regression model.</td>
<td>115</td>
</tr>
<tr>
<td>6-2 BPMs with p-Values less than 0.1 [%].</td>
<td>119</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>A portion of colorectal cancer pathway obtained from KEGG database.</td>
<td>13</td>
</tr>
<tr>
<td>1-2</td>
<td>This figure illustrates the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway.</td>
<td>14</td>
</tr>
<tr>
<td>3-1</td>
<td>Part of pancreatic cancer pathway adapted from KEGG showing the gene-gene interactions.</td>
<td>25</td>
</tr>
<tr>
<td>3-2</td>
<td>We plot the real data points $R_{P_{10}}$, $\cdots$, $R_{P_{100}}$ along with the constructed function $f(s)$ against $s$, the fraction of the current pathway.</td>
<td>35</td>
</tr>
<tr>
<td>3-3</td>
<td>Comparison of test accuracy of our signature and van't Veer's 70-gene prognostic signature on real pathway.</td>
<td>39</td>
</tr>
<tr>
<td>3-4</td>
<td>The standard deviation of the accuracy of our method for different number of features. The x-axis denotes different number of features.</td>
<td>40</td>
</tr>
<tr>
<td>3-5</td>
<td>Comparison of test accuracy of our method (BPFS) to that of I-RELIEF on real pathway.</td>
<td>41</td>
</tr>
<tr>
<td>3-6</td>
<td>Comparison of test accuracy of our method (BPFS) to that of I-RELIEF on fully connected pathway.</td>
<td>45</td>
</tr>
<tr>
<td>3-7</td>
<td>Comparison of test accuracy of our method (BPFS) to when we select the genes only based on marginal classification power.</td>
<td>46</td>
</tr>
<tr>
<td>4-1</td>
<td>Illustration of the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway.</td>
<td>52</td>
</tr>
<tr>
<td>4-2</td>
<td>Illustration of a small hypothetical gene network with a perturbation and the graph for Markov random field created for that gene network.</td>
<td>56</td>
</tr>
<tr>
<td>4-3</td>
<td>Frequency of average distance of rankings over training and testing data.</td>
<td>67</td>
</tr>
<tr>
<td>4-4</td>
<td>Comparison of our method to SSEM and t-test.</td>
<td>69</td>
</tr>
<tr>
<td>4-5</td>
<td>Comparison of accuracies with SSEM and Student's t test while varying the ratio of gaps of primarily and secondarily affected genes.</td>
<td>71</td>
</tr>
<tr>
<td>5-1</td>
<td>Illustration of the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway.</td>
<td>75</td>
</tr>
<tr>
<td>5-2</td>
<td>A hypothetical gene network and corresponding Markov random graph.</td>
<td>77</td>
</tr>
<tr>
<td>5-3</td>
<td>Illustration of different components of CMRF and connectivity between them.</td>
<td>83</td>
</tr>
</tbody>
</table>
5-4 Comparison of our method CMRF to SMRF, SSEM and t-test. The number of primarily differentially affected genes is 40. 99

5-5 Illustration of the statistical significance test. 102

6-1 Illustration of the concepts of Synthetic Sickness and Lethality and Between Pathway Models. 106

6-2 This figure depicts the layered neighbor structure around a gene interaction edge $(g_a, g_b)$. 112

6-3 Comparison of SSLPred with the method from Hescott et al. 118

6-4 Comparison of SSLPred with the method from Hescott et al. on Brady and Ma datasets for p-Values $\leq 0.1$. 120
Genes interact with each other through gene networks. A perturbation in the organism tissue can propagate to different parts of the network and affect the expression of the other genes. A perturbation can take place because of a disorder due to cancer, radiation or medication. In this work, we propose to model and analyze the impact of perturbations in gene networks. In particular, we are interested in determining the subset of genes that are directly and indirectly affected due to these perturbations, and how this impact may vary in multiple groups with different characteristics.

This problem have been touched upon in the existing literature. However, most of those methods employs statistical and machine learning based techniques that consider the gene expression alone as the sole input data and do not consider the genetic interaction network to solve the problems.

In this work, we resort to a hypothesis for solving the problem that we just specified. We formally state the hypothesis as follows– integrating gene networks with gene expression enables us to analyze the effect of perturbations in a more effective and comprehensive fashion. We address our problem by breaking it into four connected sub-problems. In all the four sub-problems of this work, we leverage the knowledge from gene network to build machine learning based solutions to solve the sub problems.
Our thesis establishes the justification behind this hypothesis in two ways. First, our methods are able to produce more accurate methods compared to the existing ones as evident in the results of the experiments. Instead of producing binary decisions, our methods build probabilistic models that enable us to estimate the confidence in their predictions. Second, the computational biologists will be able to inspect which pathways have been affected and to which extent.
CHAPTER 1
INTRODUCTION

Genes interact with each other through gene networks. In this work, the term \textit{gene networks} encompasses both regulatory and signaling networks. We model gene networks as graph structure, where every node corresponds to a gene. One gene has a directed edge to another one, if the first gene regulates the expression of the second gene. \textit{Expression} is the process by which a gene uses its sequence information to synthesize a functional protein product. We denote two genes as \textit{neighbors} if they share an edge. Figure 1-1 shows an example of a small gene network; a portion of the colorectal cancer pathway adapted from KEGG database. \( \rightarrow \) stands for \textit{activation}, where the first gene at the start of the arrow increases the expression (up-regulation) of the other gene. \( \leftarrow \) implies \textit{inhibition} where the first gene decrease the expression (down-regulation) of the other gene. For example, the gene K-Ras activates Raf, while PKB/Akt inhibits CASP9. Thus, K-Ras and Raf are neighbors.

As genes interact through the connected gene network, if there is a \textit{perturbation} in the organism tissue, it potentially propagates to different parts of the network and affects the expression of the genes. In this proposal, we define perturbation, in a broader sense, as an external cause that can affect the genes in the gene network. This definition includes different kinds of cancers, radiation, medication and toxic elements as they all change gene expression. A perturbation initially affects some genes and changes their expressions. We call these genes as \textit{primarily affected genes}. These primarily affected genes in the course of time change the expressions of their neighbors through gene networks. We denote these second category of genes as \textit{secondarily affected genes}.

Figure 1-2 illustrates the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway. In this figure, gene K-Ras, Raf and Cob42Roc
Figure 1-1. A portion of colorectal cancer pathway obtained from KEGG database (©1995-2011 Kanehisa Laboratories). The figure represents the genes and genetic interactions in that pathway. The green rectangles represent the genes. There are two classes of interactions indicated by two kinds of arrows. The $\rightarrow$ arrows stand for activation where the first gene at the start of the arrow increases the expression (up-regulation) of the other gene. $\leftarrow$ implies inhibition while the first gene decreases the expression (down-regulation) of the other gene. For example, the gene K-Ras activates Raf, while PKB/Akt inhibits CASP9.

are primarily affected and MEK, Ral and RalGDS are secondarily affected through interactions.

The gene expression dataset with perturbation typically consists of two parts, control data and non-control data. The control data is a snapshot of the tissue without any affect of perturbation. The non-control dataset is obtained after the administration of perturbation. Sometimes the non-control data can be collected over several time points as a time series data after the perturbation takes place. The exact way to prepare the data depends on the type of the experiment. For cancer, control datasets are prepared by collecting tissues from non-cancerous organisms, or non-cancerous tissues from
Figure 1-2. This figure illustrates the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway. The pathway is taken from the KEGG database. The solid rectangles denote the genes affected directly by the perturbation, the dashed rectangles indicate genes secondarily affected through interactions. The dotted rectangles denote the genes that are not affected by the perturbation. → implies activation and ⊥ implies inhibition. In this figure, gene K-Ras, Raf and Cob42Roc are primarily affected and MEK, Ral and RalGDS are secondarily affected through interactions.

the patients. The non-control dataset is collected from actual diseased tissue. For the deletion mutant experiments, an experiment specific gene is knocked out from the chromosome and gene expressions are taken before and after the deletion.

Our high level problem, in this context, is to analyze the characteristic of perturbation in gene networks. To solve this problem, we propose to solve the following sub problems:

1. Which genes are affected as a result of the perturbation? Can we refine the set of affected genes to classify between the control and non-control groups?

2. Can we predict whether a gene is affected as a result of the perturbation or due to its interaction with the other genes in the gene network?

3. Sometimes we have datasets from two slightly different populations, such as African American vs Caucasian American. Can we identify those genes that behave in a different way between those two groups? What is the reason of this discrepancy?

4. Some pathways (sub-network) in gene network have duplicate pathways as back up systems. If one of those pathways is damaged because of the perturbation, the
backup system activates and compensates for the damage. Can we identify those backup pathways?

Some of this problems have been touched upon in the existing literature, a brief summary of which can be found in Chapter 2. However, most of these methods employ statistical and machine learning based techniques that considers the gene expression alone as the sole input data and do not consider the genetic interaction network to solve the problems.

In this work, we resort to a hypothesis for solving the set of problems that we just enlisted. We formally state the hypothesis as follows—integrating gene networks with gene expression enables us to analyze the effect of perturbation in a more effective and comprehensive fashion. In all the four subtasks of this work, we leverage the knowledge from gene network to build machine learning based solutions to solve these sub problems.

Our work establishes the justification behind this hypothesis in two ways. First, our methods are able to produce more accurate methods compared to the existing methods as evident in the experimental results. In stead of producing binary decisions, our methods build probabilistic models that enable us to estimate the confidence in their predictions. Second, the computational biologists will be able to inspect which pathways have been affected and to which extent.

Here we elaborate in brief, our approach to these described problems.

**Pathway based feature selection of cancer microarray data.** Classification of cancer is an important problem, as class specific medication improves the treatment efficiency and reduces unwanted side effects. Classification algorithms based on gene expressions often have better prediction accuracy than the ones based on clinical markers. However, classification algorithms are prone to overfitting for microarray data due to large number of features. One way to avoid this problem is to select a small set of relevant features and ignore the remaining ones.
This chapter develops a new feature selection method called *Biological Pathway based Feature Selection (BPFS)* for microarray data. Note that here we consider the cause of cancer as the perturbation in the context of the main work. Unlike most of the existing methods, our method integrates signaling and gene regulatory pathways with gene expression data so that the selected features have significant classification accuracy and also have least interaction among each other on the pathway. Thus, BPFS selects a biologically meaningful feature set that is minimally redundant.

Our experiments on published breast cancer datasets demonstrate that all of the top 20 genes found by our method are associated with cancer. Furthermore, the classification accuracy of our signature is up to 18% better than that of van’t Veer’s 70 gene signature, and it is up to 8% better accuracy than the best published feature selection method, I-RELIEF. Chapter 3 discusses this work.

**Modeling perturbations using gene networks.** External factors such as radiation, drugs or chemotherapy can alter the expressions of a subset of genes. We call these genes the *primarily affected* genes. Primarily affected genes eventually can change the expressions of other genes as they activate/suppress them through interactions. Measuring the gene expressions before and after applying an external factor (i.e., perturbation) in microarray experiments can reveal how the expression of each gene changes. It however can not identify the cause of the change.

We consider the problem of identifying primarily affected genes given the expression measurements of a set of genes before and after the application of an external perturbation. We develop a novel probabilistic method to quantify the cause of differential expression of each gene. Our method considers the possible gene interactions in regulatory and signaling networks for a large number of perturbed genes. It uses a Bayesian model to capture the dependency between the genes.

Our experiments on both real and synthetic datasets demonstrate that our method can find primarily affected genes with high accuracy. Our experiments also suggest
that our method is significantly more accurate than the competing method SSEM and Student’s t-test. Chapter 4 provides a detailed description of this work.

**Identifying differentially regulated genes.** Microarray experiments often measure expressions of genes taken from sample tissues in the presence of external perturbations such as medication, radiation, or disease. Typically in such experiments, gene expressions are measured before and after the application of external perturbation. We focus on an important class of such microarray experiments that inherently have two groups of tissue samples. The external perturbation can change the expressions of some genes directly or indirectly through gene interaction network. When such different groups exist, the expressions of genes after the perturbation can be different between the two groups. It is not only important to identify the genes that respond differently across the two groups, but also to mine the reason behind this differential response. We aim to identify the cause of this differential behavior of genes, whether because of the perturbation or due to interactions with other genes in two group perturbation experiments.

We propose a new probabilistic Bayesian method with Markov Random Field to find such genes. Our method incorporates information about relationship from gene networks as prior information. Experimental results on synthetic and real datasets demonstrate the superiority of our method compared to existing techniques. Chapter 5 elaborates on this work.

**Prediction of synthetic sickness lethality relationships.** Two genes in an organism have a *Synthetic Sickness Lethality* (SSL) interaction, if their joint deletion leads to a lower than expected fitness. Synthetic Gene Array (SGA) is a technique that helps in identifying SSL values for pairs of genes in a given set of genes. SSL interactions are useful to discover the co-expressed gene groups in the regulatory and signaling networks. Also, they are used to unravel the pair of pathways (subset of physically interacting genes) that substitute the functions of each other. Generating an
SGA entry is costly as it requires producing and monitoring a *double mutant* (a progeny with two mutated genes). Generating a comprehensive SGA can be very expensive as the number of gene pairs is quadratic in the number of genes of the corresponding organism.

In this chapter, we develop a new method *SSLPred* to predict the SSL interactions in an organism. In the present context, single and double deletion of genes have been modeled as the external perturbation. Our method is built on the concept of *Between Pathway Models (BPM)*, where majority of the SSL pairs span across the two functionally complementing pathways. We develop a regression based approach that learns the mapping between the gene expressions of single deletion mutant to the corresponding SGA entries. We compare our method to the one by Hescott et al. for predicting the GI (Genetic Interaction) score of *Saccharomyces cerevisiae* (*S. cerevisiae*) on four benchmark datasets. On different experimental setups, on average SSLPred performs significantly better compared to the other method. Chapter 6 elaborates on this approach.
CHAPTER 2
BACKGROUND

In this chapter we briefly describe the existing work relevant to my work.

2.1 Feature Selection in Gene Expression Data

Feature selection is an important area in data mining for preprocessing of data. Feature selection techniques select a subset of features to reduce relatively redundant, noisy and irrelevant part of the data. The reduced set of features increases the speed of the data mining algorithms, improves accuracy and understandability of result. Feature selection is often used in areas such as sequence, microarray, and mass-spectra analysis [1]. The popular feature selection methods can be broadly categorized into the following:

- **FILTER METHODS.** [2–4] These are widely studied methods that work independent of the classifier. They rank the features depending on the intrinsic properties of data. One such method is to select sets of features whose pairwise correlations are as low as possible.

- **WRAPPER METHODS.** [5, 6] These methods embed the feature selection criteria into the searching of subset of features. They use a classification algorithm to select the feature set and evaluate its quality using the classifier.

- **EMBEDDED METHODS.** [7] These approaches select features as a part of the classification algorithm. Similar to the wrapper methods, they interact with the classifier, but at a lower cost of computation.

All the above-mentioned traditional feature selection methods ignore the interactions of the genes. Considering each gene as an independent entity can lead to redundancy and low classification accuracy as many genes can have similar expression patterns.

Several recent works on microarray feature selection have leveraged metabolic and gene interaction pathways in their methods. Vert et al. [8] encoded the graph and the set of expression profiles into kernel functions and performed a generalized form of canonical correlation analysis in the corresponding reproducible Hilbert spaces. Rapaport et al. [9] proposed an approach based on spectral decomposition of gene expression profiles with respect to the eigenfunctions of the graph. Wei et al. [10]
proposed a spatially correlated mixture model where they extracted gene specific prior probabilities from gene network. Wei et al. [11] developed a Markov random field based method for identifying genes and subnetworks that are related to a disease. A drawback of the last two model based approaches is that the number of parameters to be estimated is proportional to the number of genes. So optimizing the objective function is costly as the number of genes in microarrays are more than 20,000. Li et al. [12] introduced a network constraint regularization procedure for linear regression analysis, which formulates the Laplacian of the graph extracted from genetic pathway as a regularization constraint.

One limitation of all the above mentioned methods that use biological pathway is that, all of them consider genetic interactions between immediate neighbors on the pathway. None of them explicitly consider interactions that are beyond immediate neighbors. Also, most of them performed quantitative analysis of the selected features on simulated datasets. So it is not possible to quantify the accuracy of those selected features on some real datasets only from the results in these papers. Additional experiments on real microarray datasets are required to justify those methods and their set of features. Also, as the reconstruction of genetic pathway is yet to be completed, we cannot always map a microarray entry to the biological pathway. They do not consider the implications of those missing information. In this chapter, we introduce a new microarray feature selection method that addresses these issues.

2.2 Analysis of Perturbation in Gene Networks

Existing methods to identify the primarily affected genes such as association analysis techniques [13, 14], haplo-insufficiency profiling [15–17] and chemical-genetic interaction mapping [18] require additional information such as fitness based assays of drug response or a library of genetic mutants. Bernardo et al. suggested a regression based approach called MNI based on the assumption that the internal genetic interactions are offset by the external perturbation [19]. It estimates gene-gene
interaction coefficients from the control data. It then uses those coefficients to predict the target genes in the non-control data. Cosgrove et al. proposed a method named SSEM that is similar to MNI [20]. SSEM models the effect of perturbation by an explicit change of gene expression from that of the unperturbed state. Vaske et al. developed a method to infer the affected regulatory networks due to external perturbations using a graphical model called probabilistic factor graph [21]. These methods have several limitations.

- **LACK OF GENE INTERACTION DATA.** The existing methods do not employ regulatory or signaling networks to model gene-gene interactions. Since gene networks are manually curated using domain experts, they are reliable sources of gene interactions. Utilizing them has the potential to more accurately solve the problem of identifying primarily affected genes.

- **LIMITED PERTURBATIONS.** These methods are suitable when only a very small number of genes are perturbed, e.g., the genetic perturbation experiments are often designed for single gene perturbations [13]. However, external effects such as radiation can alter the expression of many genes directly making the existing methods to be of limited use.

- **SIMPLISTIC MODELS.** Most of these methods provide only the set of genes that are directly affected by the perturbations and do not specify any error bounds. However, a non-probabilistic inference oversimplifies the problem especially in cases when a small number of gene expression measurements are available. As a result, these methods can overfit the data, making the solution unreliable.

Several recent studies aim to identify DE genes in multiple groups of data points. maSigPro is a two stage regression based method that identifies genes that demonstrate differential gene expression profiles across multiple experimental groups [22]. Hong et al. proposed a functional hierarchical model for detecting temporally differentially expressed genes between two experimental conditions [23]. They modeled gene expressions by basis function expansion and estimate the parameters using a Monte Carlo EM algorithm. Tai et al. ranked DE genes using data from replicated microarray time course experiments, where there are multiple biological conditions [24]. They derived a multisample multivariate empirical Bayes
statistic for ranking genes. Angelini et al. proposed a Bayesian method for detecting temporally DE genes between two experimental conditions [25]. Deun et al. developed a Bayesian method to find the genes that are differentially expressed in a single tissue or condition over multiple tissues or conditions [26]. All these methods identify differentially expressed genes in multiple groups. However, none of these methods analyzed the primary and secondary effects in a two group perturbation experiment.

### 2.3 Synthetic Sickness Lethality

Recent studies on synthetic sickness and lethality analysis opened up new directions in the areas of functional genomics. These works can be classified into two categories, named experimental and computational. We provide brief overviews on both the groups.

**Experimental.** In a seminal work by Tong et al. pairwise single deletion strains are crossed to produce arrays of double-mutant strains called synthetic gene array (SGA) [27]. The final result is achieved by an ordered array of double-mutant haploid strains whose growth rates are monitored by visual inspection or image analysis of colony size. A different approach, dSLAM (diploid-based synthetic lethal analysis with microarrays) identifies genetic interactions using pools of barcoded yeast mutants [28]. This method measures differential enrichment of double mutants growing in competitive culture using barcode microarrays and determines genetic interactions. The EMAP (epistatic mini-array profile) strategy enables comprehensive and quantitative measurements of genetic interactions between pairs of mutations within a defined subset of genes that are related to one or more specific biological processes. EMAPs are produced by systematically generating yeast strains carrying each pair of mutations and measuring their growth rates [29].

**Computational.** Kelly and Ideker introduced Between Pathway Models (BPM) by combining synthetic sickness and lethality informations from EMAP data with information on protein-protein, protein-DNA or metabolic networks [30]. In a probabilistic way, their
model uncovered almost two thousand genetic interactions spanned either between or within pathways. Their paper showed that two pathways in a BPM have significantly more SSL edges than physical edges between them. This is just the opposite for within pathways, where physical edges outnumber SSL edges. This observation is logically supported by the functionally complementing nature of the consisting pathways. A smaller number of between pathway physical edges indicates that the two pathway modules are functionally independent. So, destroying one of them is not lethal to the other pathway. A higher number of between pathway SSLs implies that the two pathways are complementary, and disrupting both of them can make the organism non-viable.

Hescott et al. [31] proposed a new method to validate BPMs using single gene deletion microarray data. They evaluated the quality of the BPMs from four different studies and described how their methods might be extended to refine BPM pathways.

Kelley and Kingsford [32] developed a new method called Expected Graph Compression to identify compensatory pathways (BPMs) by clustering genes into modules and establishing relationships between those modules. They used this framework to apply a graph clustering method called graph summarization to EMAP dataset.

Though researches in these two avenues enriched our understanding of gene interactions and gene networks, we did not find any predictive model to predict the EMAP scores.
CHAPTER 3
PATHWAY BASED FEATURE SELECTION FOR CANCER MICROARRAY DATA

An important challenge in cancer treatment is to classify a patient to an appropriate cancer class. This is because class specific treatment reduces toxicity and increases the efficacy of the therapy [33]. Traditional classification techniques are based on different kinds of clinical markers such as the morphological appearance of tumors, age of the patients and the number of lymph nodes [34]. These techniques however have extremely low (9%) prediction accuracies [35].

Class prediction based on gene expression monitoring is a relatively recent technology with a promise of significantly better accuracy compared to the classical methods [33]. These algorithms often use microarray data [36] as input. Microarrays measure gene expression and are widely used due to their ability to capture the expression of thousands of genes in parallel. A typical microarray database contains gene expression profiles of a few hundred patients. For each patient (also called observation), the microarray records expressions of more than 20,000 genes. We define an entry of a microarray as a feature.

Classification methods often build a classification function from a training data. The class labels of all the samples in the training data are known in advance. Given new sample, the classification function assigns one of the possible classes to that sample. However, as the number of features is large and the number of observations is small, standard classification algorithms do not work well on microarray data. One potential solution to this problem is to select a small set of relevant features from all microarray features and use only them to classify the data.

The research on microarray feature selection can be divided into three main categories: filter, wrapper and embedded [1]. These methods often employ statistical scoring techniques to select a subset of features. Selection of a feature from a large number of potential candidates is however difficult as many candidate features have
Figure 3-1. Part of pancreatic cancer pathway adapted from KEGG showing the gene-gene interactions. $\rightarrow$ implies activation and $\leftarrow$ implies inhibition. The rectangles with solid line represent valid genes mapped to the pathway. They are referred to by the name of the genes. $+p$ denotes phosphorylation. For example PKB/Akt activates IKK through phosphorylation. IKK in turn activates NFkB. Thus, PKB/Akt indirectly activates NFkB. The rectangles with dotted lines are genetic sequence that do not have Entrez Gene ID and not mappable to pathway. We can not yet associate them to some pathway. We denote them as unresolved genes. They are referred to by GenBank Accession numbers.

similar expressions. This potentially leads to inclusion of biologically redundant features. Furthermore, selection of redundant features may cause exclusion of biologically necessary features. Thus, the resultant set of features may have poor classification accuracy.

One way to select relevant features from microarray data is to exploit the interactions between these features, which is the problem considered in this chapter. More specifically we consider the following problem.

Problem statement. Let $D$ be the training microarray dataset where each sample belongs to one of the $T$ possible classes. Let $P$ be the gene regulatory and the signaling network. Choose $K$ features using $D$ and $P$ so that these features maximize the classification accuracy for an unobserved microarray sample that has the same distribution of values as those in $D$. 

25
**Contributions.** Unlike most of the traditional feature selection methods, we integrate gene regulatory and signaling pathways with microarray data to select biologically relevant features. On the pathway, one gene can interact with another in various ways, such as by activating or inhibiting it. In Figure 3-1, RacGEF activates RAC, BAD inhibits Bcl-xl and PKB/Akt inhibits BAD by phosphorylation. We use the term *influence* to imply this interaction between two genes. We quantify influence by considering the number of intermediate genes between two genes on the pathway that connects them. The influence is highest when two genes are directly connected. *Our hypothesis in this chapter is that selecting two genes that highly influence each other often implies inclusion of biologically redundant genes.* The rationale behind this is that manipulating one of these genes will have significant impact on the other one. Thus, selecting one of them produces comparable prediction accuracy. So we choose the set of features such that each of them has lowest influence on other selected features.

We propose a novel algorithm called *Biological Pathway based Feature Selection* algorithm (BPFS) based on the above hypothesis that has the following characteristics:

1. Let the complete set of features be $G$ and the set of already selected features be $S$. BPFS ranks all the features in $G - S$ with an SVM based scoring method MIFS [37]. The score quantifies the capacity of a feature to improve the already attained classification accuracy. BPFS ranks features in decreasing order of their scores.

2. BPFS chooses a small subset $C$ of highly ranked features from $G - S$ and evaluates the influence of every feature in $C$ on the features in $S$. Finally, it selects the feature in $C$ that has the lowest influence on the features in $S$ and moves it to $S$ from $G - S$. BPFS repeats this step for a fixed number of iterations.

We observe that a significant fraction of the gene entries in the microarray do not have any corresponding gene in the pathway. We use the term *unresolved genes* to represent these genes. We propose a probabilistic model to estimate the influence of those genes on selected features.

We tested the performance of our method on five breast cancer data sets [38–42] to predict whether breast cancer for those patients relapsed before five years or not. Our
Algorithm Biological Pathway based Feature Selection Algorithm (BPFS) /* G and S denote the set of all features and the set of selected features respectively. Set \( G - S \) represents all the remaining features. */

1. Select the first feature from \( G \) that has highest mutual information.
2. \textit{Repeat} till there is more features to select.
   
   (a) Calculate marginal classification power for all the features in \( G - S \).
   
   (b) Select top \( t \) features with highest marginal classification power as candidate set \( C \).
   
   (c) Calculate Total Influence Factor (TIF) for all the features in \( C \).
   
   (d) Select the feature with lowest TIF and include it into \( S \).

 experiments show that our method achieves up to 18% and 8% better accuracy than the 70-gene prognostic signature [38] and I-RELIEF [34] respectively.

The organization of the rest of this chapter is as follows. Section 3.1 describes the proposed algorithm. Section 3.2 presents experimental results. Section 3.3, briefly, concludes the chapter.

3.1 Algorithm

This section describes our Biological Pathway based Feature Selection algorithm (BPFS) in detail. BPFS takes a labeled two class microarray data as input and selects a set of features. Algorithm 1 portrays a synopsis of BPFS. We discuss an overview of BPFS next.

We denote the set of all features by \( G \). Let \( S \) be the set of features selected so far. The set \( G - S \) represents all the remaining features. BPFS iteratively moves one feature in \( G - S \) to \( S \) using the following steps, till the required number of features are selected (along with their rank):

1. \textbf{DETERMINE THE} \( t \) \textbf{BEST CANDIDATES.} (2(a) to 2(b) in Algorithm 1) This step creates a candidate set of features from \( G - S \) by considering their classification accuracy alone. To do this, BPFS first sorts all the available features in decreasing order of their marginal classification power and chooses the top \( t \) (typically \( t = 10 \) in practice) of them as the candidate set for next step. We define the marginal classification power of a feature as its ability to improve the classification accuracy
when we include it into \( S \). Let us denote the set that contains these top \( t \) features by the variable \( C \).

2. **PICK THE BEST GENE USING PATHWAYS.** (2(c) to 2(d) in Algorithm 1) In this step, we use signaling and regulatory pathways to distinguish among the feature set \( C \) obtained in Step 1. Given a set of already selected features \( S \), BPFS aims to select the next most biologically relevant feature from \( C \). We define a metric to compare the features in \( C \) for this purpose. This metric estimates the total influence between a candidate feature and the set of selected features. We denote this total influence as the **Total Influence Factor (TIF)**. TIF is a measure of the potential interaction (activation, inhibition etc.) between a candidate gene and all the selected genes. A high value of TIF for a gene implies that, the gene is highly influenced by some or all of the already selected set of genes. We choose the gene in \( C \) that has lowest TIF. We then include it in \( S \). We elaborate this step in Section 3.1.3.

In the following subsections we discuss the above aspects of our algorithm in more detail. Section 3.1.1 defines how we select our first feature. Section 3.1.2 discusses the first round of selection procedures based on classification capability. Section 3.1.3 describes the use of pathways for feature selection. Section 3.1.4 presents a technique to utilize the training space efficiently in order to improve the quality of features.

### 3.1.1 Picking the First Feature: Where to Start?

BPFS incrementally selects one feature at a time based on the features that are already selected. The obvious question, then is, how do we select the first feature? There are many alternative ways to do this. One possibility is to get an initial feature using domain knowledge. This, however, is not feasible if no domain knowledge exists on the dataset.

We use mutual information to quantify the discriminating power of a feature. Let us represent the \( k \)th feature of microarray using a random variable \( F \) and the class label of the data using another random variable \( L \). Assume that there are \( n \) observations in the data. \( F \) and \( L \) can assume different values over those \( n \) observations. Let \( f \) be an instance of \( F \) and \( l \) be an instance of \( L \). The mutual information of \( F \) and \( L \) is

\[
I(F, L) = \sum_{f \in F, l \in L} \psi_{F,L}(f, l) \log \frac{\psi_{F,L}(f, l)}{\psi_F(f) \psi_L(l)}
\]

where \( \psi_{F,L} \) is the joint probability mass function of \( F \) and \( L \); \( \psi_F \) and \( \psi_L \) are the respective marginal probability mass function of \( F \) and \( L \).
Thus, we use $I(F, L)$ to quantify the relevance of the $k$th feature for classification. We choose the feature with maximum mutual information as the starting feature.

Another way to select the first feature can be to utilize the marginal classification power. Essentially, it is a way we can apply the second step of our algorithm which $S=\{\}$ and select the top candidate as the first candidate.

Next we discuss how we select the remaining features.

3.1.2 Selecting the Candidate Features

In this step, BPFS sorts all the available features in $G - S$ in decreasing order of their marginal classification power. We define the marginal classification power of a feature later in this section. BPFS then chooses the $r$ features with the highest marginal classification power as the candidate set that will be explored more carefully in the subsequent steps. We elaborate on this next.

We use an SVM based algorithm, MIFS [37], to calculate the marginal classification power of all available features as follows. BPFS, first, trains SVM using the features in $S$ to get the value of the objective function of SVM. We use linear kernel for the SVM in our experiments. For very high dimensional data a linear kernel performs better than or comparable to a non-linear kernel [43]. A linear kernel is a simple dot product of the two inputs. So the objective function of SVM becomes

$$J = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

(3–1)

where $\alpha_i$, $y_i$, and $x_i$ denote the Lagrange multiplier, the class label and the value of selected set of features of $i$th observation respectively. Here, $x_i \cdot x_j$ are vectors and $x_i \cdot x_j$ is the dot product of them. Then for each feature $m \in G - S$, BPFS calculates the objective function $J$ if $m$ is added to $S$ as

$$J(S \cup \{m\}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i (+m) \cdot x_j (+m)$$
Here \( x_i(+m) \) denotes the value of the selected set of features along with the aforementioned feature \( m \) for the \( i \)th observation. Using the last two equations, we calculate the marginal classification power of a feature \( m \), as the change in the objective function of SVM when \( m \) is included in \( S \). We denote this value with variable \( \Delta J \). To paraphrase, marginal classification power of a feature is the capability of a new feature to improve the classification accuracy of a set of selected features, when the new feature is added to the already selected set. Formally, we compute \( \Delta J(m) \) for all \( m \in G - S \) as \( \Delta J(m) = J(S \cup \{m\}) - J(S) \). BPFS sorts all the features \( m \in G - S \) in descending order of \( \Delta J(m) \). It considers the top \( t \) (\( t = 10 \) in our experiments) genes as possible candidates for the next round. Let \( C \) denote the set of these \( t \) genes. In the next steps, BPFS examines biological networks to find out the most biologically meaningful feature in \( C \).

### 3.1.3 Selecting the Best Candidate Gene

All the features in the candidate set \( C \) often have high marginal classification power. In this step, we distinguish the features in \( C \) by considering their interactions with the features in \( S \) (the set of features that are already selected). We hypothesize that if a feature in \( C \) is influenced by the features in \( S \) greatly, then that feature is redundant for \( S \) even if it has high marginal classification power. We discuss how we measure the influence of a feature on another one next.

Consider the entire pathway as a graph, where all the genes are vertices and there is an edge between two vertices if they interact with each other. In this chapter, we do not consider any specific pathway such as p53 signaling pathway, rather a consolidation of all the available human signaling and regulatory pathways. If we have had the knowledge about the pathways that are affected by that specific biological condition (such as cancer), we could select features only from those pathways. However, the available literature does not provide the comprehensive list of affected pathways most of the time. Thus, we create a consolidation of all the regulatory and signaling pathways.
There are different kinds of interactions such as activation and inhibition. If two genes do not have a common edge, but they are still connected by a path, it means that they interact indirectly through a chain of genes. For example, in Figure 3-1, RacGEF activates RAC and RAC activates NFkB. Thus, RacGEF indirectly activates NFkB through RAC. We, therefore, compute the distance between them as two. A higher number of edges on the path that connects two genes implies feeble influence.

An abnormally expressed gene does not necessarily imply that its neighbor will be abnormally expressed [44]. This is because the interaction between two genes is a probabilistic event. For example, in Figure 3-1, if RacGEF becomes aberrantly expressed, there is a probability that RAC is also expressed aberrantly. Let us denote this probability with the variable $h$. Similarly, if RAC is abnormally expressed, NFkB is abnormally expressed with a probability $h$. So, if RacGEF is over expressed, NFkB can be over expressed with a probability of $h^2$. This leads to the conclusion that as the number of hops increases the influence decreases exponentially. Thus, we use an exponential function to model this influence.

To quantify influence we define a metric, termed Influence Factor (IF), between two genes $g_i$ and $g_j$ as $\text{IF}(g_i, g_j) = \frac{1}{2^{|d(g_i,g_j)|-1}}$, provided $i \neq j$. Here $d(g_i, g_j)$ is the length of the shortest path that connects $g_i$ and $g_j$ on the pathway. To calculate total influence on a candidate gene asserted by a selected set of genes we calculate IF between the candidate gene and every gene in the selected set and sum it up. We call it the total influence factor (TIF) of a candidate gene $g$ with respect to already set of selected genes. Formally,

$$TIF(g, S) = \sum_{s \in S} \frac{1}{2^{|d(g,s)|-1}}$$

$d(g, s)$ is zero if there is no path between $g$ and $s$.

For example, in Figure 3-1 consider PKB/Akt as a candidate gene and assume that $S$ consists of two genes NFkB and CASP9. CASP9 is one hop away from PKB/Akt.
So IF(PKB/Akt, CASP9) = 1. The shortest path from PKB/Akt to NFkB is of two hops through IKK. So IF(PKB/Akt, NFkB) = 0.5. Thus, TIF(PKB/Akt, \{CASP9, NFkB\}) = 1.5.

BPFS calculates TIF for all the genes in \(C\) and selects the one that generates the lowest value of TIF. A low TIF value implies lower aggregate influence on the set of selected features \(S\). To paraphrase, the gene with lowest TIF is least interacting with the genes in \(S\). So, we select the gene that is biologically most independent from \(S\).

The gene databases, like KEGG, are still evolving. Thus, many of the genes cannot be mapped from microarray data to these database. In Section 3.2 we describe a probabilistic technique to handle this problem.

3.1.4 Exploration of Training Space

We have described the key components of our feature selection algorithm (BPFS) in the previous subsections. As a dataset consists of comparatively smaller number of observations and a large feature set, BPFS is prone to overfitting. To avoid this problem, we propose a method that utilizes the training space efficiently.

Let \(D_T\) be the training data. We create \(K\) data subsets (\(K = 50\) in our experiments) \(D_{T_1}, D_{T_2}, \cdots, D_{T_K}\) each containing 80% of the \(D_T\) randomly sampled from it. We, then, run BPFS on each of them and get \(K\) sets of features. We store these \(K\) feature sets in a \(K \times N\) matrix \(M\), where the \(i\)th row contains first \(N\) features obtained from \(D_{T_i}\). Thus, \(m_{ij}\) is the \(j\)th feature obtained from \(D_{T_i}\). We use this matrix to rank all the features in the following fashion:

1. We assign a linearly decreasing weight across a row to emphasize the importance of the features that come first. More specifically, we assign a weight of \(N - k\) to a feature that appears in the \(k\)th column of a row.

2. We sum the weights of the features over all the rows to determine the overall weight of the features in \(M\). For example, assume that, a feature appears in three rows of \(M\), at \((5,3), (17,14)\) and \((29,10)\) where the first number in each pair indicates the row and second number indicates the column. Also, assume that we want to choose a total of 150 features. Then, the total weight of this feature is \((150-3) + (150-14) + (150-10) = 423.\)
We pick the \( N \) features with the highest weight from our feature set. Weighing the features based on their positions helps us to prioritize the features that occur frequently and/or appear with high rank. We discuss the impact of the value of \( N \) in Section 3.2.

### 3.2 Data Set and Experiments

In this section we evaluate BPFS experimentally. We use multiple real microarray datasets instead of synthetically generated data, as synthetic data may not accurately model different aspects of a real microarray data [45]. We observe that we can map only a small portion (25%) of the microarray entries to KEGG regulatory pathway. Some of them do not take part in any single interaction. So the only information we have about them is their measured expression value on the microarray dataset. Due to this missing data problem it is difficult to quantify the implication of biological pathway in our algorithm. To handle this problem we have conducted our experiments on two different kind of information. In one case, we use the KEGG pathways as it is and used a randomized technique to handle the interactions with unresolved genes. In the other case, we map all the microarray genes to KEGG pathway and assume that genes within a single pathway are fully connected and there is no common gene between two pathways. Still, we need to be careful while interpreting the results with fully connected pathway as it is only a simplistic view of the actual pathway. We cover the experiments with real pathways in the chapter from Section 3.2.2 - 3.2.6. In Section 3.2.7 we discuss the experiments with fully connected pathways.

In Section 3.2.1 we describe the experimental setup. In Section 3.2.2 we describe the randomization technique. We show the biological validity of our feature set, by tabulating the supporting publications against every feature (Section 3.2.3). We compare our signature against van 't Veer’s [38] on four data sets (Section 3.2.4). We compare the testing accuracy of our method to that of I-RELIEF, a leading microarray feature selection method (Section 3.2.5). We conducted cross validation experiments where we extracted features from one dataset and tested its accuracy on another dataset in
Section 3.2.6. Finally, we executed BPFS and I-RELIEF on an idealistic fully connected pathway in Section 3.2.7.

3.2.1 Experimental Setup

**Microarray data.** In our experiments we used five breast cancer microarray datasets from literature. We name these datasets as BCR [41], JNCI [39], Lancet [40], CCR [42] and Nature [38] respectively according to the name of the journals they were published. BCR, CCR and Lancet use Affymetrix® GeneChip Human Genome U133 Array Set HG-U133A consisting of 24,481 entries. Nature has its own microarray platform with 24,481 entries. JNCI has the same platform as that of Nature, but it consists of a much smaller feature set of 1,145 entries. We removed the observations whose class labels were not defined. For the rest of the data points we created two classes depending on whether relapse of the disease happened in five years or not, counting from the time of the primary disease. The datasets Nature, JNCI, BCR, CCR and Lancet contain 97, 291, 159, 190 and 276 observations respectively.

**Pathway data.** We used the gene regulatory and signaling pathways of Homo sapience in KEGG. We merged all the relevant pathway files to build a consolidated view of the entire pathway. The final pathway consists 8,270 genes and 7,628 interactions. Clearly, some genes do not take part in any interaction.

**Training and testing data.** We randomly divided a microarray dataset (e.g. BCR dataset) in 4:1 ratio to create training and testing subset. We maintained the distribution of two classes in the undivided dataset unchanged in the training and testing subset. We collected features from the training dataset and tested the classification accuracy using those features on the test dataset. Now we elaborate on how we utilized the training space to select features. We created a number of subsets $K$ (typically 50 in our experiments) using bootstrapping from training dataset. Each subset contains 80% samples of the training data. We selected features from each of those subspaces using
Figure 3-2. We plot the real data points $R_{P_{10}}$, $R_{P_{20}}$, $R_{P_{30}}$ against $s$, the fraction of the current pathway. We extrapolate $f(s)$ up to $s = 500$. $f(s)$ converges around 180.

Section 3.1.1 to Section 3.1.3. Then we combined the $K$ obtained set of features using the method of Section 3.1.4.

**Implementation and system details.** We implemented our feature selection algorithm (BPFS) using MATLAB®. For SVM, we used “MATLAB® SVM Toolbox”, a fast SVM implementation in C based on sequential minimal optimization algorithm [46]. For pathway analysis code we used Java™. We ran our implementation on a cluster of ten Intel Xeon 2.8 GHz nodes on Ubuntu Linux.

**Availability of code.** The implementation of the proposed method can be downloaded from http://bioinformatics.cise.ufl.edu/microp.html.

### 3.2.2 Pathways with Unresolved Genes

To calculate the influence factor (IF) we need to calculate the number of hops between two genes on the pathway. This requires a mapping of those microarray entries to pathway genes. However, as some of the microarray entries are not complete
genes and biological pathway construction is not yet finished [79], we can not map all microarray entries. We denote all the unmapped genes as *unresolved genes*. For example, Affymetrix® microarray HG-U133A contains 24,481 entries. We are able to map only 6,500 entries to KEGG (Kyoto Encyclopedia of Genes and Genomes). For example, in Figure 3-1 we draw four rectangles with dotted lines that correspond to four microarray entries in Affymetrix® platform. As they do not have Entrez Gene identification number, we can not associate them with any pathway. Hence, these are unresolved genes.

### Table 3-1. List of publications supporting the first twenty features obtained from BCR data set about their responsibility for cancer.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Supporting Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCNK2</td>
<td>Acute Lymphoblastic Leukemia [47]</td>
</tr>
<tr>
<td>ZNF222</td>
<td>Breast cancer [48]</td>
</tr>
<tr>
<td>P2RY2</td>
<td>Human lung epithelial tumor [49], Non-melanoma skin cancer [50], Thyroid cancer [51]</td>
</tr>
<tr>
<td>SLC2A6</td>
<td>Human Leukemia [52]</td>
</tr>
<tr>
<td>CD163</td>
<td>Breast cancer [53], Human colorectal cancer [54]</td>
</tr>
<tr>
<td>HOXC13</td>
<td>Acute myeloid leukemia [55, 56]</td>
</tr>
<tr>
<td>PCSK6</td>
<td>Breast cancer [57], Ovarian cancer [58]</td>
</tr>
<tr>
<td>AQP9</td>
<td>Leukemia [59]</td>
</tr>
<tr>
<td>PYY</td>
<td>Peptide YY (PYY) is a naturally occurring gut hormone with mostly inhibitory actions on multiple tissue targets [60]</td>
</tr>
<tr>
<td>KLRC4</td>
<td>KLRC4 is a member of the NKG2 group that are expressed primarily in natural killer (NK) cells [61]</td>
</tr>
<tr>
<td>CYP2A13</td>
<td>Lung adenocarcinoma [62]</td>
</tr>
<tr>
<td>GRM2</td>
<td>Metastatic Colorectal Carcinoma [63]</td>
</tr>
<tr>
<td>PHOX2B</td>
<td>Neuroblastoma [64]</td>
</tr>
<tr>
<td>ASCL1</td>
<td>Prostate cancer [65], Lung cancer [66]</td>
</tr>
<tr>
<td>PKD1</td>
<td>Polycystin-1 induced apoptosis and cell cycle arrest in G0/G1 phase in cancer cells [67], PKD1 inhibits cancer cells migration and invasion via Wnt signaling pathway in vitro [68]. Gastric cancer [69]</td>
</tr>
<tr>
<td>ANGPT4</td>
<td>Gastrointestinal stromal tumor, leiomyoma and schwannoma [70], renal epithelial and clear cell carcinoma [71]</td>
</tr>
<tr>
<td>PSMB1</td>
<td>Breast cancer [72]</td>
</tr>
<tr>
<td>RUNX1</td>
<td>Gastric cancer [73, 74], Ovarian cancer [75], Classical tumor suppressor gene [76]</td>
</tr>
<tr>
<td>CD1C</td>
<td>Prostate cancer [77]</td>
</tr>
<tr>
<td>ZNF557</td>
<td>Myeloid Leukemia [78]</td>
</tr>
</tbody>
</table>
Our preliminary experiments suggest that unresolved genes represent a large fraction of the genes in set \( C \) (more than 60% on average). To estimate the TIF of the unresolved genes, we develop a probabilistic model. Let \( C \) be the set of candidate genes and \( S \) be the set of selected genes in an iteration of BPFS. While calculating TIF for a \( g \in C \) we consider two cases:

1. **THE CANDIDATE GENE IS RESOLVED.** Assume that \( g \) is resolved. Let \( Q \subseteq S \) be the set that contains all the unresolved features in \( S \). Let \( R = S - Q \) be the set of resolved features in \( S \). Let \( p \) be the expected influence of a gene \( q \in Q \) on gene \( g \) if all genes were mapped and the pathway construction was complete. We discuss how we estimate the value of \( p \) later in this section. Then the expected number of genes from \( Q \) having influence on \( g \) is \( TIF(g, Q) = p \cdot |Q| \), where \( |Q| \) is the number of genes in \( Q \). So the Total Influence Factor becomes:

\[
TIF(g, S) = TIF(g, Q) + TIF(g, R) = p|Q| + \sum_{s \in R} 2^{1 - d(g, s)} \tag{3–3}
\]

2. **THE CANDIDATE GENE IS UNRESOLVED.** When \( g \) is unresolved we consider it as a special case of Case 1. As the connectivity between \( g \) and all genes in \( S \) is unknown, we estimate TIF as:

\[
TIF(g, S) = p|S| \tag{3–4}
\]

In summary, to handle the unresolved genes we augment the probabilistic model to biological pathway based selection and replaced equation (3–2) by equation (3–3). Among the genes in candidate set \( C \) we select the gene with smallest TIF using Equation 3–3. Now, we describe how we derive the value of \( p \) in Equation 3–3 and 3–4.

To derive the value of \( p \) we propose the following approach. It is reasonable to assume that there are many missing interactions in the currently available pathway databases, since missing interactions are continually being discovered. Let us denote the present incomplete pathway graph by \( P_i \) and the hypothetical complete pathway by \( P_C \). Assume that \( P_C \) contains \( z \) times more interactions than that of \( P_i \). From \( P_i \) we estimate \( p \) as a function of \( z \) and the expected number of the genes in \( P_i \) as follows. We, first, build a graph \( P_i \) as described in section 3.3 from the KEGG database. We,
then, randomly delete edges of $P_i$, and create the subgraphs $P_{10}, P_{20}, \ldots, P_{100}$ of $P_i$ corresponding to 10%, 20%, ..., 100% of edges of that of $P_i$. For each of these sub-graphs we calculate the average number of vertices reachable from a vertex.

Formally, let $V$ denote the set of vertices in $P_s$, where $s \in \{10, 20, \ldots, 100\}$. We denote the number of reachable vertices from $g$ ($g \in V$) by $R(g)$. Then, the average reachability of $P_s$ is:

$$R_{P_s} = \frac{\sum_{g \in V} R(g)}{|V|} \quad (3-5)$$

where $|V|$ is number of vertices in $V$. We calculate $R_{P_{10}}, \ldots, R_{P_{100}}$, the average reachability for all the subgraphs.

We, then, construct the function $f(s)$ that evaluates to $f(s) = R_{P_s}$. To construct $f(s)$, we use a converging power series. We derive the value of parameters using $R_{P_{10}}, \ldots, R_{P_{100}}$. To calculate the average reachability of the hypothetical pathway $P_C$ we use value of $s$ greater than 100 in $f(s)$. In Figure 3-2 we plot $R_{P_{10}}, \ldots, R_{P_{100}}$ along with the constructed function $f(s)$ against the $s$, the fraction of the current pathway.

We observe that $f(s)$ interpolates the values $R_{P_{10}}, \ldots, R_{P_{100}}$ accurately and it converges at around $s = 500$ with the value of 180. Thus, we can conclude that the average reachability of $P_C$ is around 180. The probability that an unresolved gene has an interaction with a randomly chosen gene in $S$ in $P_C$ is given by $p = \frac{R_{P_C}}{|V|}$ where $|V|$ is number of nodes in the pathway. As the total number of human genes is close to 20,500 [80], we get 0.0088 as the value of $p$.

### 3.2.3 Biological Validation of Selected Features

We collected the list of publications that support the relevance of the first twenty features selected from BCR on cancer. Table 3-1 lists the publications and cancer types for each of the genes.

To get the features, we created the training dataset as described in Section 3.2.1. We, then, trained BPFS on the training data and obtained a ranked set of features. We repeated this process ten times on BCR. We selected first twenty features from each
Figure 3-3. Comparison of test accuracy of our signature and van’t Veer’s 70-gene prognostic signature on real pathway. For all the datasets our signature performs significantly better than their signature.

rankings and merged them using the method described in Section 3.1.4 to get the final twenty features. We found relevant publication for all the twenty genes. We observed that four of them are directly responsible for breast cancer. Another seven genes are associated to breast cancer from the point of histology (two prostrate, four gastric and one colorectal). Rest of them are related to other kinds of cancer such as ovarian cancer and lung cancer.

In some cases a gene is involved for more than one kind of cancer. For example, ASCL1 is associated with both prostrate cancer and lung cancer. We concluded that BPFS chooses the set of genes that are responsible for breast cancer and other kind
of cancers. Hence, BPFS selects a biologically meaningful feature set that reduces the number of redundant features and improves generalization accuracy by selecting more relevant feature set.

In general, the above approach of combining features obtained from ten different runs may lead to selecting features from the entire dataset. We did it only for this experiment to filter out the biologically significant features from a dataset. For the remaining experiments we kept separate training and testing datasets.

3.2.4 Comparison with Van ’t Veer’s Gene Signature

In this section we compare our gene signatures to the breast cancer prognostic signature found by van ’t Veer et al. [38]. van ’t Veer et al. generated the 70 gene signature using a correlation based classifier on 98 primary breast cancer patients. van ’t Veer’s 70 gene experiment [38] demonstrated that genetic signature can have a much higher accuracy in predicting disease relapse free survival against clinical markers (50% vs 10%).
Figure 3-5. Comparison of test accuracy of our method (BPFS) to that of I-RELIEF on real pathway. In three datasets JNCI, BCR and Nature our method performs better than I-RELIEF. In Lancet and BCR both the methods have similar accuracy.
In our experiment, we demonstrate that our method finds a better gene signature than these 70-gene signature for a particular dataset. We created training and testing data from the four datasets JNCI, Lancet, BCR and CCR as described in Section 3.2.1. From those four training dataset we created four set of features using our method. We calculated the accuracy of the four set of features using the corresponding four test datasets. Also, we computed the test accuracy using this 70-gene signature on the same four test dataset just mentioned. Finally we plotted the testing accuracies obtained from both set of features in Figure 3-3.

Figure 3-3 illustrates the results for up to 150 genes for both the signature on BCR, CCR and Lancet. We observe that for all four data sets our accuracy is better than that of 70-gene signature. BPFS attains 18% better accuracy for JNCI dataset. From this we conclude that BPFS finds a better gene signature for all the datasets.

3.2.5 Comparison with I-RELIEF

We compare the accuracy of our method to that of I-RELIEF [34]. I-RELIEF is a nonlinear feature selection algorithm. It produced significant accuracy over van’t Veer’s 70 gene prognostic signature and standard clinical markers [81].

We, first, created training and testing dataset from the given data as discussed in Section 3.2.1. We obtained the ordered feature set by training the BPFS on the training data. We tested the quality of those features using the SVM classifier. We used identical set up and data sets for I-RELIEF. We used 2.0 as the kernel width for I-RELIEF as recommended [34]. We repeated the experiments ten times on each data set and present the average accuracy for different features. Figure 3-4 plots the standard deviation of the accuracy of our method over this 10 times of running. For all of them except the Nature dataset, the standard deviation is less than .1 (i.e. 10% accuracy). So we can conclude that our method is quite stable while we execute it over several subsets of data created from a single dataset.
Figure 3-5 compares our algorithm to I-RELIEF. We observe that for two data sets (BCR and JNCI), BPFS outperforms I-RELIEF for all the features selected. BPFS shows highest improvement (8%) over I-RELIEF for JNCI dataset, at around 50 features. JNCI has higher fraction of resolved genes (45% vs 25%). Thus, BPFS has a higher chance to select more resolved genes. This implies less dependability on the probabilistic model, the selection of genes is more accurate. From this observation, we expect that BPFS would produce better result when the missing links of the pathway would be discovered. For Nature dataset, BPFS produces better accuracy than I-RELIEF up to 130 features. BPFS has similar accuracy with that of I-RELIEF for CCR and Lancet data.

Our algorithm is based on linear kernel which is in general more appropriate when the number of features is much higher than number of samples. On the other hand I-RELIEF employs a non-linear kernel [43]. It is possible that the distribution of Lancet data works better with the type of kernel that I-RELIEF uses. We can potentially improve the classification accuracy of BPFS by using a non-linear kernel.

We observe that for all the datasets our method reaches its highest accuracy at around 50-70 features. We conclude that these 50-70 features consist the most important set of genes that are associated with the breast cancer.

3.2.6 Cross Validation Experiments

We conduct several cross validation experiments where we generate a set of features on one data set and validate its quality by testing it on some other data set. For this cross validation, we limit ourselves to the same microarray platform that we use to generate the feature set. For example, we test BCR dataset’s features on Lancet as the microarray platforms on which they were generated are same. The main reason for doing so is that the set of genes used in two different microarrays can be different. Even for the same gene they use different part of the genomic sequence. Thus, inter-platform validation may not be representative of the actual generalization. Table 3-2 displays the result of our cross validation experiments.
Table 3-2. Accuracy of our algorithm in terms of percentage obtained from Cross Validation experiments on real pathway. Feature set obtained from one data set is tested against another data set. We always chose target data from the same class of microarray in order to avert cross platform problems. The cross validation results implies that the feature set generated by BPFS provides satisfactory performance in cross data sets without significant loss of accuracy. We also compare our method with a trimmed version where we skip step 3.3. The complete version of the algorithm (with pathway) is indicated by 1 at the superscript while the trimmed version (without pathway) is denoted by the superscript 2.

<table>
<thead>
<tr>
<th>Test Data</th>
<th>Feature Data</th>
<th>Number of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5 10 20 40 60 80 100 120 140</td>
</tr>
<tr>
<td>BCR</td>
<td>BCR(^1)</td>
<td>65 65 72 72 73 75 74 74 73</td>
</tr>
<tr>
<td></td>
<td>CCR(^1)</td>
<td>63 61 61 63 66 68 66 71 69</td>
</tr>
<tr>
<td></td>
<td>CCR(^2)</td>
<td>64 62 65 63 64 65 66 65 68</td>
</tr>
<tr>
<td></td>
<td>Lancet(^1)</td>
<td>57 59 58 63 61 61 65 65 67</td>
</tr>
<tr>
<td></td>
<td>Lancet(^2)</td>
<td>55 55 63 62 58 64 66 69 70</td>
</tr>
<tr>
<td></td>
<td>70-Gene-Sig(^1)</td>
<td>62 59 59 66 67 68 70 73 71</td>
</tr>
<tr>
<td></td>
<td>70-Gene-Sig(^2)</td>
<td>55 58 61 62 58 64 66 69 70</td>
</tr>
<tr>
<td></td>
<td>70-Gene-Sig(^1)</td>
<td>62 59 59 66 67 68 70 73 71</td>
</tr>
<tr>
<td></td>
<td>Nature(^1)</td>
<td>68 65 65 65 72 70 72 70 66</td>
</tr>
<tr>
<td></td>
<td>JNCI(^1)</td>
<td>71 71 65 61 59 64 60 57 61</td>
</tr>
<tr>
<td></td>
<td>JNCI(^2)</td>
<td>67 70 63 60 66 74 73 74 72</td>
</tr>
</tbody>
</table>
Figure 3-6. Comparison of test accuracy of our method (BPFS) to that of I-RELIEF on fully connected pathway. In three datasets JNCI, BCR and CRR our method performs better than I-RELIEF. In Lancet and BCR I-Relief has a better accuracy.
Figure 3-7. Comparison of test accuracy of our method (BPFS) to when we select the genes only based on marginal classification power.
We use two different versions of our algorithm, one with the pathway information and another without the pathway information on observe the contribution of inclusion of the pathway information into our algorithm. In Table 3-2 we denote the the version of our method with pathway by putting 1 at superscript of the result. Similarly we denote the version without pathway by putting 2 at the superscript of the datasets.

Also, to establish the relevance of our signature on a standard benchmark, we compare our signatures with van’ t Veer’s 70-gene signature in the context of cross validation. For example, in Table 3-2 BCR dataset is cross validated with three feature sets obtained from Lancet, CCR and 70-gene signature. We observe that on an average our signatures perform better than the 70-gene one. For CCR data, both Lancet and BCR features generate better accuracy. For Lancet data the accuracies obtained using CCR and BCR features are similar to that of 70-gene signature. For BCR data, up to 80 features our signatures outperform the 70-gene signature. Beyond that the 70-gene signature has a better accuracy. To sum up, we get better accuracy while testing with the features on a different platform compared to van’ t Veer’s prognostic signature.

Regarding the comparison of the two versions of our algorithm (with and without the pathway information) it’s difficult to reach a conclusive decision. For example, while we extract the feature set from CCR dataset and cross validate it BCR dataset the algorithm without pathway information is doing better than the other for upto 20 features, but for the larger number of features the algorithm with pathway information provides a better accuracy.

When we compare the accuracy with features from a different dataset to that of its own feature set, we observe that on the average, the drop of the accuracies are not more than 6%. For some extreme cases the drop can be higher. Specifically when we cross validate with features extracted from Lancet, the accuracy is lower.
3.2.7 Experiments on Fully Connected Pathways

In this section we describe the experiments with idealistic fully connected pathway. Here the approach we took was to evaluate our experiments on an idealistic pathway, where we map all the genes including the unresolved genes into the KEGG pathway. The unmapped genes become singleton pathway with only a single member gene. For other pathways that are already listed in the KEGG database we assume that all the genes within a pathway are fully connected and there is no connection between two pathway. If we consider each pathway as the smallest indivisible module of interactions, then all the genes within a pathway coexpress in a similar fashion. We compare the accuracy of BPFS with this pathway with that of I-RELIEF. In Figure 3-6 we see that for BCR, JNCI and CCR environment, BPFS has a better accuracy. For BCR there is an improvement of 5% around when BPFS uses around 40 features. For JNCI, the improvement is over 10% with around 70 features. For CCR there is a 3% improvement for 30 features. For Nature dataset, I-RELIEF has slightly (around 1.5%) better accuracy up to 100 features. For Lancet dataset I-RELIEF performs better than BPFS. So, we observe that for three dataset BPFS has better accuracy, for one dataset the accuracy is almost comparable.

3.2.8 Contribution of Pathway Information

In this section we describe the experiments that we conduct to understand the contribution of pathway information in our algorithm in terms of classification accuracy. In one version we use the complete version of the algorithm as it is, in the other version we select the features only based on the marginal classification power and skip the next step.

However, we observe that the contribution of biological network is not very decisive. For some dataset and some set of features the complete version of the method generates better accuracy, sometimes the trimmed version produces more accurate result. For example, for BCR dataset the complete method has upto 6% higher accuracy,
where for Nature dataset the method with pathway generates better result for 50-100 number of features. The reason behind this fluctuation of accuracy might be that reconstruction of gene regulatory and signaling network is still in progress. Among almost 22,000 Affymetrix® transcripts, we could map only 3,300 genes that take part into at least one KEGG pathway. Even for those genes, the pathway construction is not complete. We hope that our algorithm can generate better accuracy in the near future when we can have a more comprehensive pathway structure.

3.3 Discussion

In this chapter we considered feature selection problem for a classifier on cancer microarray data. Instead of using the expression level of a gene as the sole feature selection criteria we also considered its relation with other genes on the biological pathway. Our objectives were to develop an algorithm for finding a set of biologically relevant features and to reduce the number of redundant genes. The key contributions of the chapter are:

- We proposed a new feature selection method that leverages biological pathway information along with classification capabilities to reduce the redundancy in gene selection based on biological pathway.
- We proposed a probabilistic solution to handle the problem of *unresolved genes* that are currently not mappable from microarray to biological pathway.
- We presented a new framework of utilizing the training subspace that improve the quality of feature set.

Our algorithm improve quality of features by a biological way by excluding the features that have total influence factor, and includes genes that are apart in the biological network and still have high marginal classification power. Thus, we believe that instead of selecting a close set of genes as features our method identify biologically important key features for a significant number of pathways. We demonstrated the biological significance of our feature set by tabulating the relevant publications. We also established the quality of our feature set by cross validating them on other data sets and
comparing them against van’t Veer’s 70-gene prognostic signature. Our experiments showed that it is better than best published available method I-RELIEF.
A significant set of microarray experiments measure gene expressions in the presence of external perturbations [82, 83]. In these perturbation experiments, radiation [84], drug [14] or other biological conditions are administered on tissues and their responses are monitored using microarrays. The expressions of the genes before perturbations correspond to control data, while the expressions of genes after perturbations correspond to non-control data [85].

A fraction of genes respond to the external perturbation by changing their expression values significantly between control and non-control groups. Such genes are called differentially expressed (DE) genes [86]. The remaining genes, without noticeable changes in their expressions, are called equally expressed (EE) genes.

The DE genes that are directly affected by the external perturbation [19] are denoted as primarily affected genes. Rest of the DE genes change their expressions due to interactions with primarily affected genes and each other through signaling and regulatory networks [20, 87–89]. We call them as secondarily affected genes. In this chapter, the term gene networks is used to refer signaling and regulatory networks. Figure 4-1 shows the state of the genes in the pancreatic cancer pathway after a hypothetical external perturbation is applied [90, 91]. In this figure, genes K-Ras, Raf and Cob42Roc are primarily affected and MEK, Ral and RalGDS are secondarily affected through interactions.

We consider the problem of identifying the primarily affected genes in a perturbation experiment where gene expressions are provided before and after the application of perturbation for a set of samples. We assume that the underlying gene network can be modeled as a directed graph where each node represents a gene, and a directed edge from gene $a$ to gene $b$ represents a genetic interaction (e.g. $a$ activates or inhibits $b$). We define two genes as neighbors of each other if they share an edge. For example,
Figure 4-1. Illustration of the impact of a hypothetical external perturbation on a small portion of the pancreatic cancer pathway. The pathway is taken from the KEGG database. The solid rectangles denote the genes affected directly by the perturbation, the dashed rectangles indicate genes secondarily affected through interactions. The dotted rectangles denote the genes that are not affected by the perturbation. → implies activation and ⊣ implies inhibition. In this figure, gene K-Ras, Raf and Cob42Roc are primarily affected and MEK, Ral and RalGDS are secondarily affected through interactions.

in Figure 4-1, genes K-Ras and Raf are neighbors as K-Ras activates Raf. A neighbor can be classified as incoming or outgoing if it is at the start or at the end of the directed edge, respectively. In Figure 4-1, Raf is an incoming neighbor of MEK and MEK is an outgoing neighbor of Raf.

Contributions.

• We propose a new probabilistic method to find the primarily affected genes in microarray dataset. Our method employs gene networks as a prior belief in a Bayesian framework. When the expression level of a gene alters, it can affect some of its outgoing neighbors. Thus, the expression of a gene can change due to external perturbation or because of one or more of the affected incoming neighbors. We build our solution based on this observation.

• Let $G = \{g_1, g_2, \ldots, g_M\}$ denote the set of all genes. We represent the external perturbation by a hypothetical gene (i.e. metagene) $g_0$ in our gene network. An edge from the metagene to all the other genes implies that the external perturbation has the potential to affect all the other genes. So, $g_0$ is an incoming neighbor to all the other genes. We call the resulting network the extended gene network. Our method estimates the probability that a gene $g_j$ is DE due to an alteration in the activity of gene $g_i$ ($g_i \in G \cup \{g_0\}, g_j \in G$) if there is an edge from $g_i$ to $g_j$ in the extended network. We use a Bayesian model in our solution with
the help of Markov Random Field (MRF) to capture the dependency between the genes in the extended gene network. We optimize the pseudo-likelihood of the joint posterior distribution over the random variables in the MRF using Iterative Conditional Mode (ICM) [92]. The optimization provides us the state of the genes and the pairwise causality among the genes including the metagene.

Our experiments on both real and synthetic datasets demonstrate that our method can find primarily affected genes with high accuracy. We compared our method with SSEM and Student’s t test and obtained significant higher accuracy in finding out the primarily differentially expressed genes.

The rest of the chapter is organized as follows. In Section 4.1 we describe our method in detail. In Section 4.2 we discuss the experiments and results. Finally, in Section 4.3 we describe our key conclusions for this chapter.

4.1 Methods

In this section we describe our method. Section 4.1.1 presents the notations. Section 4.1.2 provides an overview of our solution. Section 4.1.3 discusses the modeling of the MRF based prior distribution. Section 4.1.4 describes how we formulate a tractable approximate version of the objective function. Section 4.1.5 discusses how we compute the likelihood of the expression of a gene. Section 4.1.6 explains how we optimize the objective function.

4.1.1 Notation and Problem Formulation

We start by describing the notation used in the rest of this chapter and provide a formal definition of the problem. We use two types of parameters to model this problem, namely observed and hidden. The values of observed variables are available in the given data set. We estimate the hidden variables from the observed data.

Observed variables. There are two sets of observed variables.

- MICROARRAY DATASET. We denote the number of microarray samples and the number of genes by \( N \) and \( M \) respectively. We represent the set of all genes in
the dataset with \( G = \{ g_1, g_2, \ldots, g_M \} \). For each gene \( g_i \), the dataset contains the expressions before and after the perturbation (i.e. control and non-control) respectively. We denote the expressions of \( g_i \) with \( y_{ij} \) and \( y'_{ij} \) in control and non-control group respectively, \( 1 \leq i \leq M, 1 \leq j \leq N \). Let \( y_i = \{ y_{i1}, y_{i2}, \ldots, y_{iN} \} \) and \( y'_i = \{ y'_{i1}, y'_{i2}, \ldots, y'_{iN} \} \) denote the expression values of \( g_i \) in control and non-control groups respectively.

\[ y_{ij}, y'_{ij} \]

\[ \text{NEIGHBORHOOD VARIABLES.} \] We use the term \( \mathcal{W} = \{ W_{ij} \} \) to represent if any two genes \( g_i \) and \( g_j \) are neighbors. \( W_{ij} (1 \leq i, j \leq M) \) is set to 1 if \( g_i \) is an incoming neighbor of \( g_j \) (i.e. \( g_j \) has an incoming edge from \( g_i \) in the extended gene network) and 0 otherwise.

\[ \mathcal{W} = \{ W_{ij} \} \]

\[ \text{Hidden Variables.} \] There are two sets of hidden variables.

\[ \mathcal{W} = \{ W_{ij} \} \]

\[ \text{STATE VARIABLES.} \] Each gene \( g_i \) can attain one of the two states (i.e. DE or EE). We introduce the variables \( S = \{ S_i \} \) to indicate the states of the genes. Formally, \( S_i \) is DE if \( g_i \) is differentially expressed and EE if \( g_i \) is equally expressed.

\[ S_i = \{ \text{DE, EE} \} \]

\[ \text{INTERACTION VARIABLES.} \] We define the set of random variables \( \mathcal{X} = \{ X_{ij} \} \) to represent the joint state of genes \( g_i \) and \( g_j \) \( (0 \leq i \leq M, 1 \leq j \leq M) \). Formally,

\[ X_{ij} = \begin{cases} 
1 & \text{if } S_i = \text{DE and } S_j = \text{DE}; \\
2 & \text{if } S_i = \text{DE and } S_j = \text{EE}; \\
3 & \text{if } S_i = \text{EE and } S_j = \text{DE}; \\
4 & \text{if } S_i = \text{EE and } S_j = \text{EE}; 
\end{cases} \]

It is evident that the value of \( X_{ij} \) depends on the values of two independent variables \( S_i \) and \( S_j \). Note that the values of \( X_{ij} \) are categorical in nature.

\[ X_{ij} \]

\[ \text{Problem formulation.} \] We have microarray expression data \( \mathcal{Y} \) and the gene network \( \{ G, \mathcal{W} \} \) as input to the problem. From now on, the gene network \( \{ G, \mathcal{W} \} \) will be referred to by \( \mathcal{V} \). We would like to estimate the posterior density \( f(\mathcal{X} \mid \mathcal{Y}, \mathcal{V}, W_{ij} = 1) \). Specifically, a lower value of \( f(\mathcal{X}_{0j} = 2 \mid \mathcal{X} \mid \mathcal{Y}, \mathcal{V}, W_{ij} = 1) \) indicates a higher chance that the gene \( g_j \) is primarily affected, as \( X_{0j} = 2 \) indicates that the metagene is DE and gene \( g_j \) is EE. Based on this probability estimation, we create a list of primarily affected genes.
4.1.2 Overview of Our Solution

An approach to solve our problem can be to maximize a likelihood distribution over the gene expression $\mathcal{Y}$ where $\mathcal{X}$ are the parameters of the distribution. The objective is to obtain the maximum likelihood estimate (MLE) of $\mathcal{X}$. However, there are two problems in this approach. First, MLE requires a large number of data points to accurately estimate the parameters. Second, MLE depends only on the observed data and cannot utilize domain specific knowledge; as a result it leads to overfitting of data and poor generalization.

We develop a Bayesian framework for estimating $\mathcal{X}$ that addresses the above-mentioned limitations of the existing approaches. Bayesian approaches can generally estimate the parameters with fewer data-points, which makes our approach more suitable for perturbation experiments [93].

We estimate the probability of $X_{ij}$ given the other observed and hidden variables. In this approach, we aim to maximize the joint density of the $\mathcal{X}$ variables given the gene expressions $\mathcal{Y}$ and the gene network $\mathcal{V}$. Thus, the objective to maximize is given by,

$$P(\mathcal{X}|\mathcal{Y}, \mathcal{V}, \theta_{\mathcal{Y}}, \theta_{\mathcal{X}}) = \frac{P(\mathcal{Y}|\mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}})P(\mathcal{X}|\mathcal{V}, \theta_{\mathcal{X}})}{\sum_{\mathcal{X}} P(\mathcal{Y}|\mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}})P(\mathcal{X}|\mathcal{V}, \theta_{\mathcal{X}})}$$  \hspace{1cm} (4–1)

$\theta_{\mathcal{Y}}$ is the set of parameters for the likelihood function $P(\mathcal{Y}|\mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}})$ and $\theta_{\mathcal{X}}$ is the set of parameters for the prior density function $P(\mathcal{X}|\mathcal{V}, \theta_{\mathcal{X}})$. $\theta_{\mathcal{X}}$ and $\theta_{\mathcal{Y}}$ will be discussed in Sections 4.1.3 and 4.1.5 respectively.

Since a direct optimization of Equation 4–1 is impractical due to exponential number of terms in the denominator, we define a more tractable objective function as discussed in Section 4.1.4. We use iterative conditional mode (ICM) to optimize the objective function and obtain an assignment of $\mathcal{X}$, $\theta_{\mathcal{X}}$ and $\theta_{\mathcal{Y}}$. Finally we estimate the posterior probability $p(X_{ij}|\mathcal{X}, \mathcal{V}, \theta_{\mathcal{X}}, \theta_{\mathcal{Y}})$ for every $X_{ij}$ when $W_{ij} = 1$. Using this posterior probability, we quantify the chance that one gene is DE due to one of its incoming neighbors.
4.1.3 Computation of the Prior Density Function

In this section, we describe how we build the prior density function \( P(\mathbf{X}|\theta_{\mathbf{X}}) \). We incorporate information from biological networks as prior belief in this density function. The following two assumptions encapsulate our belief about gene interactions.

- Each gene can affect the expressions of its outgoing neighbors. If the activity of a gene is altered, the effect can propagate to its outgoing neighbors.
- The metagene \( g_0 \) (i.e. external perturbation) can affect the expression of every other gene. This is easy to visualize as the external perturbation such as radiation can change the activity of any of the genes.
Clearly, when the data does not follow one or more of the hypotheses, the optimization function can overcome the prior belief with a strong support from the data.

In order to compute the prior density function, we define a Markov Random Field (MRF) over the $\mathcal{X}$ variables [94]. MRF is a probabilistic model, where the state of a variable depends only on the states of its neighbors. MRF is useful to model our problem as the states of genes depend on their neighbors. Here, the MRF is an undirected graph $\Psi = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X} = \{X_{ij}\}$ variables represent the vertices of the graph (i.e. each interaction variable $X_{ij}$ corresponds to a vertex). We denote the set of edges with $\mathcal{E} = \{(X_{ij}, X_{jk}) | W_{pi} = W_{pi} = 1\} \cup \{(X_{ij}, X_{ik}) | W_{jk} = W_{jk} = 1\}$. Thus, two variables in $\mathcal{X}$ share an edge if they share a common subscript at the same position and the two genes corresponding to the other subscript interact in the gene network (this is a necessary condition). For example, in Figure 4-2B, $X_{35}$ and $X_{25}$ are neighbors, as they share 5 (i.e. gene $g_5$) as the second subscript and $g_2$ and $g_3$ interact in the gene network in Figure 4-2A.

One important point to note is that, this graph does not use the state variables $\mathcal{S}$ to model the dependencies between the genes. Rather, it establishes those dependencies over the $\mathcal{X}$ variables. For example, in Figure 4-2B we draw the MRF graph corresponding to the hypothetical gene network in Figure 4-2A. In the gene network, there is an edge from $g_2$ to $g_3$. So, $g_2$ can potentially change the state of $g_3$. We create an edge from $X_{12}$ to $X_{13}$ that corresponds to the edge from $g_2$ to $g_3$. As $g_3$ is common for $X_{12}$ and $X_{13}$, if they assume the same value (i.e. $X_{12} = X_{13}$), it implies that the genes $g_2$ and $g_3$ are in the same state (i.e. $S_2 = S_3$). We formulate these dependency constraints using a set of unary and binary functions called feature functions. We discuss these feature functions next.

We denote the neighbors of $X_{ij}$ in the MRF graph as $X_{ij}^* = \{X_{ji} | W_{pi} = 1\} \cup \{X_{ik} | W_{jk} = 1\}$. We define a clique over each $X_{ij}$ and its neighbors $X_{ij}^*$ by $C_{ij}$ provided $W_{ij} = 1$. A feature function $f(C_{ij})$ is a Boolean function defined over the cliques $C_{ij}$ of $\Psi$. This
Table 4-1. The table enumerates the truth values for the two binary feature functions. Only the permitted entries are annotated with 0/1. The blank entries correspond to combinations that are not possible. (a) \( f_3(X_{ij}, X_{j\bar{i}}) \) represents the feature function for left equality. (b) \( f_4(X_{ij}, X_{ik}) \) represents the feature function for right equality.

<table>
<thead>
<tr>
<th>( X_{ij} )</th>
<th>( X_{j\bar{i}} )</th>
<th>( X_{ij} )</th>
<th>( X_{ik} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

(a) \( f_3(X_{ij}, X_{j\bar{i}}) \)  
(b) \( f_4(X_{ij}, X_{ik}) \)

function evaluates to 1 or 0, if it is satisfied or not, respectively. We define a potential function \( \psi(C_{ij}) \) corresponding to \( f(C_{ij}) \) as an exponential function given by \( \exp(\gamma f(C_{ij})) \). Here \( \gamma \) is a coefficient associated with \( f(C_{ij}) \) that represents the relevance of \( f(C_{ij}) \) in the MRF. According to Hammersley-Clifford theorem, we express the joint density function of the MRF over \( \mathcal{X} \) as product of potential functions defined over that MRF as,

\[
\prod_{C_{ij}} \prod_{C_{\bar{j}}} \psi(C_{ij}) \prod_{C_{ik}} \prod_{C_{\bar{i}}} \psi(C_{ik})
\]

In this formulation, \( \Delta \) is the normalization function \( \Delta = \sum_{\mathcal{X}} \prod_{C_{ij}} \prod_{C_{\bar{j}}} \psi(C_{ij}) \). To limit the complexity of our model, we consider only cliques of size one and two.

We define four feature functions to capture the dependencies among the variables in \( \mathcal{X} \) according to the two hypotheses. Based on the number of input variables, they are classified as unary and binary feature functions.

**Unary feature functions.** A primary component of the prior density function is modeling the frequency of \( X_{ij} \) itself. We capture this frequency using two unary feature functions defined over singleton cliques. We define a feature function \( F_1(X_{ij}) \) which returns one when \( X_{ij} = 1 \) and 0 otherwise. To capture the complemented events, we define another feature function \( F_2(X_{ij}) \), which returns to 1 when \( X_{ij} = 0 \) and returns 0 otherwise.
**Binary feature functions.** These feature functions are defined to incorporate the two assumptions stated at the beginning of this section. Consider a sequence of four genes $g_1, g_2, g_3$ and $g_5$ in Figure 4-2A. $X_{23}$ is a variable in the MRF graph that depends on the states of $g_2$ and $g_3$. $X_{13}$ is a neighbor of $X_{23}$ in MRF graph as $g_1$ is an incoming neighbor of $g_3$ in the gene network. Similarly, $X_{25}$ is a neighbor of $X_{23}$ as $g_5$ is an outgoing neighbor of $g_3$. If $S_2$ equals to $S_2$ then $X_{23} = X_{13}$. Similarly if $S_3$ equals to $S_5$ then $X_{23} = X_{25}$. We capture these events in two feature functions for $X_{ij}$ based on the incoming neighbors of $g_i$ and the outgoing neighbors of $g_j$.

- **LEFT EQUALITY.** Let us denote the incoming neighbors of $g_i$ with $ln(g_i)$. We write a feature function $f_3(X_{ij}, X_{pj}), \forall p, p \in ln(g_i)$. $f_3(X_{ij}, X_{pj}) = 1$ if $S_i = S_p$ and $W_{pi} = W_{ij} = 1$. Otherwise, $f_3(X_{ij}, X_{pj}) = 0$. We denote the summation of this function over all the incoming neighbors of $g_i$ as,

$$F_3(X_{ij}) = \sum_{p, W_{pi} = 1, W_{ij} = 1} f_3(X_{ij}, X_{pj}).$$

- **RIGHT EQUALITY.** Let us denote the outgoing neighbors of $g_j$ as $Out(g_j)$. We define a feature function $f_4(X_{ij}, X_{ik}), \forall k, g_k \in Out(g_j)$. $f_4(X_{ij}, X_{ik}) = 1$ if $S_k = S_j$ and $W_{ij} = W_{ik} = 1$. Otherwise, $f_4(X_{ij}, X_{ik}) = 0$. We denote the summation of this function over all the outgoing neighbors of $g_j$ as,

$$F_4(X_{ij}) = \sum_{k, W_{ij} = 1, W_{ik} = 1} f_4(X_{ij}, X_{ik}).$$

Table 4-1 enumerates the truth values of the binary feature functions for different values of their arguments. Only the permitted entries are annotated with zero and one. The other entries require illegal combination of argument values.

In the binary feature functions $X_{pj}$ or $X_{ik}$ may not represent any interactions from the extended gene network when $W_{pj} = 0$ or $W_{ik} = 0$, respectively. We represent them by rectangles in Figure 4-2B.

Based on these feature functions, we define the joint density function of $\mathcal{X}$ as,
\[ p(\mathbf{X} | \theta) = \frac{1}{\Delta} \exp \left( \sum_{i,j, W_j = 1} \gamma_k F_k(X_{ij}) \right) \] (4–2)

In the above equation \( \gamma_k, k \in \{1, 2, \ldots, 4\} \) are the coefficients of the four feature functions in MRF. In the next section, we discuss how we define the objective function with respect to the MRF. We also describe how we formulate the posterior probability density function for \( X_{ij} \).

### 4.1.4 Approximation of the Objective Function

A direct maximization of the objective function given by Equation 4–1 is impractical, as it requires evaluation of exponential number of terms in the denominator. We employ pseudo-likelihood as an established substitute to Equation 4–1 [96]. Pseudo-likelihood is the simple product of the conditional probability density function of the \( X_{ij} \) variables. Geman et al. proved the consistency of the maximum pseudo-likelihood estimate [97].

The approximated objective function can be written as,

\[ F = \arg \max_{\mathbf{X}} \left( \prod_{i,j} F_{ij} \right) \] (4–3)

We derive the posterior density function \( F_{ij} \) of \( X_{ij} \) when \( W_{ij} = 1 \) as,

\[ F_{ij} = \frac{p(X_{ij} | \mathbf{X} - X_{ij}, \mathbf{Y}, \theta_X, \theta_Y, W_{ij} = 1)}{\sum_{X_j \in \{1, \ldots, 4\}} p(Y_i, Y_j | X_{ij}, X_{ij}^*, \theta_Y, W_{ij} = 1) p(X_{ij} | \mathbf{X} - X_{ij}, \mathbf{Y}, \theta_X, W_{ij} = 1)} \] (4–4)

There are two different terms in objective function of Equation 4–4. \( p(X_{ij} | \mathbf{X} - X_{ij}, \mathbf{Y}, \theta_X, \theta_Y, W_{ij} = 1) \) stands for the conditional prior density function of \( X_{ij} \) which can be derived from
Equation 4–2 using Bayes rule. We discuss \( p(Y_i, Y_i^c | X_j, X_j^c, \theta, W_j = 1) \), the likelihood function in the next section.

### 4.1.5 Calculation of Likelihood Density Function

In this section, we describe how we derive the likelihood function. We assume that gene expressions in a group follow a normal distribution. We can redo the derivations if gene expressions follow some other distribution.

Consider a set of measurements for a gene \( g_i \) that follows a single Gaussian distribution by \( z_i = \{z_{i1}, z_{i2}, \cdots, z_{iN}\} \). We denote the latent mean of \( z_i \) by \( \mu \) and the standard deviation by \( \sigma \). As different genes can have different average expressions, we assume that \( \mu \) follows a genome-wise normal distribution with mean \( \mu_0 \) and standard deviation \( \tau \) [98]. Thus, for \( z_i \), the likelihood for the data points in that group is given by,

\[
L(z|\mu_0, \sigma^2, \tau^2) = \int \left[ \prod_{i=1}^{n} N(z_i|\mu, \sigma^2) \right] N(\mu|\mu_0, \tau^2) d\mu \\
= \frac{\sigma}{(\sqrt{2\pi}\sigma)^n \sqrt{n\tau^2 + \sigma^2}} \exp\left( -\frac{\sum z_i^2}{2\sigma^2} - \frac{\mu_0^2}{2\tau^2} \right),
\]

\[
\exp\left( \frac{\tau^2 n^2 \sigma^2 + \sigma^2 \mu_0^2}{2(n\tau^2 + \sigma^2)} \right)
\]

The reader can find the derivation of Equation 4–5 in Demichelis et al. [99].

If a gene is DE, its expression measurements in control and non-control groups follow separate distributions. On the other hand, for equally expressed genes, all the measurements in both the groups share the same mean. The joint data likelihood for a DE gene is given by,

\[
L(g_i) = \begin{cases} 
L(y_i|\mu_0, \sigma^2, \tau^2) L(y_i^c|\mu_0, \sigma^2, \tau^2), & \text{if } S_i = DE. \\
L(y_i \cup y_i^c|\mu_0, \sigma^2, \tau^2), & \text{if } S_i = EE
\end{cases}
\]

\[
(4–6)
\]
Now we are ready to derive the joint likelihood distribution for different values of $X_{ij}$. Let us denote the set of parameters $\{\mu, \sigma, \tau\}$ by $\theta_\gamma$.

We have four different forms for the likelihood of $(Y_i, Y_j)$ due to four different values it can assume. However, we shall derive only for $X_{ij} = 1$, as for the other values of $X_{ij}$ we have similar derivations.

\[
p(Y_i, Y_j | X_{ij} = 1, X_{ij}^*, \theta_\gamma, W_{ij} = 1)
= \sum_{\tau_i, \tau_j \in \{DE, EE\}} p(Y_i, Y_j | S_i = \tau_i, S_j = \tau_j, \theta_\gamma, W_{ij} = 1) 
= p(S_i = \tau_i, S_j = \tau_j | X_{ij} = 1, X_{ij}^*, \theta_\gamma, W_{ij} = 1)
\tag{4–7}
\]

From the definition of $X_{ij}$, $p(S_i = \tau_i, S_j = \tau_j | X_{ij} = 1, X_{ij}^*, \theta_\gamma)$ equals to 1 when $S_i = DE$ and $S_j = DE$. Its value is zero for all other values of $S_i$ and $S_j$. So, continuing from the last step of Equation 4–7,

\[
p(Y_i, Y_j | X_{ij} = 1, X_{ij}^*, \theta_\gamma, W_{ij} = 1)
= p(Y_i, Y_j, Y_{ij} | S_i = DE, S_j = DE, \theta_\gamma, W_{ij} = 1)
= p(Y_i | S_i = DE, S_j = DE, \theta_\gamma)
= p(Y_j | S_i = DE, \theta_\gamma) \cdot p(Y_j | S_j = DE, \theta_\gamma)
= \mathcal{L}(g_i) \mathcal{L}(g_j)
\]

In a similar way, we can derive the likelihood functions for the other three values of $X_{ij}$ variable. A special case arises when $g_i$ is the metagene, i.e. $g_0$. Specifically, $\mathcal{L}(g_0) = 1$ if $S_0 = DE$ and 0 otherwise, as, according to our assumption the metagene is always DE.
4.1.6 Objective Function Optimization

So far, we have described how we compute the posterior density function. The final challenge is to find the values of the hidden variables that maximize the objective function (Equation 4–3). We develop an iterative algorithm to address this challenge.

In our model we have three different sets of parameters. The nodes of the MRF given by \( \mathcal{X} \) consist of one set. Other two sets are the parameters of conditional probability density function of \( X \) and likelihood function of observed data given by \( \theta_X = \{\gamma_1, \cdots \gamma_4\} \) and \( \theta_Y = \{\mu_0, \sigma, \tau\} \), respectively. In each iteration, we first estimate \( \theta_X \) and \( \theta_Y \) based on the estimated value of \( \mathcal{X} \) in the previous iteration. Next, based on the estimated parameters, we estimate \( \mathcal{X} \) that maximize the objective function in Equation 4–3.

The likelihood function is non-convex in terms of the parameters \( \theta_Y = \{\mu_0, \sigma, \tau\} \). Also, the conditional density is non-convex in terms of \( \theta_X = \{\gamma_1, \cdots \gamma_4\} \). We use a global optimization method called differential evolution to optimize both of them [100]. To optimize the objective function in equation 4–3, we employ the ICM algorithm described by Besag [92]. Briefly, our iterative algorithm works as follows.

- Obtain an initial estimate of \( \mathcal{S} \) variables. In our implementation we use student’s t-test assuming the data follows normal distribution. We use 5% confidence interval for this purpose.
- Estimate parameters \( \theta_Y \) that maximizes the data likelihood function given by,

\[
\arg\max_{\theta_Y} \prod_{X_i} p(Y_i, Y_j|X_{ij}, X_{ij}^*, \theta_Y, W_{ij} = 1)
\]

We implement this step using Differential Evolution, which is similar to the genetic algorithm.
- Calculate an estimate of the parameters \( \theta_X \) that maximizes the conditional prior density function by,

\[
\arg\max_{\theta_X} \prod_{X_i} p(X_{ij}|\mathcal{X} = \{X_{ij}\}, \theta_X, W_{ij} = 1)
\]

We also implement this step using Differential Evolution.
• Carry out a single cycle of ICM using the current estimate of $S$, $\theta_X$ and $\theta_Y$. For all $S_i$, maximize $\prod_{X_{m}} p(X_{m} | X' - X_{mn}, Y, \theta_X, \theta_Y, W_{mn} = 1)$ when $X_{mn} \in \{X_{rt} | r = i \text{ or } t = i\}$ and $W_{rt} = 1$.

• Go to step 4.1.6 for a fixed number of cycles or until $X$ converges to a certain predefined value.

We optimize the objective function in terms of the $S_i$ ($1 \leq i \leq M$) variables instead of $X_{ij}$ variables. Specifically, in step 4, we go over all the $S_i$ variables, and optimize $F_{ij}$ function (given by Equation 4–4) for only those $X_{ij}$ variables that are impacted by the change of $S_i$. The optimization procedure is guaranteed to converge since in every iteration the value of the objective function increases. We continue the iterative process, until the changes in estimates of the parameters between two consecutive iterations reach below a certain cutoff level.

4.2 Experiments

In this section we discuss the experiments we conducted to evaluate the quality of our method. We implemented our method in MATLAB® and Java™. We obtained an implementation of Differential Evolution from the http://www.icsi.berkeley.edu/~storn/code.html. We compared our method with SSEM [20] as SSEM is one of the most recent methods that can be used to solve the problem considered in this chapter. We obtained SSEM from http://gardnerlab.bu.edu/SSEMLasso. We ran our code on an AMD Opteron 2.4 Ghz workstation with 4GB memory.

Dataset. We use the dataset collected by Smirnov et al. [101]. It was generated using 10 Gy ionizing radiation over immortalized B cells obtained from 155 members of 15 Centre d’Etude du Polymorphisme Humain (CEPH) Utah pedigrees [102]. Microarray snapshots were obtained at 0th hour (i.e., before the radiation) and 2 and 6 hours after the radiation. We adapt the time series data to create the control and non-control data for our experiments. We use the data before radiation as control data. For the non-control data we calculate the expected expressions of a gene at each points after the radiation. We select the one with higher absolute difference from the expected
Table 4-2. List of top 25 genes that are mostly affected by external perturbation for the real data adapted from Smirnov et al.

<table>
<thead>
<tr>
<th></th>
<th>PGF</th>
<th>IL8RB</th>
<th>FOSL1</th>
<th>F2R</th>
<th>PPM1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDM2</td>
<td>CDKN1A</td>
<td>TNC</td>
<td>PLXNB2</td>
<td>EPHA2</td>
<td></td>
</tr>
<tr>
<td>DDB2</td>
<td>TP53I3</td>
<td>PLK1</td>
<td>TNFSF9</td>
<td>ADRB2</td>
<td></td>
</tr>
<tr>
<td>MAP3K12</td>
<td>JUN</td>
<td>SORBS1</td>
<td>LRDD</td>
<td>SDC1</td>
<td></td>
</tr>
<tr>
<td>MYC</td>
<td>PRKAB1</td>
<td>EI24</td>
<td>DDIT4</td>
<td>FAS</td>
<td></td>
</tr>
</tbody>
</table>

expression of control data for that gene. This dataset is used in the experiments described in Sections 4.2.1 and 4.2.2. For the experiments described in Sections 4.2.3 and 4.2.4, we derive new datasets using this data. The details of this process can be found in corresponding sections.

We also collect 24,663 genetic interactions from the 105 regulatory and signaling pathways of KEGG database [103]. Overall 2,335 genes belong to at least one pathway in KEGG. We consider only the genes that take part in the gene networks in our model.

4.2.1 Evaluation of Biological Significance

In this section, we investigate the support in existing literature for susceptibility to radiation based perturbation for the primarily affected genes found by our method. We train our method on the dataset described above. After the optimization we rank each gene $g_i$ in decreasing order of $\mathcal{L}(g_0) \mathcal{L}(g_i)$, where $\mathcal{L}(g_i)$ is given by Equation 4–6. We tabulate the top 25 genes in Table 4-2.

Nine out of the ten highest ranked genes have significant biological evidence that they are impacted by radiation. Imaoka et al. [104] compared the gene expression between normal mammary glands to spontaneous and $\gamma$-radiation induced cancerous glands of rat. The PGF (parental growth factor) gene showed differential expression in both spontaneous and irradiated carcinomas. Nagtegaal et al. [105] applied radiation to human rectal adenocarcinoma and compared the gene response to that of normal tissues. The cytokines and receptor IL8RB showed differential expression between normal and irradiated rectal tissues. Amundson et al. [106] administered $\gamma$-radiation to p-53 wild type ML-1 human myeloid cell line. FOSL1 (known by FRA1 that time) showed
differential expression as the stress response. Lin et al. [107] applied ionizing radiation on human lymphoblastoid cells. F2R, a coagulation factor II receptor, was upregulated in that experiment. Jen et al. [108] investigated the effect of ionizing radiation on the transcriptional response of lymphoblastoid cells in time series microarray experiments. PPM1D, a gene related to DNA repair, showed response to both 3Gy and 10Gy radiation. Wu et al. [109] conducted a high dose UV radiation experiment to observe the relation between MDM2 gene on p53 gene. Their experiment revealed that initially both protein and mRNA level of MDM2 increases in a p53 independent manner, which clearly substantiated the direct effect of radiation on MDM2. Jakob et al. [110] irradiated human fibroblasts with accelerated lead ions. Confocal microscopy discovered a single, bright focus of CDKN1A protein in the nuclei of human fibroblast within 2 minutes after radiation. Rieger et al. [111] applied both ultra violet and infrared radiation on fifteen human cell lines and observed that PLXNB2 was up-regulated for both kind of radiations. Zhang et al. [112] reported that EPHA2 worked as an essential mediator of UV-radiation induced apoptosis.

This experiment demonstrates that we find sufficient support in existing literature that the top ranked genes found by our method (i.e. highly likely to be primarily affected) are affected by radiation.

4.2.2 Evaluation of the Rankings of Neighbor Genes

Recall that our goal is to find the primarily affected genes. We achieve this objective by computing the probability for each DE gene to contribute towards the change in the expression of its outgoing neighbors. In this experiment, we evaluate our success in terms of how accurately we rank the contribution probabilities of the genes as discussed in the next paragraph.

We divide the dataset of 155 samples into training and testing set in 2:1 ratio. We create a ranked list for each DE gene as follows. For each DE gene, we sort its incoming DE neighbors in decreasing order of their data likelihood probability with respect to
The average ranking distance between training and testing data is less than one position in 92.7% of the cases.

For a gene \( g_i \), we define the average ranking distance between training and testing data as

\[
\delta(g_i) = \frac{\sum_{g_j \in \text{IN}(g_i)} \text{abs}(\rho(g_j) - \rho'(g_j))}{|\text{IN}(g_i)|},
\]

where \( \text{IN}(g_i) \) is the set of incoming DE neighbors for \( g_i \), \( \text{abs}(\cdot) \) denotes the absolute value and \(|\text{IN}(g_j)|\) stands for the cardinality of \( \text{IN}(g_j) \).

We calculated the average ranking distance for all the genes that have incoming neighbors apart from the metagene. This experiment was repeated three times with
a different set of training and testing data. We create a histogram for the average
differences from the three experiments in Figure 4-3. It shows that the difference in
average ranking distance is very close to zero. The average difference between the
ranks obtained in the training and the testing data is less than one position in 92.7% of
the cases. Thus, we have demonstrated that we can accurately rank the contribution
probabilities of incoming neighbors for DE genes in test dataset based on the model
parameters learned from the training dataset.

4.2.3 Comparison to Other Methods

In this section, we compare the accuracy of our method to that of SSEM and a
simpler method Student’s t test.

Synthetic data generation. We simulated real perturbation events to prepare
synthetic data with known primarily and secondarily affected genes in a controlled
setting. We use the gene network derived from KEGG first to select a random gene from
the network and denote it as a primarily affected DE gene. We traverse the ancestors
in a breadth first manner. For each of the ancestor, we made it a secondarily affected
DE gene with a probability of \(1 - (1 - q)^\eta\), where \(\eta\) is the number of incoming DE
neighbors. Here \(q\) (0.4 in our experiments) is the probability that a gene is DE due to
a DE predecessor. We repeat these steps to create the desired number of primarily
affected genes. After the classification of the genes we create control and non-control
data for each of them for over \(N\) patients. We use the control part of the real dataset
in Smirnov et al. [101] as the control part of our synthetic dataset. To generate the
non-control dataset, we traverse each of the genes that participate in the gene networks.
Suppose, for a gene \(g_i\), the mean and standard deviation of its expression in the control
dataset are given by \(\mu_i\) and \(\sigma_i\) respectively. If the gene is EE we generate its non-control
data points from the a normal distribution given by the parameters \((\mu_i, \sigma_i^2)\). If the gene is
DE, we use the same variance as that of the control group. However, we use a different
Figure 4-4. Comparison of our method to SSEM and t-test. The number of primarily affected genes is 50. The gap between the mean of primarily affected and secondarily affected genes are 0.2 to 0.6 $\times \sigma$, where $\sigma$ is estimated from the real dataset. The figures indicate that our method outperforms SSEM and t-test.

mean. For the primarily and secondarily affected genes we use $\mu_i' = \mu_i \pm d_p$ and $\mu_i' = \mu_i \pm d_s$ respectively, where $d_p > d_s$.

**Experimental setup.** Given an input dataset, using each of the three methods, we ranked all the genes. Highly ranked genes have higher chance of being a primarily affected gene according to each method. We explain how we do the ranking in the following.

- **OUR METHOD.** We sort the genes in decreasing order of joint likelihood with the metagene. A higher joint likelihood implies a higher chance of being primarily affected.

- **SSEM.** We train SSEM on the control dataset, where it learns the correlation between the genes. We test SSEM on the non-control dataset, where it produces a rank for each single data point.

- **STUDENT’S T TEST.** We used the function called $ttest2$ from MATLAB®. We apply it on every individual gene, where it takes control and non-control dataset as input and produces a p-value as output. By default, null hypothesis is that “the differences of two input data set are a random sample from a normal distribution with mean 0 and unknown variance, against the alternative that the mean is not 0”. Thus, the null hypothesis corresponds to the assumption that the gene is EE. So a
substantially lower p-value implies a higher chance of being primarily affected. We performed the test on all the genes and rank them according the increasing order of p-values.

Let us assume the set of primarily affected genes as \( PG \) and first \( k \) elements of the ranking as \( RG_k \). We define the sensitivity of the ranking at position \( k \) by \( \eta_k = \frac{|PG \cap RG_k|}{|PG|} \).

Thus, a higher value of \( \eta_k \) denotes a higher sensitivity. We prepare a sensitivity vector \( \{ \eta_1, \eta_2, \cdots \eta_{|R|} \} \), by arraying the sensitivity of a ranking at all the positions of the ranks. Here, \( |R| \) denotes the cardinality of the ranking. For SSEM we obtain a sensitivity vector for every data points in the non-control dataset. We create a consolidated sensitivity vector by averaging them.

**Results.** We conducted experiments by for \( \frac{d_s - d_p}{\sigma} = \{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.5, 1.75\} \), number of primarily affected genes = \{10, 50\} and number of data points = \{10, 20, 40, 60, 80, 100, 125, 155\}. Here, \( \sigma \) corresponds to the standard deviation of the expressions of genes in the dataset. However, due to space limitation we discuss only two of them in this chapter (see Figure 4-4). The results we discuss correspond to the case when we have 50 primarily affected genes and 155 data points. The results of the other experiments are similar to those in Figure 4-4B.

Figures 4-4A and 4-4B show the sensitivity of the three methods when \( (d_s - d_p) = 0.2 \times \sigma \) and \( 0.6 \times \sigma \) respectively. The former one corresponds to the computationally harder case as the difference between the control and non-control datasets is small. As the gap between \( d_s \) and \( d_p \) increases identifying primarily affected genes becomes easier.

From the figure, we observe that our method is significantly more sensitive than the other two methods for all datasets consistently. It reaches high sensitivity (more than 90%) using the top 150 ranked genes when the gap is small, and using the top 50 genes as the gap increases to \( 0.6 \times \sigma \). The results were similar for larger gap values (results not shown). The t test reaches around 40% and 50% sensitivity at 200 ranking position...
Figure 4-5. Comparison of accuracies with SSEM and Student’s t test while varying the ratio of gaps of primarily and secondarily affected genes. For a category of gene, the gap denotes the absolute difference of average expressions in control and non-control groups. The x-axis represents the ratio of gaps of primarily and secondarily affected genes. The y-axis denotes the accuracy of our method as described in Section 4.2.4. The figure demonstrates that our method obtains very high accuracy except when the ratio equals to zero, i.e., the gap is equal for both the primarily and secondarily affected genes.

respectively. SSEM’s sensitivity is below 0.25 for all experiments even within the top 200 positions.

We believe that there are two major factors for improved results using our method. First, our method can successfully incorporate the gene interactions while other methods ignore this information. Second, our method is capable of dealing with a broad range of primarily affected genes while other methods’ performance deteriorates as this number grows. In real perturbation experiments, often multiple genes are primarily affected. Thus, we conclude that our method is more suitable for real perturbation experiments.
4.2.4 Sensitivity to the Gap between Primary and Secondary Effects

The experiments over the real dataset suggest the validity of our model. One question however follows from these experiments. How does our method compare when we vary the distinction between primarily and secondarily affected genes in terms of their gap between control and non-control data. To answer this question we conducted experiments on synthetic datasets, where we change the differences between primarily and secondarily affected genes and compare our the accuracy of our method with SSEM and student’s t test.

**Synthetic data generation.** We generate the data in the presence of a hypothetical perturbation to simulate the real dataset. The primarily and secondarily affected genes are ascertained in the way described in Section 4.2.3.

To utilize the real dataset to maximum possible extent, we employ an innovative approach. Let us denote the mean of gene $g_i$ in the control and non-control by $\mu_i$ and $\mu'_i$, respectively. We subtract the difference $(\mu'_i - \mu_i)$ from all the expressions in the non-control group of $g_i$. We repeat this subtraction for all the genes. Once the non-control group is *leveled* to control group, we re-modify the non-control expressions of DE genes. If a gene is primarily DE according to the decided set of genes, we increase or decrease its expression over the data points in non-control group by $d_p$. Similarly, we modify the expression value by $d_s$, if the gene is secondarily affected. Here, $d_p$ is greater than $d_s$.

**Results.** We created three different sets of data by varying $d_p$ and $d_s$. For all the datasets the number of primarily affected genes was 40. For every dataset, we used different values of $d_p$ given by $\{0.8, 1.2, 1.6\} \times \sigma$, respectively. However, within a dataset $d_p$ was fixed and $d_s/d_p$ ratio was varied as $\{0.1, 0.2, \cdots 1.0\}$. We discuss only the result for the dataset $d_p = 0.8 \times \sigma$ as the results for the other are similar. The accuracy of the methods can fluctuate for different set of affected genes. Hence, for a particular value of
\(d_s\) and \(d_p\) we repeated the experiment five times with different sets of affected genes and averaged the result.

We run the three methods on all the datasets and extract ranks of genes as described in Section 4.2.3. A higher position in the rank indicates a higher chance of being primarily differentially expressed. Let the set of true primarily affected genes be \(PA\). Let \(RG\) be the set of first \(|PA|\) genes from the rank produced by a method, where \(|PA|\) is the cardinality of \(PA\). We define accuracy of that method as \(\frac{|PA \cap RG|}{|RG|}\).

Figure 4-5 depicts the result from this experiment. It is clear that our method outperforms SSEM all the time. The accuracy of our method is substantially better than Student’s t test for all the cases except when the ratio \(d_s/d_p\) equals to one. From this experiment, we can conclude that our method performs very well over a wide range of difference between the non-control groups for primarily and secondarily affected genes. Specifically, for the case where these groups have the same mean, our method perform almost as well as other methods.

4.3 Discussion

In this chapter, we considered the problem of identifying primarily affected genes in the presence of an external effect that can perturb the expressions of genes. We assumed that we were given the expression measurements of a set of genes before and after the application of an external perturbation. We developed a new probabilistic method to quantify the cause of differential expression of each gene. Our method considers the possible gene interactions in regulatory and signaling networks, for a large number of perturbations. It uses a Bayesian model with the help of Markov Random Fields to capture the dependency between the genes. It also provides the underlying distribution of the impact with confidence interval.

Our experiments on both real and synthetic datasets demonstrated that our method could find primarily affected genes with high accuracy. It achieved significantly better accuracy than two competing methods, namely SSEM and the student’s t test method.
CHAPTER 5
IDENTIFYING DIFFERENTIALLY REGULATED GENES

Microarray experiments often measure expressions of genes taken from sample tissues in the presence of external perturbations such as medication, radiation, or disease [82, 83]. Typically in such experiments, gene expressions are measured before and after the application of external perturbation, and are called control data and non-control data, respectively. In this chapter, we focus on an important class of such microarray experiments that inherently have two groups of tissue samples. Different groups in a microarray measurement can exist in many different ways. For instance, samples can be taken from members of multiple closely related species (e.g. rat versus mouse). Within the same species there can be subgroups with different phenotypes (e.g. African American versus Caucasian American). Another example is when the samples have already been through several alternative external perturbations (e.g. fasting and not fasting). When such different groups exist, it is not only important to observe overall changes in gene expression, but also to observe how different groups respond to the external perturbation. For example, Taylor et al. applied medications on 36 Caucasian American and 33 African American patients infected with Hepatitis C [47]. Gene expressions were collected before and after the medication.

In a perturbation experiment, some of the genes respond by noticeably changing their expression values between the control and non-control data. Genes that change their expressions in a statistically significant way are referred to as differentially expressed (DE), while those that do not, are referred to as equally expressed (EE) genes. In the context of two groups, we refer to a gene that has the same state in both the groups, i.e. either DE or EE for both the groups, as equally regulated (ER) gene. On the contrary, if a gene is DE in one group and EE in the other, we denote it as differentially regulated (DR).
Genes for any organism typically interact with each other via regulatory and signaling networks. For simplicity, we will refer to them as *gene networks* for the rest of this chapter. A small portion of an example gene network can be seen in Figure 5-1.

Once an external perturbation is applied, a gene can be DE in one of the two ways – as a direct effect of the perturbation or via interaction with other DE genes through gene networks. We denote a gene as *primarily affected* DE, if it is DE due to the external perturbation. Similarly, a gene is *secondarily affected* DE, if it is DE due to another gene in the gene network. Figure 5-1 shows the state of the genes in the pancreatic cancer pathway after a hypothetical external perturbation is applied. In this figure, genes K-Ras, Raf and Cob42Roc are primarily affected and MEK, Ral and RalGDS are secondarily affected through interactions.

Recall that for a gene to be DR, it has to be EE in one group and DE in another group. For such a gene, if it happens to be DE in one group because of the external perturbation, we call it as *primarily differentially regulated (PDR)* gene. When it is DE in one group because of the interaction with other DE genes in the gene networks, we will...
refer to it by secondarily differentially regulated (SDR) gene. In this chapter, we consider the problem of identifying the PDR genes in a given set of control and non-control gene expressions from two groups of samples.

**Our approach.** In this chapter, we propose a new probabilistic Bayesian method CMRF to find the PDR genes in two group perturbation experiment dataset as defined above. We call this method CMRF (Comparative MRF) for it applies MRF on two groups of data for comparison purpose. Our Bayesian method incorporates information about relationship from gene networks as prior beliefs. We consider the gene network as a directed graph where each node represents a gene, and a directed edge from gene \( g_i \) to gene \( g_j \) represents a genetic interaction (e.g. \( g_i \) activates or inhibits \( g_j \)). We define two genes as *neighbors* of each other if they share a directed edge. For example, in Figure 5-1, genes K-Ras and Raf are neighbors as K-Raf activates Ras. We also classify a neighbor as *incoming* or *outgoing*, if it is at the start or at the end of the directed edge respectively. In Figure 5-1, Raf is an incoming neighbor of MEK and MEK is an outgoing neighbor of Raf. When the expression level of a gene is altered, it can affect some of its outgoing neighbors. Thus, the gene expression can change due to external perturbation or because of one or more of the affected incoming neighbors.

We represent the external perturbation by a hypothetical gene (i.e. *metagene*) \( g_0 \) in the gene network. We add an edge from the metagene to all the other genes because the external perturbation has the potential to affect all the other genes. So, \( g_0 \) is an incoming neighbor to all the other genes. We call the resulting network the *extended gene network*.

CMRF estimates the probability that a gene \( g_j \) is DR due to an alteration in the activity of gene \( g_i \) (\( \forall g_i \in G \cup \{g_0\}, \ g_j \in G \)) if there is an edge from \( g_i \) to \( g_j \) in the extended network. We use a Bayesian model in our solution with the help of Markov Random Field (MRF) [94] to capture the dependency between the genes in the extended gene network. We define feature functions that encapsulate the domain knowledge available
Figure 5-2. A hypothetical gene network and corresponding Markov random graph. (a) A small hypothetical gene network with perturbation in two datasets $D_A$ and $D_B$. The genes in the two datasets interact through identical network, although they assume different states. The circle $g_0$ represents the abstraction of the external perturbation. Rectangles denote genes. $\rightarrow$ implies activation and $\leftarrow$ implies inhibition. The potential effect of metagene to all other genes is indicated by dotted arrows from the metagene to all the other genes. For example, $g_1$ is primarily affected in $D_A$, but not affected in $D_B$. $g_2$ is primarily affected in both the datasets. $g_3$ is secondarily affected in both $D_A$ and $D_B$. (b) The Markov Random Filed graph constructed based on the small hypothetical gene network in (a). The numbers in the parenthesis are the expected assignments to the variables based on the states of the genes in (a). Nodes with dotted boundaries indicate that those nodes are required for completeness of the model, however the corresponding interactions do not exist.

from gene networks and gene expression data. CMRF optimizes the joint posterior distribution over the random variables in the MRF using Iterated Conditional Modes (ICM) [92]. The optimization provides the state of the genes, the regulation of the genes and the probabilistic estimate of pairwise interactions between the genes including the metagene. Given this, we can rank the genes according to the data likelihood that a gene is DR because of the metagene $g_0$, and obtain a list of possible PDR genes.

Figure 5-2 illustrates different components of CMRF and the connectivity between them. Note that, (C) corresponds to the Bayesian prior based on MRF.
We compare the accuracy of CMRF with that of SSEM and Students t test on semi-synthetic dataset generated from microarray data in Cosgrove et al. [20]. We also compare CMRF with our old method SMRF that we developed to identify the primarily affected DE genes in a single group perturbation data [113]. CMRF obtains high accuracy and outperforms all the other three methods. Also, we conduct a statistical significance test using a parametric noise based experiment to evaluate the accuracy of CMRF. In this experiment our model demonstrates reasonable confidence regions for various values of the parameters.

The rest of the chapter is organized as follows. Section 5.1 describes our methods in detail. Section 5.2 presents the results of our experiments. Section 5.3 concludes our discussion.

5.1 Methods

In this section we describe different components of CMRF. Section 5.1.1 describes the notation and formulates the problem. Section 5.1.2 provides a high level overview of the solution. Section 5.1.3 describes the calculation of the prior density function of MRF. Section 5.1.4 discusses the definition of a tractable objective function. Section 5.1.5 discusses the calculation of the likelihood function. Finally, Section 5.1.6 describes the algorithm to optimize the objective function.

5.1.1 Notation and Problem Formulation

In this section, we describe our notation and formally define the problem. We define a Bayesian model for gene expression in a two-group perturbation experiment. We classify the random variables of the model into two different groups, namely observed variables and hidden variables. We have the values for the observed variables, while we estimate the values of the hidden variables.

**Observed variables.** We define two sets of observed variables, one for microarray gene expression data and another for the neighborhood in the extended gene network.
Table 5-1. Enumeration of the values of $Z_i$, $Z_j$ and $X_{ij}$ for different values of $S_{Ai}$, $S_{Bi}$, $S_{Aj}$ and $S_{Bj}$. The hidden variables are oriented in a hierarchical structure. For instance, the value of $Z_i$ depends on the values of $S_{Ai}$ and $S_{Bi}$. Similarly, the value of $X_{ij}$ depends on the values of $Z_i$ and $Z_j$. Thus, the value of the dependent variable $X_{ij}$ in turn depends on the values of four independent variables $S_{Ai}$, $S_{Bi}$, $S_{Aj}$ and $S_{Bj}$.

<table>
<thead>
<tr>
<th>$S_{Ai}$</th>
<th>$S_{Bi}$</th>
<th>$S_{Aj}$</th>
<th>$S_{Bj}$</th>
<th>$Z_i$</th>
<th>$Z_j$</th>
<th>$X_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>EE</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DE</td>
<td>DE</td>
<td>EE</td>
<td>DE</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DE</td>
<td>DE</td>
<td>EE</td>
<td>EE</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DE</td>
<td>EE</td>
<td>DE</td>
<td>DE</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>DE</td>
<td>EE</td>
<td>DE</td>
<td>EE</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>DE</td>
<td>EE</td>
<td>EE</td>
<td>DE</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>DE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>EE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>3</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>EE</td>
<td>DE</td>
<td>DE</td>
<td>EE</td>
<td>3</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>EE</td>
<td>DE</td>
<td>EE</td>
<td>DE</td>
<td>3</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>EE</td>
<td>DE</td>
<td>EE</td>
<td>EE</td>
<td>3</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>EE</td>
<td>EE</td>
<td>DE</td>
<td>DE</td>
<td>4</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>EE</td>
<td>EE</td>
<td>DE</td>
<td>EE</td>
<td>4</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>DE</td>
<td>4</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>4</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

- **MICROARRAY DATA.** We denote the number of genes by $M$ and the number of data points in the two groups $D_A$ and $D_B$ by $N_A$ and $N_B$ respectively. We represent the set of genes with $G = \{g_1, g_2, \ldots, g_M\}$. For each gene and for each group the microarray data contains the gene expression values before and after the perturbation, i.e. control and non-control data respectively. We denote the expression value of the $i$th gene from the $j$th sample in the control data of group $D_A$ with $y_{Ai}$. We represent the same for the non-control data with $y_{Ai}'$. Thus the expression values of the gene $g_i$ for all the samples in $D_A$ for control and non-control data are $y_{Ai} = \{y_{Ai1}, y_{Ai2}, \ldots, y_{AiN_A}\}$ and $y_{Ai}' = \{y_{Ai1}', y_{Ai2}', \ldots, y_{AiN_A}'\}$ respectively. We denote all the expression values in group $D_A$ for gene $g_i$ with $Y_{Ai}$ (i.e. $Y_{Ai} = y_{Ai} \cup y_{Ai}'$). We denote the collection of the gene expressions of all the genes in group $D_A$ by $Y_A = \bigcup_{i=1}^{M} Y_{Ai}$. We define $Y_B$ similarly for all the genes in $D_B$. We refer the complete gene expression data using variable $Y = Y_A \cup Y_B$.

- **NEIGHBORHOOD VARIABLES.** We use the term $W = \{W_{ij}\}$ to indicate if two genes $g_i$ and $g_j$ are neighbors in the extended gene network. If $g_i$ is an incoming neighbor of $g_j$ (i.e. $g_j$ has an incoming edge from $g_i$), then we set the value of $W_{ij}$ ($1 \leq i, j \leq M$) to 1. It is 0 otherwise.
**Hidden variables.** We define three sets of hidden variables. These variables govern the state of genes, regulations of genes and interactions among genes respectively.

- **STATE VARIABLES.** We use $S_A = \{S_{Ai}\}$ and $S_B = \{S_{Bi}\}$, $(1 \leq i \leq M)$ to denote the states of the genes in group $D_A$ and $D_B$. $S_{Ai} = 1$ if $g_i$ is DE in $D_A$ and 0 if it is EE in $D_A$. We define $S_{Bi}$ similarly. We assume that the metagene $g_0$ is DE for both $D_A$ and $D_B$. Thus, $S_{A0} = S_{B0} = 1$.

- **REGULATION VARIABLES.** We denote the regulation condition of gene $g_i$ with $Z_i$. Table 5-1 enumerates different values of $X_{ij}$ for the values of $S_{Ai}$ and $S_{Bi}$. In this formulation, the cases $Z_i = \{2, 3\}$ indicate that $g_i$ is DR, whereas $Z_i = \{1, 4\}$ indicate that $g_i$ is ER. The metagene is guaranteed to be ER, since $S_{A0} = S_{B0} = 1$.

- **INTERACTION VARIABLES.** In order to govern the joint regulation states of genes $g_i$ and $g_j$ we define interaction variables $X = \{X_{ij}\}$, $(1 \leq i, j \leq M)$. Mathematically, $X_{ij} = 4 \times (Z_i - 1) + Z_j$. Note that, this equation is created to maintain brevity of the mapping between the interaction variables and the regulation variables by carefully assigning different numeric constants between one and 16 to appropriate values of an interaction variable. Table 5-1 enumerates different values of $X_{ij}$ for values of $Z_i$ and $Z_j$. Specifically, $X_{0j} \in \{2, 3\}$ and $X_{ij} \in \{1, 4\}$ correspond to the cases where $g_j$ is DR and ER respectively because of interaction with the metagene $g_0$.

It is easy to see that the hidden variables follow a hierarchical structure. For instance, the value of $Z_i$ depends on the values of $S_{Ai}$ and $S_{Bi}$. Similarly, the value of $X_{ij}$ depends on the values of $Z_i$ and $Z_j$. Thus, the value of the dependent variable $X_{ij}$ is based on the values of four independent variables $S_{Ai}$, $S_{Bi}$, $S_{Aj}$ and $S_{Bj}$. Table 5-1 enumerates the values of $Z_i$, $Z_j$ and $X_{ij}$ for different values of $S_{Ai}$, $S_{Bi}$, $S_{Aj}$ and $S_{Bj}$.

It is worth noting that the different values that we assign to the hidden variables are categorical in nature.

**Problem formulation.** Let $G = \{g_1, g_2, \cdots, g_M\}$ denote the set of all genes. Using the definition of the neighborhood variables $W$, we denote the collection $(G, W)$ by $V$ which essentially represents the gene networks. We denote the metagene by $g_0$. Given an observed data $\{\mathcal{V}, \mathcal{Y}\}$ we want to estimate the probabilities $p(X_{ij} = x|X - X_{ij}, \mathcal{Y}, \mathcal{V})$, $x \in \{1, 2, \cdots, 16\}$. 

80
A higher value of $p(X_{o|} = \{2, 3\}|\cdot)$ indicates a higher probability of a gene $g_j$ being PDR. Using the estimated values of $p(X_{o|}, \forall j \in \{1, 2, \cdots M\}$, we can create an ordered list of candidate PDR genes.

### 5.1.2 Overview of the Solution

This section describes a high level overview of our approach to estimate $p(X_{o|}, \forall j \in \{1, 2, \cdots M\}$. One simple approach can be using a hypothesis test to find out the PDR genes in the given dataset [24]. However, the available hypothesis tests do not consider the interactions among genes in the gene network. Also, deciding on the significance of test can be a complex step. Another approach can be to use SSEM to create a rank of the potential primarily affected genes in each group separately [20]. Then we can select the top $k$ genes in each group and perform a set difference to obtain the PDR genes. Though SSEM considers the correlation between the genes, it does not utilize any known information from the gene networks.

We build a Bayesian probabilistic method based on Markov Random Field where we leverage the information from gene networks as the prior belief of the model. Using Bayes theorem [93] we can write the joint probability density of interaction variables $\mathcal{X}$ as,

$$
P(\mathcal{X} | \mathcal{Y}, \mathcal{V}) = \frac{P(\mathcal{Y} | \mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}}) P(\mathcal{X} | \mathcal{V}, \theta_{\mathcal{X}})}{\sum_{\mathcal{X}} P(\mathcal{Y} | \mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}}) P(\mathcal{X} | \mathcal{V}, \theta_{\mathcal{X}})} \quad (5-1)
$$

The first term in the numerator, $P(\mathcal{Y} | \mathcal{X}, \mathcal{V}, \theta_{\mathcal{Y}})$, is the likelihood of the observed expression data $\mathcal{Y}$ given the interaction variables and gene network. $\theta_{\mathcal{Y}}$ represents the parameters for the likelihood function. A detailed discussion of how we compute this likelihood can be found in Section 5.1.5.

The second term in the numerator $P(\mathcal{X} | \mathcal{V}, \theta_{\mathcal{X}})$ represents this prior belief. $\theta_{\mathcal{X}}$ represents the parameters for the prior density function. We define a Markov Random Field (MRF) over the interaction variables $\mathcal{X}$ and the priors are encoded via feature functions in the MRF. Details of the priors and the associated feature functions are outlined in Section 5.1.3. The denominator of Equation 5–1 is the normalization constant.
that represents the sum of the product of the likelihood and the prior over all possible assignments of interaction variables \( \mathcal{X} \).

Given the joint probability density function outlined in Equation 5–1, our original problem reduces to obtaining assignments for the interaction variables \( \mathcal{X} \) and the parameters \( \theta_x \) and \( \theta_y \) that maximize it.

A Maximum Likelihood Estimation (MLE) of Equation 5–1 is practically infeasible even for a small number of genes since the number of terms in the denominator grows exponentially. Instead we use a pseudo-likelihood version of the objective function as shown in Section 5.1.4. We use Iterative Conditional Modes (ICM) [92] and Differential Evolution [100] in an alternating optimization technique to maximize the pseudo-likelihood with respect to \( \mathcal{X}, \theta_x \) and \( \theta_y \).

After the optimization, we obtain an assignment for \( \mathcal{X}, \theta_x \) and \( \theta_y \). Using these assignments and the observed data, we estimate the posterior probability of all \( X_{ij} \) variables. Using the estimated values of \( p(X_{0j} | \cdot), \forall j \in \{1, 2, \cdots M\} \), we create an ordered list of candidate PDR genes. We elaborate on each of these steps next.

Figure 5-3 illustrates different portions of CMRF and the connectivity between them.

### 5.1.3 Computation of the Prior Density Function

In this section, we describe how we incorporate gene network as the the prior belief into our Bayesian model. From the structure and properties of gene network, we build three hypotheses and embed them into our model. We present the entire concept in three numbered subsections.

**Statement of Hypotheses.** Here we state the three hypotheses on the biological networks in brief.

- **HYPOTHESIS 1.** In each group \( D_T (T \in \{A, B\}) \), the metagene \( g_0 \) can change the state of all the other genes. Thus, all the genes can be directly affected by the external perturbation.
Figure 5-3. Illustration of different components of CMRF and connectivity between them. (A) obtains an initial estimates of state variables using Student's t test. (B) estimates parameters in a way that maximizes data likelihood. (C) estimates parameters in order to maximize prior density. Both (B) and (C) use a global optimization technique called Differential Evolution. (D) employs Iterated Conditional Modes to maximize the pseudo-likelihood. (B), (C) and (D) consist of an alternating optimization technique. These three steps (B), (C) and (D) are repeated till the algorithm meets a criteria for completion. Finally, once the optimization is complete, the DR genes are sorted in decreasing order of their likelihood with respect to the metagene $g_0$. The genes at the top of the list are declared PDR.

- **HYPOTHESIS 2.** In each group $D_T$ ($T \in \{A, B\}$), a gene $g_i$ can change the states of its outgoing neighbors $g_j$ in the same data group, i.e. a gene can be indirectly affected by the perturbation through genetic interactions.

- **HYPOTHESIS 3.** Each gene has a high probability of being equally regulated. This follows from the observation that, often the difference between the expressions of most of the genes in two groups is small. We expect that the response of genes in these groups is very similar.

Clearly, when the data does not follow one or more of the hypotheses, the optimization function can overcome the prior belief with a strong support from the data.
Markov Random Field construction. In order to compute the prior density function, we define a Markov Random Field (MRF) over the $x$ variables \cite{94}. MRF is a probabilistic model, where the state of a variable depends only on the states of its neighbors. MRF is useful to model our problem as the states of genes depend on their neighbors. Here, the MRF is an undirected graph $\psi = (x, e)$, where $x = \{x_{ij}\}$ variables represent the vertices of the graph (i.e. each interaction variable $x_{ij}$ corresponds to a vertex). We denote the set of edges with $e = \{(x_{ij}, x_{pq}) | W_{ij} = W_{pq} = 1\} \cup \{(x_{ij}, x_{ik}) | W_{ik} = W_{ij} = 1\}$. Thus, two variables in $x$ share an edge if they share a common subscript at the same position and the two genes corresponding to the other subscript interact in the gene network. For example, in Figure 5-2(b), $x_{35}$ and $x_{25}$ are neighbors, as they share 5 (i.e. gene $g_5$) as the second subscript and $g_2$ and $g_3$ interact in the gene network in Figure 5-2(a).

One important point to note is that, this graph does not use the state variables $s$ or the regulation variables $z$ to model the dependencies between the genes. Rather, it establishes those dependencies over the $x$ variables. For example, in Figure 5-2(b) we draw the MRF graph corresponding to the hypothetical gene network in Figure 5-2(a). In the gene network, there is an edge from $g_2$ to $g_3$. So, $g_2$ can potentially change the state of $g_3$. We create an edge from $x_{12}$ to $x_{13}$ that corresponds to the edge from $g_2$ to $g_3$. As $g_1$ is common for $x_{12}$ and $x_{13}$, if they assume the same value (i.e. $x_{12} = x_{13}$), it implies that the genes $g_2$ and $g_3$ are in same state (i.e. $S_T = S_{T3}$, $T \in \{A, B\}$). We formulate these dependency constraints using a set of unary and binary functions called feature functions. We discuss these feature functions next.

Development of feature functions. We denote the neighbors of $x_{ij}$ in the MRF graph as $x_{ij}^* = \{x_{ik} | W_{ki} = 1\} \cup \{x_{ip} | W_{ip} = 1\}$. We define a clique over each $x_{ij}$ and its neighbors $x_{ij}^*$ by $C_{ij}$ provided $W_{ij} = 1$. A feature function $f(C_{ij})$ is a Boolean function defined over the clique $C_{ij}$. This function evaluates to one or zero, if it is satisfied or not, respectively. We define a potential function $\psi(C_{ij})$ corresponding to $f(C_{ij})$ as an
exponential function given by \( \exp(\gamma f(C_{ij})) \). Here \( \gamma \) is a coefficient associated with \( f(C_{ij}) \) that represents the relevance of \( f(C_{ij}) \) in the MRF. According to Hammersley-Clifford theorem, we express the joint density function of the MRF over \( \mathcal{X} \) as product of potential functions defined over that MRF as, \( \rho(\mathcal{X}|\theta_{\mathcal{X}}) = \frac{1}{\Delta} \prod_{C_{ij},w_{ij}=1} \psi(C_{ij}) \) [95]. In this formulation, \( \Delta \) is the normalization function \( \Delta = \sum_{\mathcal{X}} \prod_{C_{ij}} \psi(X_{ij}) \). To limit the complexity of our model, we consider only cliques of size one and two.

Table 5-2. Enumeration of five different unary feature functions \( F_1, F_2, F_3, F_6 \) and \( F_7 \)

<table>
<thead>
<tr>
<th>( X_{ij} )</th>
<th>( F_1 )</th>
<th>( F_2 )</th>
<th>( F_3 )</th>
<th>( F_6 )</th>
<th>( F_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We define seven feature functions to capture the dependencies among the variables in \( \mathcal{X} \) according to the three hypotheses.

**Unary feature functions** \( F_1, F_2, F_3 \). A primary component of the prior density function is modeling the frequency of \( X_{ij} \) itself. Here, we focus on two values of \( X_{ij} \) namely \( X_{ij} = \{2, 3\} \), since they correspond to the events that a gene \( g_j \) is DR due to the metagene \( g_0 \). When \( X_{ij} = 2 \), \( g_j \) is DE in \( D_A \) and EE in \( D_B \). To capture this, we define a feature function \( F_1(X_{ij}) \) which returns one when \( X_{ij} = 2 \). It returns zero otherwise. Similarly, \( X_{ij} = 3 \) when \( g_j \) is EE in \( D_A \) and DE in \( D_B \). We define another feature function \( F_2(X_{ij}) \), which returns one when \( X_{ij} = 3 \). We capture all the other values of \( X_{ij} \) by a
feature function called $F_3(X_{ij})$. It returns zero when $X_{ij} \in \{2, 3\}$ and equals to one otherwise. Table 5-2 enumerates the the domains and ranges of $F_1$, $F_2$ and $F_3$.

**Binary feature functions** $F_4$, $F_5$. Let $\gamma$ represent the hypothesis that in a group $D_T$, $T \in \{A, B\}$ a gene $g_i$ including the metagene can change the state of one of its outgoing neighbors $g_k$. We make a stronger hypothesis $\gamma^o$ that, $\gamma$ holds simultaneously in $D_A$ and $D_B$ with high probability. Note that, this stronger hypothesis is based on the assumption that the genes in both $D_A$ and $D_B$ express in a similar fashion. This assumption is meaningful as in these two-group perturbation experiments the different groups belong to similar biological conditions [47].

$\gamma^o$ is encoded in $\mathcal{X}$ domain as follows. Consider four genes $g_p$, $g_i$, $g_j$ and $g_k$, such that $g_p \rightarrow g_i$, $g_i \rightarrow g_j$ and $g_j \rightarrow g_k$. Here $\rightarrow$ indicates that the gene on the left activates or inhibits the gene on the right. By definition, $(X_{pj}, X_{ji})$ and $(X_{ij}, X_{ik})$ are edges in the MRF. Note that the first edge corresponds to an incoming neighbor $g_p$ of $g_i$, while the second edge corresponds to an outgoing neighbor $g_k$ of $g_j$. We discriminate between these two sets of neighbors of $X_{ij}$, as they are related to the incoming neighbors of $g_i$ and outgoing neighbors of $g_j$ respectively. It can be shown that, for the first set of edges, $X_{pj}$ equals to $X_{ji}$ if and only if (iff) $Z_p = Z_i$, i.e. $\gamma^o$ holds true. Similarly, for the second set of edges $X_{ij}$ equals to $X_{ik}$ iff $Z_j = Z_k$, which in tern implies that $\gamma^o$ is satisfied.

We define two sets of feature functions to formalize these equalities based on the incoming neighbors of $g_i$ and the outgoing neighbors of $g_j$.

- **LEFT EXTERNAL EQUALITY.** We denote the incoming neighbors of $g_i$ with $ln(g_i)$. We write a feature function $f_4(X_{pj}, X_{ji})$, $\forall g_p \in ln(g_i)$. $f_4(X_{pj}, X_{ji}) = 1$ if $Z_i = Z_p$ and $W_{pi} = W_{ji} = 1$. Otherwise, $f_4(X_{pj}, X_{ji}) = 0$. We denote the summation of this function over all the incoming neighbors of $g_i$ as,

$$F_4(X_{ji}) = \sum_{p, W_{pi}=1, W_{ji}=1} f_4(X_{ij}, X_{pi}).$$

- **RIGHT EXTERNAL EQUALITY.** We denote the outgoing neighbors of $g_j$ as $Out(g_j)$. We define a feature function $f_5(X_{ik}, X_{ji})$, $\forall k, g_k \in Out(g_j)$. $f_5(X_{ik}, X_{ji}) = 1$ if $S_k = S_j$ and $W_{jk} = W_{ij} = 1$. Otherwise, $f_5(X_{ik}, X_{ji}) = 0$. We denote the summation of this
function over all the outgoing neighbors of \( g_j \) as,

\[
F_5(X_{ij}) = \sum_{k, W_{ik} = 1, W_{ij} = 1} f_5(X_{ij}, X_{ik}).
\]

Table 5-3. The table enumerates the truth values for the binary feature function *left external equality* (*f_4*). Only the possible entries are annotated with zero and one. The other entries require different values of \( Z_j \) in \( X_{ij} \) and \( X_{pi} \), which is not possible. Note, that the feature function can assume one only when \( X_{ij} \) and \( X_{pi} \) are equal, which is in accordance with the definition of that feature function.

<table>
<thead>
<tr>
<th>( X_{ij} )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Tables 5-3 and 5-4 enumerate the values of \( f_4 \) and \( f_5 \) for different values of \( X_{ij} \). The missing entries in these tables correspond to the cases which can not occur during the optimization. For instance, in Table 5-3, a missing entry corresponds to different values of \( Z_j \) in \( X_{ij} \) and \( X_{pi} \) which is not possible.

For feature functions \( f_4 \) and \( f_5 \), \( X_{pj} \) or \( X_{ik} \) may not represent any interactions from the extended gene network when \( W_{pj} = 0 \) or \( W_{ik} = 0 \) respectively. We represent them by dotted rectangles in Figure 5-2(b).
Table 5-4. The table enumerates the truth values for the binary feature function *right external equality* \( f_6 \). Only the possible entries are annotated with zero and one. The other entries require different values of \( Z_i \) in \( X_{ij} \) and \( X_{ik} \), which is not possible. Note, that the feature function can assume the value one only when \( X_{ij} \) and \( X_{ik} \) are equal, which is in accordance with the definition of right external equality.

\[
\begin{array}{cccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
X_{ij} \rightarrow & 1 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 0 \\
3 & 0 & 0 & 1 & 0 \\
4 & 0 & 0 & 0 & 1 \\
5 & 1 & 0 & 0 & 0 \\
6 & 0 & 1 & 0 & 0 \\
7 & 0 & 0 & 1 & 0 \\
8 & 0 & 0 & 0 & 1 \\
9 & 1 & 0 & 0 & 0 \\
10 & 0 & 1 & 0 & 0 \\
11 & 0 & 0 & 1 & 0 \\
12 & 0 & 0 & 0 & 1 \\
13 & 1 & 0 & 0 & 0 \\
14 & 0 & 1 & 0 & 0 \\
15 & 0 & 0 & 1 & 0 \\
16 & 0 & 0 & 0 & 1 \\
\end{array}
\]

**Unary feature functions** \( F_6, F_7 \). We introduce two unary feature functions to incorporate our last hypothesis, that all genes are ER with a high probability. We consider two genes \( g_i \) and \( g_j \) such that \( g_i \rightarrow g_j \). This hypothesis holds true, if \( g_i \) is equally regulated or \( g_j \) is equally regulated.

- **LEFT INTERNAL EQUALITY.** We define this feature function to capture the events when \( g_i \) is equally regulated. As, \( g_j \) can assume any state, this feature function holds true for eight different values of \( X_{ij} \). We denote the feature function by \( f_6(X_{ij}, t) \) that returns one if its two arguments are equal and zero otherwise. We denote the summation of this functions over all these eight values of \( X_{ij} \) as,

\[
F_6(X_{ij}) = \sum_{i,j, W_6=1, t \in \{1, \ldots, 4, 13, \ldots, 16\}} f_6(X_{ij}, t).
\]

- **RIGHT INTERNAL EQUALITY.** We define this feature function to capture the events when \( g_j \) is equally regulated. As, \( g_i \) can assume any state, this feature function holds true for eight different values of \( X_{ij} \). We denote the feature function by
$f_t(X_{ij}, t)$ that returns one if its two arguments are equal and zero otherwise. We denote the summation of this functions over all these eight values of $X_{ij}$ as,

$$F_t(X_{ij}) = \sum_{i,j, W_{ij}=1, t \in \{1,4,5,8,9,12,13,16\}} f_t(X_{ij}, t).$$

The last two columns of Table 5-2 enumerate these two internal equalities.

Based on these feature functions, we define the joint density function of $\mathcal{X}'$ as,

$$p(\mathcal{X}'|\theta_{\mathcal{X}}) = \frac{1}{\Delta} \exp\left( \sum_{i,j, W_{ij}=1, k \in \{1,2,\ldots,7\}} \gamma_k F_k(X_{ij}) \right)$$

(5–2)

In the above equation $\gamma_k, k \in \{1, 2, \ldots, 7\}$ are the coefficients of the seven feature functions in MRF.

In the next section, we discuss how we approximate the objective function of the MRF and the data. We also describe how we formulate the posterior probability density function for $X_{ij}$.

**5.1.4 Approximation of the Objective Function**

A direct maximization of the objective function given by Equation 5–1 is intractable, as it requires evaluation of exponential number of terms in the denominator. We employ pseudo-likelihood as an established substitute to Equation 5–1 [96]. Pseudo-likelihood is the simple product of the conditional probability density function of the $X_{ij}$ variables. Geman et al. proved the consistency of the maximum pseudo-likelihood estimate [97].

The approximated objective function can be written as,

$$F = \arg \max_{\mathcal{X}} \left( \prod_{i,j} F_{ij} \right)$$

(5–3)

The posterior density function $F_{ij}$ of $X_{ij}$ as,

$$F_{ij} = p(X_{ij}|\mathcal{X} - X_{\bar{ij}}, \mathcal{Y}, \theta_{\mathcal{X}}, \theta_{\mathcal{Y}})$$

$$= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}|X_{\bar{ij}}, X_{ij}^*, \theta_{\mathcal{Y}}) p(X_{ij}|\mathcal{X} - X_{\bar{ij}}, \theta_{\mathcal{X}})}{\sum_{X_{ij} \in \{1,\ldots,16\}} p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}|X_{ij}, X_{ij}^*, \theta_{\mathcal{Y}})}$$

(5–4)
Derivation of $F_{ij}$.

\[
F_{ij} = p(X_{ij} | X - X_{ij}, Y_A, Y_B, Y_{ij}, \theta_X, \theta_Y)
= p(X_{ij} | X - X_{ij}, Y_A, Y_B, Y_{ij}, \theta_X, \theta_Y)
= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}, X - X_{ij} - X_{ij}^*, X_{ij}^*, \theta_X, \theta_Y)}{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}, X - X_{ij} - X_{ij}^*, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}
= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}, X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}, X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}
= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}
= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}
= \frac{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}{p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X - X_{ij} - X_{ij}^* | X_{ij}, X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y) p(X_{ij}^*, \theta_X, \theta_Y)}
= \sum_{X_{ij} \in \{1, 2, 3, \ldots, 16\}} p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj} | X_{ij}, X_{ij}^*, \theta_Y) p(X_{ij} | X - X_{ij}, \theta_X)
In step 2 of the derivation, we substitute \( Y \) by \( Y_{Ai}, Y_{Bi}, Y_{Aj} \) and \( Y_{Bj} \) as \( X_{ij} \) is independent of all \( Y_{Ck} \) such that \( k \neq \{i, j\} \) and \( C \neq \{A, B\} \). Also, in the 15th step we assume that \( X_{ij} \) is independent of \( \theta_Y \) given \( \mathcal{X} - X_{ij} \) and \( \theta_X \).

**Derivation of** \( p(X_{ij}|\mathcal{X} - X_{ij}, \theta_X), W_{ij} = 1 \).

\[
p(X_{ij}|\mathcal{X} - X_{ij}, \theta_X) = \frac{p(\mathcal{X}, \theta_X)}{P(\mathcal{X} - X_{ij}, \theta_X)} = \frac{p(\mathcal{X}, \theta_X)}{\sum_{X_{ij} \in \{1,2,3,\ldots,16\}} p(\mathcal{X}, \theta_X)} = \frac{A(X_{ij}) \cdot B(ij)}{\sum_{t = \{1,2,3,\ldots,16\}} A(t) \cdot B(ij)}
\]

\( A(X_{ij}) \) is \( \exp(\sum_{k \in \{1,2,\ldots,7\}} \gamma_k F_k(X_{ij})) \) and \( B(ij) \) is given by \( \exp(\sum_{m,n,i,j,m,n,k \in \{1,2,\ldots,7\}} \gamma_k F_k(X_{mn})) \).

Here, we denote the prior density parameters \( \{\gamma_1, \gamma_2, \cdots, \gamma_7\} \) by \( \theta_X \). Canceling \( B(ij) \) from numerator and denominator the density function simplifies to,

\[
p(X_{ij}|\mathcal{X} - X_{ij}, \theta_X) = \frac{\exp(\sum_{k \in \{1,2,\ldots,7\}} \gamma_k F_k(X_{ij}))}{\sum_{t = \{1,2,3,\ldots,16\}} \exp(\sum_{k \in \{1,2,\ldots,7\}} \gamma_k F_k(X_{ij} = t))}
\]

There are two different terms in objective function of Equation 5–4. \( p(X_{ij}|\mathcal{X} - X_{ij}, \theta_X) \) stands for the conditional prior density function of \( X_{ij} \) which we just have derived from using Bayes rule. In the next section, we discuss the likelihood function \( p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}|X_{ij}, X_{ij}^*, \theta_Y) \).

### 5.1.5 Computation of Likelihood Density Function

In this section, we describe how we derive the likelihood function in three numbered subsections. Here, we assume that gene expressions in a group follow a normal
distribution, We can rewrite the derivations if gene expressions follow some other
distribution.

**Likelihood for a single gene.** Consider a set of measurements for a gene \( g_i \) that
follows a single Gaussian distribution by \( z_i = \{ z_{i1}, z_{i2}, \cdots, z_{in} \} \). We denote the latent
mean of \( z_i \) as \( \mu \) and the standard deviation as \( \sigma \). As different genes can have different
average expressions, we assume that \( \mu \) follows a genome wise distribution with mean \( \mu_0 \nolimits \) and standard deviation \( \tau \) \cite{98}. Thus, for \( z_i \), the likelihood for the data points in that group
is given by,

\[
L(z|\mu_0, \sigma^2, \tau^2) = \int [\prod_{i=1}^{n} N(z_i|\mu, \sigma^2)] N(\mu|\mu_0, \tau^2) d\mu \\
= \frac{\sigma}{(\sqrt{2\pi}\sigma)^n} \frac{1}{\sqrt{\tau^2 +\sigma^2}} \exp\left(-\frac{\sum_i z_i^2}{2\sigma^2} - \frac{\mu_0^2}{2\tau^2} \right), \quad (5–5)
\]

The derivation of Equation 5–5 can be obtained from Demichelis et al. \cite{99}. If a gene
is DE, its expression measurements in control and non-control groups follow different
distributions \cite{98}. On the other hand, for equally expressed genes, all the measurements
in both groups share the same mean. The likelihood function for a DE gene \( g_i \) in group
\( D_T, T \in \{ A, B \} \) is given by,

\[
\mathcal{L}_{T_{DE}}(g_i) = L(y_i|\mu_0, \sigma^2, \tau^2)L(y'_i|\mu_0, \sigma^2, \tau^2) \quad (5–6)
\]

Similarly, for EE genes it is given by,

\[
\mathcal{L}_{T_{EE}}(g_i) = L(y_i \cup y'_i|\mu_0, \sigma^2, \tau^2) \quad (5–7)
\]

For instance, the likelihood of a gene to be DE in group \( D_A \) is given by \( \mathcal{L}_{A_{DE}}(g_i) \).
**Likelihood for a regulation variable.** As for a gene $g_i$, the regulation variable $Z_i$ can assume four different values from 1 to 4, the equations of the likelihood that a gene is DR or ER also take four different forms given by,

$$
L_Z(g_i) = \begin{cases} 
L_{A_{de}}(g_i)L_{B_{de}}(g_i), & \text{if } Z_i = 1. \\
L_{A_{de}}(g_i)L_{B_{ee}}(g_i), & \text{if } Z_i = 2 \\
L_{A_{ee}}(g_i)L_{B_{de}}(g_i), & \text{if } Z_i = 3 \\
L_{A_{ee}}(g_i)L_{B_{ee}}(g_i), & \text{if } Z_i = 4
\end{cases}
$$

**Likelihood for an interaction variable.** We have 16 different forms for the likelihood of the $X_{ij}$ due to its 16 different values. However, here, we shall derive only for $X_{ij} = 1$, as for the other values of $X_{ij}$ we have a similar derivation.

From the definition of $X_{ij}$, $p(Z_i = \tau_i, Z_j = \tau_j, \theta_Y|X_{ij} = 1, X_{ij}^*, \theta_Y) = 1$ when $Z_i = 1$ and $Z_j = 1$. Its value is zero for all other values of $Z_i$ and $Z_j$. So, continuing from the last step of Equation 5–8,

$$
p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}|X_{ij} = 1, X_{ij}^*, \theta_Y)
= p(Y_{Ai}, Y_{Bi}, Y_{Aj}, Y_{Bj}|Z_i = 1, Z_j = 1, \theta_Y) \\
= p(Y_{Ai}, Y_{Bi}|Z_i = 1, Z_j = 1, \theta_Y).
$$

(5–9)
In a similar way, we can derive the likelihood functions for all the 16 different values of \( X_{ij} \) variables. A special case arises when \( g_i \) is the metagene, i.e. \( g_0 \). We assume that \( \mathcal{L}_{T_{oe}}(g_0) = 1 \) and \( \mathcal{L}_{T_{ee}}(g_0) = 0 \), \( T \in \{A, B\} \). Thus, the likelihood of the metagene given \( Z_0 = 1 \) equals to 1. Its value is zero otherwise.

5.1.6 Objective Function Optimization

So far, we have described how we compute the posterior density function. The final challenge is to find the values of the hidden variables that maximize the objective function (Equation 5–3). We develop an iterative algorithm to address this challenge.

In our model we have three different sets of parameters. The nodes of the MRF given by \( \mathcal{X} \) consist of one set. Other two sets are the parameters of conditional probability density function of \( X_{ij} \) and likelihood function of observed data given by \( \theta_X = \{\gamma_1, \cdots, \gamma_7\} \) and \( \theta_Y = \{\mu_0, \sigma, \tau\} \), respectively. In each iteration, we first estimate \( \theta_X \) and \( \theta_Y \) based on the estimated value of \( \mathcal{X} \) in the previous iteration. Next, based on the estimated parameters, we estimate \( \mathcal{X} \) that maximize the objective function in Equation 5–3.

The likelihood function is non-convex in terms of the parameters \( \theta_Y = \{\mu_0, \sigma, \tau\} \). Also, the conditional density is non-convex in terms of \( \theta_X = \{\gamma_1, \cdots, \gamma_7\} \). We use a global optimization method called differential evolution to optimize both of them \[100\]. To optimize the objective function in Equation 5–3, we employ the ICM algorithm described by Besag \[92\]. Briefly, our iterative algorithm works as follows.

1. Obtain an initial estimate of \( \mathcal{S} \) variables. In our implementation we use student’s t-test assuming the data follows normal distribution. We use 5% confidence interval for this purpose.

2. Estimate parameters \( \theta_Y \) that maximizes the data likelihood function given by,

\[
\operatorname{arg\,max}_{\theta_Y} \prod_{X_0, W_0 = 1} p(Y_{A_i}, Y_{B_i}, Y_{A_j}, Y_{B_j} | X_{ij}, X_{ij}', \theta_Y)
\]

We implement this step using Differential Evolution, which is similar to the genetic algorithm.
3. Calculate an estimate of the parameters $\theta_X$ that maximizes the conditional prior density function by,

$$\arg\max_{\theta_X} \prod_{X_i, W_i=1} p(X_{ij} | X' - \{X_{ij}\}, \theta_X)$$

We also implement this step using Differential Evolution.

4. Carry out a single cycle of ICM using the current estimate of $S$, $\theta_X$ and $\theta_Y$. For all $S_t$, maximize $\prod_{X_m} p(X_{mn} | X' - X_{mn}, Y, \theta_X, \theta_Y)$ when $X_{mn} \in \{X_{rt} | r = i \text{ or } t = i, W_{rt} = 1\}$.

5. Go to step 2 for a fixed number of cycles or until $X'$ converges to a certain predefined value.

We optimize the objective function in terms of the $S_i$ ($1 \leq i \leq M$) variables instead of $X_{ij}$ variables. Specifically, in step 4, we go over all the $S_i$ variables, and optimize $F_{ij}$ function (given by Equation 5–4) for only those $X_{ij}$ variables that are impacted by the change of $S_i$. Figure 5-3 illustrates different components of CMRF and the connectivity between them.

The optimization procedure is guaranteed to converge since in every iteration the value of the objective function increases. We continue the iterative process, until the changes in estimates of the parameters between two consecutive iterations reach below a certain cutoff level.

5.2 Results and Discussion

In this section we discuss the experiments we conducted to evaluate the quality of CMRF. We implemented CMRF in MATLAB® and Java™. We obtained the code for Differential Evolution from http://www.icsi.berkeley.edu/~storn/code.html. We compared CMRF with SSEM as SSEM is one of the most recent methods that considers identifying primarily affected genes in a perturbation experiments [20].

We obtained SSEM from http://gardnerlab.bu.edu/SSEMLasso. We executed our code on a Quad-Core AMD Opteron 2 Ghz workstation with 32 GB of memory.

Dataset. We used four different sets of data to conduct the experiments in this chapter.
• **DATASET 1.** The first dataset was collected by Smirnov et al. [101]. This dataset was generated using 10 Gy ionizing radiation over immortalized B cells obtained from 155 members of 15 Centre d’étude du Polymorphisme Humain (CEPH) Utah pedigrees [102]. Microarray snapshots were obtained before (at zeroth hour) and after (at second and sixth hours) the application of radiation.

• **DATASET 2.** The second dataset corresponds to a drug response experiment conducted by Taylor et al. [47]. Medications were applied on 36 Caucasian American and 33 African American patients infected with Hepatitis C. Gene expressions were collected before the medication was started and at 1, 2, 7, 14, 28 days after the medication was administered. Both dataset 1 and 2 are microarray time series data with more than two time points. We adapted these two time series dataset two create control and non-control data suitable for our experiments. We used the data before perturbation as control data. For the non-control data we calculated the expected expression of a gene at each points after the perturbation. We selected the one with highest absolute difference from the expected expression of control data for that gene.

• **DATASET 3.** We created dataset 3 using dataset 1. We used the control group of dataset 1 as the control group of dataset 3. Then, we changed the expression values of some of the randomly selected genes to model the primary effect of external perturbation. From that perturbed dataset, we simulated the secondary effects using the sigmoid method described in Garg et al. [114]. We denote the parameter for primary perturbation effect by deviation. Deviation is the ratio of the change of expression value \( \Delta x \) of a gene to its original expression value \( x \) (i.e. \( \text{derivation} = \frac{\Delta x}{x} \) ) which is normalized between zero and one. We tuned the other parameters of the method to create a meaningful dataset as follows; \( \alpha = 1, \beta = 0.01, k_{ac} = 1.0, k_{in} = 1, h = 0.1 \).

• **DATASET 4.** We create this dataset from dataset 1 in two steps as follows.

  - **SELECTION OF GENES.** In order to carry out experiments on larger scale data with known PDR genes, we generated data in the presence of a hypothetical perturbation from the real datasets as follows. We first select three sets of genes. Each set consists of some primarily affected genes and a higher number of secondarily affected genes. Here, we describe how we construct each of the three sets of affected genes. We first select a random gene from the network and label it as a primarily affected DE gene. We then traverse its outgoing neighbors in a breadth first search manner. As we visit a gene during traversal, we label it as a secondarily affected DE gene with a probability of \( 1 - (1 - q)^\eta \), where \( \eta \) is the number of incoming DE neighbors. Here \( q \) is the probability that a gene is DE due to a DE predecessor (0.4 in our experiments). We repeat these steps to create the desired number of primarily affected genes.
After we obtain the three set of genes, we assign one set to both $D_A$ and $D_B$ groups. We assign the other two sets of genes to different groups. These two set of genes are differentially regulated as they are affected in only one group and not in the other. The three groups can contain different number of primarily and secondarily affected genes. We call these three sets of genes as primarily differentially regulated, secondarily differentially regulated and equally regulated genes.

- **Generation of Gene Expression.** Once we identify these three sets of genes in the two groups, we create control and non-control data for $D_A$ and $D_B$ over $N$ samples. We use the control part of the real dataset in Smirnov et al. as the control part of our synthetic dataset in both $D_A$ and $D_B$ [101]. To generate the non-control dataset, we traverse each of the genes that participate in the gene networks. Consider a gene $g_i$ with mean and standard deviation of expression in the control dataset given by $\mu_i$ and $\sigma_i$ respectively. If the gene is EE we generate its non-control data points from the a normal distribution given by the parameters $(\mu_i, \sigma_i^2)$. If the gene is DE, we use the same variance but different mean as that of the control group. For the primarily and secondarily affected genes we use $\mu_i \pm d_p$ and $\mu_i \pm d_s$ respectively, where $d_p > d_s$.

To summarize, we used the same variance in the non-control group as that in the control group. However, for an affected gene we changed the value of the mean in the non-control group from that in the control group. For a primarily affected gene we applied a higher deviation of mean than that of the secondarily affected genes.

**Regulatory networks.** We collected 24,663 genetic interactions from the 105 regulatory and signaling pathways of KEGG database [103]. Overall 2,335 genes belong to at least one pathway in KEGG. In our model, we considered only the genes that take part in the gene networks.

### 5.2.1 Comparison to Other Methods

Our method provides us a list of differentially regulated genes. We sort the list of those genes as follows. Consider a DR gene $g_i$, which is DE in $D_A$ and EE in $D_B$. We calculate the likelihood of being EE in $D_A$ and DE in $D_B$ for that gene. We can interpret this step as the probability of being DR, but in a reverse way. We could instead use the probability that the gene is DE in $D_A$ and EE in $D_B$. However, according to our observation, the earlier metric provides a much better accuracy. We sort all the DR genes with increasing order of that likelihood.
As per our knowledge, no other method exists that differentiates between the primary and secondary effects in a two-group perturbation experiment. There exist some studies in identifying primarily affected genes in single group datasets. We compared the accuracy of CMRF to three such methods namely, SMRF, Student’s t test and SSEM.

**Experimental setup.** Given an input dataset, using each of the four methods, we ranked all the genes. Highly ranked genes have higher chance of being a PDR according to each method. However, as other three methods are not tailored to solve this problem, we created separate ranking on $D_A$ and $D_B$. Then, out of those two ranks, we created a unified rank of differentially regulated genes. We shall elaborate on this unified rank creation later. We, first, explain how we create ranks on individual groups $D_A$ and $D_B$ for other three methods.

- **SMRF.** We apply the SMRF to each group separately and obtain a set of differentially expressed genes. We sort the genes in decreasing order of joint likelihood with the metagene. A higher joint likelihood implies a higher chance of being primarily affected.

- **SSEM.** We train SSEM on the control dataset, where it learns the correlation between the genes. We test SSEM on the non-control dataset of each group, where it produces a rank for each single data point.

- **STUDENT’S T TEST.** We use the function called `ttest2` from MATLAB®. We apply it on every individual gene, where it takes control and non-control dataset as input and produces a p-value as output. We assume that the null hypothesis corresponds that the gene is EE. So a substantially lower p-value implies a higher chance of being primarily affected. We perform the test on all the genes and rank them according the increasing order of p-values.

Now we describe how we create an unified ranking of differentially regulated genes for these three methods. We denote the ranks from data group $D_A$ and $D_B$ by $R_A$ and $R_B$ respectively. The unified rank is defined by $R_U$. We denote the number of genes in each rank to be $\omega_A$ and $\omega_B$ respectively. We scan both the ranks simultaneously from first position to $\omega = \min(\omega_A, \omega_B)$. While scanning at the $k$th position, we denote the equally regulated set obtained till that position by $\Lambda_k = R_A(1 : k) \cap R_B(1 : k)$. We include $R_U(k)$
Figure 5-4. Comparison of our method CMRF to SMRF, SSEM and t-test. The number of primarily differentially affected genes is 40. The values for deviation (maximum perturbation to the PDR genes) are 0.6 and 0.8. The figures indicate that CMRF outperforms SMRF, SSEM and t-test.

Results. In this experiment we used dataset 3, that we have just described. To observe the accuracy of CMRF at varying degree of difficulties, we conducted experiments with four different values of deviation, namely, \{0.5, 0.6, 0.7, 0.8\}. However, we discuss only two of them in this chapter (see Figure 5-4) since for other two parameters the results are similar. The results we discuss correspond to the cases when deviation = \{0.6, 0.8\}.

Figures 2(a) and 2(b) show the sensitivity of the four methods with the two deviation settings. The former one corresponds to the computationally harder case as the difference between the non-control groups of primarily and secondarily affected genes is small. As the deviation increases identifying primarily affected genes becomes easier.

Form the figure, we observe that CMRF is significantly more accurate than the other three methods for all datasets consistently. It reaches almost 50% sensitivity (i.e., it can find around 15-18 primarily affected genes out of 30) in the top 50 ranked genes, when the deviation is 0.6. On the other hand, its achieves a sensitivity of 0.6 when
the deviation is 0.8. We obtained similar results for other deviations, which we do not discuss here. The method in SMRF reaches to 30% and 40% accuracy, however at a slower pace. The t-test reaches around 25% and 30% sensitivity at ranking position 50 for these two cases respectively. SSEM’s sensitivity is below 0.1 for all experiments even within the top 50 positions.

We believe that there are three major factors for the success of our method over the other competing methods. First, the other methods do not simultaneously handle two groups of datasets and are not able to generate an unified ranking of differentially regulated genes. CMRF encompasses both groups in a single model and probabilistically determines the PDR genes. Hence, it is more shielded against the false positives introduced during the unification of ranking. Second, CMRF can successfully incorporate the gene interactions using MRFs while others ignore this information. Finally, in real perturbation experiments, multiple genes are often primarily affected. CMRF is capable of dealing with both large and small number of primarily affected genes, while performances of other methods deteriorate as the number of primarily affected genes grows. Thus, we conclude that our method is more suitable for real perturbation experiments.

5.2.2 Statistical Significance Experiment

The experiments in the last section enable us to compare the accuracy of CMRF with that of the other methods on synthetic datasets. We also wanted to evaluate the accuracy of CMRF on real dataset. However, we do not have any gold standard available that enlists true set of PDR genes. Hence, we conducted a set of statistical significance experiments to estimate the confidence of our accuracy. Specifically, we obtained the control data from a real dataset, perturbed it in a controlled way for a number of genes. We calculated the likelihood probabilities of those genes and created a distribution. We repeated this process with varying amount of perturbation. Finally, we executed CMRF on a real dataset and analyzed the result.
Results. We obtained the real dataset from drug response experiment conducted by Taylor et al. [47], which is actually dataset 2. Apart from this real dataset, we create different versions of dataset 4 by varying $d_p$ as $\{0.1, 0.2, 0.3, \cdots, 3.0\}$. If $d_p > 1.1$, we set $d_s$ to 1, otherwise $d_s = 0.5 \times d_p$. Thus, we have 30 synthetic datasets in total. In every dataset, we fix the number of primarily and secondarily differentially regulated genes to 50 and 172 respectively.

To decide whether a gene $g_i$ is DR, when $g_i$ is DE in $D_A$ and EE in $D_B$, we define a null-hypothesis $H_0$: $g_i$ is DR, but in the reverse way, i.e. $g_i$ is EE in $D_A$ and DE in $D_B$. We calculate the likelihood of being EE in $D_A$ and DE in $D_B$ for that gene, as described. For gene $g_i$, we denote the log likelihood of accepting $H_0$ by $LL_i$. In every dataset, we create a box plot of the 50 $LL_i$ values, as the number of DR genes in each dataset is 50. A lower value $LL_i$ indicates that $g_i$ has a higher probability of being differentially regulated.

Figure 5-5 illustrates the statistical significance of the experiments over the datasets with $d_p = 1.2$ to 2.0. The box plot demonstrates a relationship between the P-value and $d_p$. A higher value of $d_p$ indicates a lower P-value and hence, a high chance of being PDR. We also observe that the variance of P-value increases with the increase of $d_p$.

We also executed CMRF on the real datasets without any modification. Interestingly, on the real dataset from Taylor et al. [47] (dataset 2), we did not obtain any genes as differentially regulated. A careful observation concludes that when both the number of data points and the gap $d_p$ (i.e. the signal to noise ratio) is low, the coefficients $\gamma_6$ and $\gamma_7$ in the prior density become strong and all genes are identified as equally regulated. However, when either the number of data points or $d_p$ is significantly high, the data can overcome the prior. In the current real dataset, the number of data points is only 33 and the gaps between the control and non-control group were less than $1.2 \times \sigma$. As a result, CMRF identifies no differentially regulated genes in the dataset. Thus, we can conclude that either there is not much difference between the two groups in the real data, or the
Figure 5-5. Illustration of the statistical significance test. Box plot demonstrates the P-value of null hypothesis of the DR genes over synthetic dataset. From the plot we clearly conclude that a higher gap between the control and non-control group of a DR gene leads to a lower P-value. The genes with a lower P-value have a higher chance of being primarily differentially regulated.

data does not contain enough data points, so that our model can highlight difference between the two groups.

In Figure 5-5 we present the results for $d_p = 1.2$ to 2.0. Note, that for $d_p < 1.2 \times \sigma$ our model did not identify any DR genes. Here also, we attribute a similar reason for not finding any DR genes as both $d_p$ and the number of data points are small. On the other hand, in Section 5.2.1, when we execute CMRF on synthetic datasets with 155 data points we were able to identify a substantial number of true PDR genes even with $d_p = 0.02 \times \sigma$.

To substantiate our conclusion that there exists little difference between the two groups in the real dataset, we conducted a set of permutation tests. We shuffled the two original groups to create new sets of data. We repeated this process for a number of
times (40 in the present experiment) and executed CMRF on each of them. For every derived dataset, CMRF did not find any DR genes. Hence, this experiment bolsters the claim that there are no DR genes in the original real data.

An interesting question can be raised is that “is there indeed no DR genes in the real dataset from Taylor et al. [47]?” Another similar question can be “will our method be able to detect DR and PDR genes from similar other real datasets?” We believe that CMRF requires a bigger dataset for DR and PDR genes to be discovered. For example, CMRF is able to identify the DR and PDR genes from the synthetic dataset that contains substantially higher number of data points than that of the real dataset. Since the difference between control and non-control groups of a DE gene is small compared to the variance of the data points, it is difficult to detect that subtle effect of perturbation with a small dataset. For a small dataset, the prior due to third hypothesis becomes strong and the two corresponding parameters $\gamma_6$ and $\gamma_7$ assumes extreme values. Thus the support from data is not sufficient to overcome the prior and hence, the method is not able to identify the DR and PDR genes. There are two solutions to overcome this problem. First of them is to employ a bigger dataset. With the advancement of comparatively inexpensive and high throughput technologies bigger dataset are increasingly common nowadays. From that perspective, CMRF is supposed to perform more accurately in the near future. A second option to circumvent the problem is to restrict the growth of the two parameters $\gamma_6$ and $\gamma_7$. If we have knowledge about the values of these two parameters, we can assign then as input to the program and refrain from estimating their values. This will enable us to employ a comparatively non-informative prior which will be easier for the data to overcome. Also, we can use specific bound over those variables while estimating them to avoid them becoming stronger.
5.3 Discussion

Microarray experiments often measure expressions of genes taken from sample tissues in the presence of external perturbations such as medication, radiation, or disease. Typically in such experiments, gene expressions are measured before and after the application of external perturbation.

In this chapter, we solved the problem of finding primarily differentially regulated genes in the presence of external perturbations when the data is sampled from two groups. The probabilistic Bayesian method based on Markov Random Field incorporates dependency structure of the gene networks as the prior to the model. Experimental results on synthetic and real datasets demonstrated the superiority of CMRF compared to other simple techniques.
CHAPTER 6
SSLPRED: PREDICTING SYNTHETIC SICKNESS LETHALITY

Analysis of gene essentiality is a crucial problem to understand the roles of different genes at the molecular and genetic levels. A gene is defined *essential* if it is required for proper growth and sustenance of that organism. Essential genes have been thoroughly investigated using techniques such as single gene deletion screening for some low level organisms such as *Escherichia coli (E. coli)* [115]. Though identification of essential genes enlightens us about the functions of individual genes in an organism, it provides little conclusive information about the nature of their genetic relationships in gene regulatory and signaling networks. Recently, studies on *Synthetic Sickness Lethality (SSL)* opened up new directions in the areas of functional genomics. Two *non-essential* genes follow an SSL interaction if their joint deletion leads to a less than expected *fitness* for the organism. Here *fitness* denotes the growth and sustenance rate of an organism. An expected fitness corresponds to that of a double mutant when the two knocked out genes are not in an SSL interaction. Note that the fitness of an organism due to an SSL interaction can be less than (aggravating) or more than (alleviating) the expected fitness [116]. A genome wise catalog of SSL interactions enables in-depth molecular analysis, by creating a functional map of the cell, predicting functions and relations of the genes and deciphering complex regulatory relations from the global genetic network [117].

The Synthetic Genetic Array (SGA) [27] and diploid-based synthetic lethality analysis on microarray (dSLAM) [28] are two pioneering approaches that enable systematic identification of SSL interactions. Both methods require generation of double mutant strains and monitoring their growth. Each entry of SGA is a triple that consists of two genes and a GI (genetic interaction) score for those two genes. A score close to zero indicates that there is no SSL interaction between the two genes. For a gene-pair, a negative GI with a large magnitude indicates an aggravating SSL
Figure 6-1. Illustration of the concepts of Synthetic Sickness and Lethality and Between Pathway Models. Figure 6-1A illustrates the concepts of synthetic sickness and lethality. A double mutant produced from the cross of two single mutants can have a specific fitness in a range of GI scores based on the relationship between the two genes. The single and double circles represent single and double mutants respectively. Here, the size of a circle corresponds to the observed fitness of the corresponding mutant. Based on whether the two genes have an epistatic, neutral or SSL interaction, the observed fitness of the double mutant can have more than, equal to or less than the expected fitness. In those cases, the GI score can be a significantly large positive, close to zero or significantly large negative number, respectively. Figure 6-1B depicts the concept of Between Pathway Models (BPM). The hypothetical BPM consists of two sub-networks (also called pathways) $G_A$ and $G_B$ who are functionally independent and complementing. The solid lines denote physical interactions, while the dashed directed lines stand for the SSL interactions. It is evident that the number of SSL edges between $G_A$ and $G_B$ is higher compared to the ones within the two groups.

B Between Pathway Models

A Synthetic Sickness Lethality

interaction. A significant large positive number denotes a higher chance of being alleviating relationship [116]. Figure 6-1A illustrates the concepts of synthetic sickness and lethality relationship. The EMAP strategy exploits the SGA technique by enabling colony sizes to be measured in an array format, thus quantifying genetic interactions in a high-throughput fashion.

Both SGA and dSLAM are costly techniques as for a pair of genes they require creation of two single mutant strains and crossing between them to produce a double mutant strain. For an organism with $N$ genes, we need to generate and monitor the growth of $\frac{N(N-1)}{2}$ different double mutants. As a result, millions of double mutants need to
be produced to tabulate all the genetic interaction scores for an organism that consists of thousands of genes. Creating such double mutants in wet-lab is an expensive and time consuming process. Therefore, we need an efficient method to predict whether there exists a synthetic lethality relation between two genes. Briefly, we can describe the problem considered in this chapter as follows: given two genes \( g_A \) and \( g_B \) what is the GI score between them?

In order to predict the GI score between two genes, we incorporate the genetic profile of single mutant strains. This is a promising strategy for the number of single mutants can not be more than the number of genes. First, we elaborate on the term genetic profile. Consider single gene knockout dataset (also termed as single gene mutant data). Here, in each experiment a non-essential gene is knocked out from an organism. For each gene, expressions are obtained before and after the knock-out and ratio of this after to before expression is calculated. Finally, the logarithm of that ratio is computed and tabulated. If the magnitude of a logarithm is large, it indicates that the expression of the corresponding gene changed significantly after the knockout of the non-essential gene under consideration. The genetic profile for a single mutant or a single gene knockout experiment consists of entries for all genes computed in the way described above.

In this chapter our objective is to learn the GI scores of gene pairs with the help of genetic profile of single mutants. Formally we solve the following problem.

**Problem.** Let \( V = \{g_1, g_2, \ldots, g_M\} \) denote the set of genes in an organism. Assume that we are given the genetic profiles of \( K \) single mutant genes. \( X \) is a \( K \times M \) matrix, where each row corresponds to the genetic profile of a single mutant. Let us represent the GI score of gene pairs \( g_a \) and \( g_b \) with \( t_{a,b} \). Let \( T \) denote the set of all the available GI scores for that organism. For any gene pair \( (g_i, g_j) \) such that \( g_i \in V, g_j \in V \), we would like to predict the GI score.
Before discussing our contribution in this chapter, we summarize the Between Pathway Models (BPM), which is a building block of our model [30]. A BPM consists of two gene subnetworks (also called pathways) $G_A$ and $G_B$, such that there are few SSL interactions within $G_A$ and within $G_B$, but many of those between $G_A$ and $G_B$. The opposite holds for the physical interaction edges. That is, many physical interactions exist within $G_A$ and $G_B$, but few of them exist between $G_A$ and $G_B$. Figure 6-1B depicts a hypothetical BPM. According to Kelley and Ideker, the two pathways in a BPM are functionally compensating due to the orientation of genetic and physical edges [30].

Now that we have introduced all the relevant building blocks, we discuss our contribution in this chapter.

**Contribution.** In this paper, we develop a new method $SSLPred$ to predict the GI scores. To our knowledge, our method is the first one to predict GI scores using a mathematical machine learning based technique.

In accordance with the concept of BPM, we propose the following conjecture. If there is an SSL interaction between two genes and if these two genes belong to two pathways of a BPM, then knocking out one of them will change the expressions of most of the genes in both of the pathways in that BPM. The pathway containing the mutated gene is directly affected and dysfunctional as most of the consisting genes have a direct connection with the mutated gene through physical edges. The other pathway compensates for its affected pair, and due to the additional activities the genes in it change their expressions noticeably.

In our regression based method $SSLPred$, we develop a mapping between the genetic profiles of single mutants and the corresponding GI score. For every genetic interaction entry $(g_a, g_b, t_{a,b})$, such that either of $g_a$ and $g_b$ has been mutated in a single mutant gene experiment and $t_{a,b}$ is the GI score for $g_a$ and $g_b$, we create a training sample. As we have already conjectured in the previous paragraph, if this genetic interaction entry represents an SSL, the mutated gene affects the expressions of all
the genes in the corresponding BPM. Thus, we use the gene expression changes only from the pathways of that BPM to extract the features of the training point and correlate it with the corresponding GI score $t_{a,b}$ using a regression model. After we estimate the parameters of SSLPred, we are able to predict the GI score for a new pair of genes.

We compare our method to the one by Hescott et al. [31] in their ability to identify BPMs in the gene networks of *S. cerevisiae* on four benchmark datasets. On average SSLPred performs significantly better compared to the other method. We summarize our contribution as follows:

1. According to our knowledge, SSLPred is the first *predictive method* to predict the GI score for a pair of genes. All other relevant computational methods are *descriptive*.

2. The GI scores predicted by SSLPred assume a real value. This is more useful than a binary prediction, since it enables to conduct statistical analysis such as permutation tests and p-Value generation associated with the validation of benchmark BPMs.

The rest of the chapter is organized as follows. Section 6.1 describes our method SSLPred. Section 6.2 presents the experimental results. Finally, Section 6.3 concludes the chapter.

### 6.1 Methods

In this section, we discuss our method in detail. Section 6.1.1 describes the notation and formulates the problem. Section 6.1.2 explains our conjectures which guide our model and the rationale behind it. Section 6.1.3 discusses the feature extraction and regression model.

#### 6.1.1 Problem Formulation and Notation

In this section, we mathematically formulate the problem, and for that purpose, we describe the relevant notation. We group our notation in three classes based on three related entities. These are gene network, single mutant data and SGA. Here, gene network stands for gene interaction network, specifically the union of gene regulatory and signaling network.
1. **GENE NETWORK.** The gene network is a union of gene regulatory and signaling networks that can be modeled as a set of genes and the directed edges (i.e., interactions) connecting these genes. Here, an edge between two genes denotes different kinds of genetic interactions such as activation, inhibition and phosphorylation. Let us denote the set of all \( M \) genes by \( \mathcal{V} = \{g_1, g_2, \ldots, g_M\} \). We denote the set of all edges in the gene network by \( \mathcal{W} = \{(g_i, g_j) | g_i \in \mathcal{V}, g_j \in \mathcal{V}\} \), where \((g_i, g_j)\) implies a directed interaction from \( g_i \) to \( g_j \). Thus, \( \mathcal{G} = (\mathcal{V}, \mathcal{W}) \) defines the gene network.

2. **SINGLE MUTANT DATASET.** In a single mutant, one gene is mutated in an organism and gene expression is obtained before and after the mutation. *Single deletion mutant* (also known as *single gene knockout*) is an important kind of gene mutant, where one gene is knocked out from an organism. In a single mutant dataset, each entry contains the logarithm of the ratio of the expressions of a gene after the gene knockout to that of the same gene before the gene knockout [13]. Let \( e'_{h,j} \) and \( e_{h,j} \) denote the expressions of the gene \( g_h \) after and before the mutation of \( g_h \) respectively. We define the genetic profile of the organism when gene \( g_h \) is mutated by \( X_h = \{x_{h,j} | x_{h,j} = \ln(e'_{h,j}/e_{h,j}), j \in \{1, 2, \cdots, M\}\} \). Let \( H \subseteq \mathcal{G} \) be the set of genes that have been mutated in total. We define the single mutant genetic profile of \( N \) genes as \( \mathcal{X} = \{X_h | g_h \in H\} \).

3. **SYNTHETIC GENE ARRAY.** An SGA is a set of triples, \( \mathcal{T} = \{(g_i, g_j, t_{i,j}) | i, j \in \{1, 2, \cdots, M\}, i \leq j\} \), where \( t_{i,j} \) is a real number that corresponds to the ratio of the observed fitness to the expected fitness when the organism has both gene \( g_i \) and \( g_j \) knocked out. A value with a large magnitude implies a potential SSL edge. A positive and a negative value stand for alleviating and aggravating relations, respectively.

**Problem Formulation.** Given a gene network \( \mathcal{G} \), the single mutant dataset \( \mathcal{X} \) and the SGA dataset \( \mathcal{T} \), find the mapping \( \Upsilon : \mathcal{X}, \mathcal{G} \longrightarrow \mathcal{T} \) which minimizes a predetermined risk function.

Risk function is a measure of expected miss prediction rate. In this chapter, while estimating the mapping function \( \Upsilon \), we minimize least square error in order to minimize expected miss prediction rate. Based on the mapping learned, we would predict the GI score \( t_{i,j} \) for a new double mutant whose two genes \( g_i \) and \( g_j \) have been mutated.

**6.1.2 Between Pathway Conjectures**

In this section, we describe our two conjectures that are central to SSLPred and the rationale for them. These two conjectures are built on the concepts of BPMs.
Incorporating the structure and properties of BPMs into our model to improve its prediction accuracy was the motivation behind these conjecture.

**Conjecture 1.** Let $B$ denote a BPM, consisting of two pathways $G_A$ and $G_B$. Also, consider an SSL edge $S = \{g_a, g_b\}$ such that $g_a \in G_A$ and $g_b \in G_B$. Then, mutating $g_a$ will significantly alter the expressions of many genes in $G_A$ and $G_B$.

Since, $g_a$ is connected to most other genes in $G_A$ through physical interactions, altering the expression level of $g_a$ will affect the expression of all the genes connected to $g_a$ as they regulate each other. This effect will propagate through the gene network and eventually may change the expressions of many genes in $G_A$. Eventually $g_a$ will severely affect $G_A$ and prohibit it from working properly. Since $G_A$ and $G_B$ constitute a BPM, $G_B$ will compensate this loss by changing the expression of the genes in $G_B$. Thus, mutating $g_a$ eventually changes the expressions of the genes in both $G_A$ and $G_B$.

From this conjecture, we conclude that there is a mapping between an SGA entry $(g_a, g_b, t_{a,b})$ and the corresponding single mutant dataset $X_h$, $g_h \in \{g_a, g_b\}$. This implies a non-trivial mapping, if the SGA entry corresponds to an SSL or epistatic relationship and we have a higher chance to find both $g_a$ and $g_b$ embedded in two pathways of a BPM. In that case, most of the genes in that BPM are supposed to have their expression changed in the single mutant dataset and an appropriate regression method can correlate the changes in the single mutant gene expressions and the corresponding GI score.

Before stating the second conjecture, we define a relevant term, *neighbor*. We say that, gene $g_b$ is an $r$th layer incoming neighbor of gene $g_a$ in the directed gene network, if the shortest path from $g_b$ to $g_a$ consists of $r$ directed edges. In that case, $g_a$ is a $r$th layer outgoing neighbor of $g_b$. Figure 6-2 depicts the incoming and outgoing neighbors for the two genes in a genetic interaction.

**Conjecture 2.** Let $B$ denote a BPM, consisting of two pathways $G_A$ and $G_B$. Consider an SSL edge $S = \{g_a, g_b\}$ such that $g_a \in G_A$ and $g_b \in G_B$. If the expression
Figure 6-2. This figure depicts the layered neighbor structure around a gene interaction edge \((g_a, g_b)\). \(N_{IN}(g_h, r)\) denotes the set of incoming neighbors of gene \(g_h\) at layer \(r\). The set outgoing neighbor \(N_{OUT}(g_h, r)\) is defined similarly. The example contains only up to 2 layers for each direction and gene. The dotted rectangles denote the putative BPM \((G_A, G_B)\) around the gene interaction edge.

of \(g_a\) changes significantly, then in \(G_A\) expression change is most prominent in the first layer of neighbors of \(g_a\) and gradually decreases with increasing layers. Similarly, in \(G_B\) the effect is most prominent for \(g_b\) and gradually decreases with increasing layers of neighbors.

In brief, our conjecture is that the effect of a gene knockout eventually wanes away through the gene network. The rationale behind this is that the neighbors that are close to \(g_a\) and \(g_b\) have a higher chance of being connected only to the nodes of \(B\). The genes in the distant neighborhood have a greater possibility to take part in other pathways. Hence, the closer nodes are more susceptible to undergo a major effect, while the distant neighbors are supposed to be partially screened from that affect due to their activity in the other pathways. Based on these two conjectures, we build a regression based model that we describe in the next section.

### 6.1.3 Regression Based Solution

This section describes the customized regression based approach that we developed to build the mapping \(\gamma : \mathcal{X}, \mathcal{G} \rightarrow \mathcal{T}\), where \(\mathcal{X}\), \(\mathcal{G}\) and \(\mathcal{T}\) denote the single mutant gene expression, gene network and GI score, respectively. Based on the
two conjectures in Section 6.1.2, we extract a set of features for training and testing samples.

We start from the SGA and for each entry \((g_a, g_b, t_{a,b})\), we create a sample point provided either \(g_a\) or \(g_b\) has been mutated in the single mutant dataset available to us, otherwise we discard that SGA entry. Without losing the generality, assume that \(g_a\) has been mutated in this case. Thus, we extract the feature functions from the single mutant data \(X_a = \{x_{a,1}, x_{a,2}, \cdots, x_{a,M}\}\). In designing the set of features, we leverage the information from gene networks by incorporating the two conjectures in our solution. According to the first conjecture, the mutated gene is suppose to perturb only the genes in the host BPM. Thus, while processing the SGA entry \((g_a, g_b, t_{a,b})\), we consider only the genes from \(G_A\) and \(G_B\) and discard the ones from \(G - (G_A \cup G_B)\). We use the GI score \(t_{a,b}\) as the label of the training sample.

Note that while we create the features for a training point, all the data we have is the single mutant data, GI scores and the gene networks. However, for a specific pair of genes \((g_a, g_b)\) we do not know the set of genes that consists of the putative BPM \(B = (G_A, G_B)\) around the gene pair. Rather, we are suppose to validate that information using our model. In fact, if the SGA entry does not correspond to an SSL, there may not be a real BPM for the pair \((g_a, g_b)\). To circumvent this problem, we assume the BPM as part of our model rather an input to the model.

Specifically, we use the concept of \(r\)th layer neighbors, introduced in Section 6.1.2. Let \(R\) represent the maximum number of layers to construct \(G_A\) and \(G_B\). (Usually, \(R\) will be set by the user.) Let us denote the \(r\)th layer incoming and outgoing neighbors of gene \(g_h\) by \(N_{IN}(g_h, r)\) and \(N_{OUT}(g_h, r)\) respectively. Figure 6-2 demonstrates the layered structure of incoming and outgoing neighbors for a pair of genes. We define the putative BPM pathway for \(g_h\) as the union of the sets of incoming and outgoing neighbors of \(g_h\).
up to the layer \( R \)th given by,

\[
G_H = \bigcup_{r=1}^{R} (N_{IN}(g_h, r) \cup N_{OUT}(g_h, r))
\]  \hspace{1cm} (6-1)

In a comprehensive SGA data each GI score is a real valued number that varies in the range of two small numbers such as -5 to +5. However, if the score \( t_{a,b} \) has a small magnitude (close to zero), the gene pair \( (g_a, g_b) \) may not have an SSL/epistatic interaction and may not be part of a BPM. Since the GI score \( t_{a,b} \), which is the label of the regression model is real valued, we still shall use this sample point to train our model. However, the regression model is expected not to discover any interesting pattern of a BPM in the gene expression, and will adjust its parameters accordingly.

To incorporate the second conjecture, we design the features of the regression in a layered approach that directly depends on the concept of layered neighbors introduced in Section 6.1.2. We denote the feature function associated with the incoming neighbors of layer \( r \) of gene \( g \) by \( \Psi_{IN}(N_{IN}(g, r)) \) and the corresponding regression parameter by \( w_{IN}(g, r) \). Similarly, the feature function and parameters for the \( r \)th layer outgoing neighbor are given by \( \Psi_{OUT}(N_{OUT}(g, r)) \) and \( w_{OUT}(g, r) \), respectively. Thus, for \( J \in \{IN, OUT\} \) we state that the feature function \( \Psi_J(N_J(g_c, r)) \) corresponds to neighbors of gene \( g_c \) in direction \( J \) at layer \( r \). Given that \( g_h \) has been knocked out and we are considering the neighborhood of \( g_c \), \( \Psi_J(N_J(g_c, r)) \) can be defined as follows,

\[
\Psi_J(N_J(g_c, r)) = \frac{\sum_{g \in N_J(g_c, r)} |x_{h,i}|}{|N_J(g_c, r)|}
\]  \hspace{1cm} (6-2)

We define another feature function for \( g_b \) by \( \Psi(g_b) \) and the corresponding parameter by \( w \). However, we do not create any feature function to capture the expression of \( g_a \), since \( g_a \) is mutated and its expression may not be available for inspection. Finally, we create the last parameter \( w_{0} \) that acts as a bias constant in the model. Table 6-1 summarizes the feature functions and the corresponding parameters. By aggregating all
Table 6-1. Summarization of the feature functions of the regression model and the corresponding parameters. Feature function represents the set of different features for the regression. A parameter quantifies the strength of the corresponding feature function.

<table>
<thead>
<tr>
<th>Feature Function</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi_{IN}(N_{IN}(g, r))$</td>
<td>$w_{IN}(g, r)$</td>
<td>For incoming neighbors of $r$th layer for gene $g$.</td>
</tr>
<tr>
<td>$\psi_{OUT}(N_{OUT}(g, r))$</td>
<td>$w_{OUT}(g, r)$</td>
<td>For outgoing neighbors of $r$th layer for gene $g$.</td>
</tr>
<tr>
<td>$\psi(g_b)$</td>
<td>$w$</td>
<td>For gene $g_b$ when considering $(g_a, g_b)$, $g_a$ is knocked out.</td>
</tr>
<tr>
<td>$w_0$</td>
<td></td>
<td>A constant representing the bias of the regression.</td>
</tr>
</tbody>
</table>

For these feature functions, we can fit the SGA entry $(g_a, g_b, t_{a,b})$, where $g_a$ is knocked out in the single mutant data as,

$$y_{a,b} = w_0 + w\psi(g_b) + \sum_{r \in \{1,2,...,R\}, \quad J \in \{IN,OUT\}, \quad c \in \{a,b\}} w_J(g_c, r)\psi_J(N_J(g_c, r))$$  \hspace{1cm} (6–3)

**Parameter Estimation.** When the ratio of number of samples to that of the parameters is small (typically less than 20), the estimated value of the parameters experience high variance due to overfitting of data \cite{118}. To alleviate this problem, we augment a regularization term on top of the regression model. Specifically, we aim to minimize the difference between the parameter values at neighbor levels $r$ and $r+1$, to smoothen the decaying of gene expression change. Formally, the regularization term can be written as,

$$Q = \sum_{r \in \{1,2,...,R-1\}, \quad J \in \{IN,OUT\}, \quad c \in \{a,b\}} |w_J(g_c, r + 1) - w_J(g_c, r)|$$  \hspace{1cm} (6–4)

We augment this regularization term with the objective function when estimating the parameters. Using least square error approach, we estimate the parameters of the regression by minimizing the following,
\[ E = \sum_{a,b \in \{1,2,\ldots,M\}, \ a < b} \left( t_{a,b} - y_{a,b} \right)^2 + \lambda Q \]  (6–5)

A detailed discussion on the simplification of the regularization term can be found in Section 3 of Tibshirani et al. [119]. We use the interior-point method to solve this parameter estimation problem [120]. The value of \( \lambda \) is estimated using five fold cross validation.

After the parameter estimation step of the regression method is complete, SSLPred is prepared to predict the GI score for a test sample. For a pair of test genes, we extract the set of features in the same way as that of a training point. Plugging those extracted features and the estimated parameters in Equation 6–3, we obtain the predicted GI score.

### 6.2 Experiments

In this section, we describe the experiments and discusses the results. Section 6.2.1 describes the datasets we used for the experiments. Section 6.2.2 demonstrates SSLPred with another relevant method recently published by Hescott et al. [31, 121].

#### 6.2.1 Datasets

As implied in Section 6.1.1, we classify the datasets into three different categories, namely, single gene mutant data, SGA data and gene networks. We decided on collecting datasets for \( S. \text{cerevisiae} \), since this is a well researched organism of yeast with extensive datasets available for all these three categories [13, 117, 122]. We extracted BPMs from four studies, which were also used by Hescott et al. to validate their method [121]. We employed these BPMs as gold standards in this chapter. Next, we describe these datasets in detail.

1. **GENE NETWORK BIOGRID.** The dataset maintains one of the most comprehensive gene networks for \( S. \text{cerevisiae} \) [122]. We collected 155,287 the genetic interactions in total from this database.

2. **SINGLE MUTANT DATA.** In this chapter, we collected 287 single gene knockout experiments from the compendium of expression profile of \( S. \text{cerevisiae} \) developed
by Hughes et al. [13]. Each experiment contains 6,316 entries. Every entry contains the the logarithm of after to before ratio of expressions of a gene as described in Section 6.1.1.

3. **SGA DATA.** Costanzo et al. generated a genome scale SGA profile for *S. cerevisiae* with neatly 5.4 millions of genetic interactions out of nearly 75% genes [117]. Out of this comprehensive profile, we selected GI scores for 370,913 interactions such that for every edge, at least one of the two consisting genes was knocked out in the gene knockout experiments.

4. **BPMS.** We obtained four sets of BPMs, all are of *S. cerevisiae*, itemized in the following – *Kelley-Ideker* [30], *Ulitsky-Shamir* [123], *Brady et al.* [124], and *Ma et al.* [125]. We denote a dataset using the authors’ names of the corresponding paper. The numbers of BPMs that contain three or more genes in each pathway in these datasets are 160, 36, 959 and 54 respectively.

### 6.2.2 Comparison with Hescott’s Method

This section describes the comparison between SSLPred and the method proposed by Hescott et al. [31, 121]. Hescott et al. employs microarray expression data of single gene knockout experiments to identify BPMs. Though their method does not predict GI score, according to our knowledge, this is the only published method that integrates the concept of single gene mutants and between pathway motifs.

Before coming to the main discussion, we describe how we create a matrix of predicted GI scores using five-fold cross validation. We divide the 287 knockout experiments into nearly equal five groups, each of them being a $57 \times 6316$ matrix. For each fold of cross validation, we use four out of five groups to create sample training points along with the corresponding GI scores as described in Section 6.1.3. If for a gene pair the corresponding GI score is not available, we discard the that sample point. After training, we create test points from the left-out one group and predict the test scores for them. Repeating this process in a five fold cross validation fashion, we predict GI scores for all possible pairs of genes from the $287 \times 6316$ matrix. Now that we have the predicted GI matrix which we denote by $T_P$, we discuss how we employ it for comparison between SSLPred and the one proposed by Hescott et al.
Consider a BPM $B = (G_A, G_B)$ obtained from a known sets of BPMs. Now, consider a gene $g_x \in G_A$. Hescott et al. ranks all the genes $G$ of the organism with respect to $g_x$. Let us denote that rank by $G_g(g_x)$. Then, from that rank, it retrieves $G_B$ and calculate the quality of retrieved $G_B$ by a scoring method called $ClusterRankScore$. We now describe $ClusterRankScore$ which is adapted from Gene Set Enrichment Analysis [126].
ClusterRankScore accepts an ordered list of genes \( L \) and another set \( C \) as input. Then, it explore the distribution of \( C \) along \( L \). Intuitively, if \( C \) appears at the head or tail of \( L \), it is enriched with the specific properties represented by the ordered list \( L \). In the current context, consider a BPM \( B = \{G_A, G_B\} \). Let us knock out a gene \( g_a \) from \( G_A \) and measure the change of expressions for all the other genes. Hescott et al. now arrays the genes according to its own criteria. Here, this ordered list is \( L \) and the pathway \( G_B \) is \( C \). Thus, a correlation between the ordered gene list and the pathway \( G_B \) implies that the BPM \( B \) is validated by Hescott et al.

Using SSLPred we create a similar rank as follows. We obtain the predicted GI score of all the gene pairs \((g_x, g_y), g_y \in \mathcal{G}\) from the predicted GI matrix \(T_P\). Then we sort \( \mathcal{G} \) in increasing value of the retrieved GI scores of \((g_x, g_y)\). Let us denote the sorted list of genes by \( \mathcal{G}_\psi \). After this we calculate the ClusterRankScore of \( G_B \) based on \( \mathcal{G}_\psi \). Let us denote the ClusterRankScore of \( G_B \) with respect to Hescott et al. and SSLPred by \( CRS_\phi(g_x, G_B) \) and \( CRS_\psi(g_x, G_B) \), respectively.

To calculate the statistical significance of the two ClusterRankScore, we design separate permutation tests for each of them and calculate p-Values with respect to those permutation tests. Here the null hypothesis can be stated as \( B \) is not a BPM. A detailed account of ClusterRankScore and the permutation test can be found at Hescott et al. [31, 121].

Consider a BPM \( B = (G_A, G_B) \). For all the combinations, \((g_x, G_B), g_x \in G_A \) and \((g_y, G_A), g_y \in G_B \) we calculate the p-Values using the procedure described above. For every dataset, we plot the histograms of those p-Values. Since all the BPMs have been obtained from published literature, we assume them to be equivalent to a gold standard.
Figure 6-4. Comparison of SSLPred with the method from Hescott et al. on Brady and Ma datasets for p-Values ≤ 0.1. SSLPred and SSLPred(2) denote variants of SSLPred with at most one or two layers of neighbors, respectively. The X axis, that ranges between zero and 0.1, represents the p-Values of the permutation tests. The Y axis represents the frequencies of the pathways at a particular p-Value of the permutation test. We display histograms for the very two datasets for which our method performs similar or worse in Figure 6-3. The two sub-figures demonstrate that on these two specific datasets, SSLPred maintains a higher frequency at the p-Value ranges between zero and 0.01.

Hence, in the histogram, an increased frequency of the BPMs with lower p-Values corresponds to a better quality of the BPM retrieval method.

Figure 6-3 compares SSLPred with the other method. SSLPred and SSLPred (2) denote variants of SSLPred with at most one or two layers of neighbors, respectively. The X axis, that ranges between zero and one, represents the p-Values of the permutation tests. The Y axis represents the frequencies of the BPMs at a particular p-Value of the permutation test. SSLPred outperforms Hescott et al. by 100%, 31%, 2% for Ulitsky, Kelley and Brady dataset when the p-Value is equal to or smaller than 0.1. For SSLPred (2) the corresponding numbers are 46%, 71% and -12% respectively. For Dataset Ma, Hescott et al. is better by 48% and 12% than SSLPred and SSLPred (2) respectively. If we relax the p-Value to 0.3 we observe that SSLPred (2) performs better than Hescott et al. by 24%, 12%, 10% and 48% for Ulitsky, Brady, Ma and
Kelley respectively. It can be concluded that that apart from on Ma dataset, SSLPred outperforms Hescott’s method, since it maintains a higher frequency at the p-Value ranges between zero and 0.1. For Kelley and Ulitsky dataset SSLPred outperforms with a high margin between 0 to 0.1 p-Value range. For Ma and Brady datasets, the two methods perform very competitively on an average. Also, its is difficult to compare between two variants of SSLPred. Though these two variants are in close competition, SSLPred (2) has a slightly better advantage over SSLPred. Table 6-2 summarizes the results in Figure 6-3 by tabulating the percentage of BPMs with p-Values less than or equal to 0.1.

Figure 6-4 highlights a special case of the frequency distribution when the p-Value is restricted to be less than or equal to 0.01. We are specifically interested in the very two datasets Brady and Ma for which our method performs similar or worse when we restrict the p-Values to 0.1. Since a lower p-value implies a lower chance of false positive detection, these results are important in determining the superiority of the competing methods. Here also we observe that SSLPred (2) has a better accuracy compared to Hescott et al. in identifying larger number of small p-Value BPMs. For Brady dataset SSLPred (2) outperforms the other method by 177%. For Ma dataset both of them detect two BPMs. This concludes that, our method demonstrates superior accuracy in validating more BPMs with very low p-Values (≤ 0.01). We also conducted a third set of experiments SSLPred (3) with highest layer of neighbors $R = 3$. However, SSLPred (3) performed poorly compared to Hescott et al. We believe that most BPMs are of small sizes with a diameter of less that equal to four edges as indicated in Kelley et al. [32]. Hence, assuming a bigger size BPMs with $R = 3$ compromises the accuracy of our method.

**Code.** All the code developed in this chapter is available from [http://bioinformatics.cise.ufl.edu/projects/SSLPred.html](http://bioinformatics.cise.ufl.edu/projects/SSLPred.html).
6.3 Discussion

In this chapter, we developed a new method SSLPred to predict SSL interactions in an organism. Our method is built on the concept of Between Pathway Models, where majority of the SSL pairs span across the two functionally complementing pathways. We developed a regression based approach that learns the mapping between the gene expressions of single deletion mutant to the corresponding synthetic gene array.

We compared our method to the one by Hescott et al. for predicting the GI scores of S. cerevisiae on four benchmark datasets. On different experimental setups, on average SSLPred performed significantly better compared to the other method.
CHAPTER 7
CONCLUSION

Analyzing the effect of perturbation on gene networks in a complex problem due to different reasons including but not limited to non-homogeneous network structure, different kinds of genetic interactions, characteristics associated to a specific perturbation and small number of samples due to cost constraint.

Due to complex and diversified nature of this problem, we selected four subproblems that well represent different aspects of the original problem, however are not necessarily comprehensive in nature. Also, while solving the problems, we made several assumptions to simplify the problems. For example, in the first and second chapters we assume that there should be only two time points (control and non control) in the gene expression data.

Despite this limitations, there is no denying the fact that this work has been able to substantiate the hypothesis that it claimed in the beginning. In all four of the chapters, our methods demonstrated the merit of the hypothesis that the integration of gene networks enables us to build superior descriptive and predictive model.

In the future, I would like to leverage the hypothesis proposed in this work in relevant problems. I believe that not only in the present context, but also in other relevant areas integration of domain specific knowledge to data in a Bayesian fashion will help us to build more accurate and useful models.


[95] Hammersley JM, Clifford P: Markov fields on finite graphs and lattices 1968. [Unpublished manuscript].


BIOGRAPHICAL SKETCH

Nirmalya Bandyopadhyay received his Bachelor of Engineering degree in Computer Science from Bengal Engineering and Science University, India. He spent three years in the industry in the area of computer networks and EDA tool development. He received his Ph.D. from the School of Engineering in the University of Florida in the fall of 2011. During his Ph.D. he worked in the areas of computational biology and machine learning under the supervision of Dr. Sanjay Ranka and Dr. Tamer Kahveci and was associated with the bioinformatics lab. Nirmalya authored two book chapters, four journal papers and four conference papers. In 2010, Nirmalya worked as a development intern in Amazon.com.