INTEGER LINEAR PROGRAMMING BASED SOLUTIONS FOR CONSTRUCTION OF BIOLOGICAL NETWORKS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖYKÜ EREN ÖZSOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
HEALTH INFORMATICS

JUNE 2014

INTEGER LINEAR PROGRAMMING BASED SOLUTIONS FOR CONSTRUCTION OF BIOLOGICAL NETWORKS

Submitted by **ÖYKÜ EREN ÖZSOY** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Health Informatics, Middle East Technical University** by,

Prof. Dr. Nazife BAYKAL
Director, **Informatics Institute** ________________________________

Assist. Prof. Dr. Yeşim AYDIN SON
Head of Department, **Health Informatics** ________________________________

Assoc. Prof. Dr. Tolga CAN
Supervisor, **Computer Engineering, METU** ________________________________

Examining Committee Members:

Assist. Prof. Dr. Yeşim AYDIN SON
Health Informatics, METU ________________________________

Assoc. Prof. Dr. Tolga CAN
Computer Engineering, METU ________________________________

Assist. Prof. Dr. Aybar Can Acar
Health Informatics, METU ________________________________

Assist. Prof. Dr. Özlen Konu
Molecular Biology and Genetics,
Bilkent University ________________________________

Assoc. Prof. Dr. Hasan OĞUL
Computer Engineering, Başkent University ________________________________

Date: 06.06.2014

ABSTRACT

INTEGER LINEAR PROGRAMMING BASED SOLUTIONS FOR CONSTRUCTION OF BIOLOGICAL NETWORKS

Eren Özsoy, Öykü
Ph.D., Department of Health Informatics
Supervisor: Assoc. Prof. Dr. Tolga Can

June 2014, 105 pages

Inference of gene regulatory or signaling networks from perturbation experiments and gene expression assays is one of the challenging problems in bioinformatics. Recently, the inference problem has been formulated as a reference network editing problem and it has been show that finding the minimum number of edit operations on a reference network in order to comply with perturbation experiments is an NP-complete problem.

In this dissertation, we propose linear programming based solutions for reconstruction of biological networks. We propose an integer linear programming (ILP) model for reconstruction of signaling networks from RNAi data and a reference network. The ILP model guarantees the optimal solution; however, is practical only for small networks of size 10-15 genes due to computational complexity. In order to scale for large networks, we propose a divide and conquer based heuristic, in which a given reference network is divided into smaller sub-networks that are solved separately and the solutions are merged together to form the solution for the large network. However the solution that we have developed for reconstruction of signaling networks using RNA interference data is not suitable for networks with multiple sources and sinks. In order to handle such networks, we use gene expression data and develop another ILP based graph theoretical method.

We validate our proposed approaches on real, semi-synthetic and synthetic data sets, and comparison with the state of the art shows that our proposed approaches are able to scale better for large networks while attaining similar or better biological accuracy.

Keywords: regulatory networks, network topology, linear optimization, RNAi, gene expression

ÖZ

BİYOLOJİK AĞLARIN OLUŞTURULMASI İÇİN TAM SAYILI DOĞRUSAL PROGRAMLAMA TABANLI ÇÖZÜMLER

Eren Özsoy, Öykü
Doktora, Tıp Bilişimi Bölümü
Tez Yöneticisi: Doç. Dr. Tolga Can

Haziran 2014, 105 sayfa

Gen düzenleme veya sinyal ağlarının pertürbasyon deneyleri ve gen ifade incelemesiyle çıkarımı biyoenformatikteki zor problemlerden birisidir. Yakın zamanda, çıkarım problemi referans ağ biçimlendirmesi problemi olarak formüle edilmiş ve pertürbasyon deneyleriyle uyumlanması için referans ağ üzerinde gerçekleştirilen minimum biçimlendirme işlem sayısının bulunmasının NP-Tam bir problem olduğu gösterilmiştir.

Bu doktora tezinde, biyolojik ağların yeniden yapılandırılması için doğrusal programlama tabanlı çözümler önerilmiştir. RNA engelleme (RNAi) verisi ve referans ağ kullanılarak, sinyal ağlarının yeniden yapılandırılması için tam sayılı doğrusal programlama (TDP) modeli geliştirilmiştir. TDP modeli optimal çözümü garanti etmektedir ancak hesaplama karmaşıklığından dolayı yalnızca 10-15 genden oluşan küçük ağlar için elverişlidir. Büyük ağları ölçeklendirmek için böl ve yönet tabanlı yeni bir yöntem önerilmiştir. Bu yöntemde verilen referans ağ, ayrı ayrı çözülen alt ağlara ayrılmakta ve bu çözümler büyük ağın çözümünü oluşturmak için birleştirilmektedir. Fakat RNAi verisini kullanarak sinyal ağlarının yeniden yapılandırılması için geliştirdiğimiz çözümler çoklu alıcı/hedef içeren ağlar için uygun değildir. Buna benzer ağların çözümü için gen ifade verisi kullanılmıştır ve yeni bir TDP tabanlı çizge teorik yöntem geliştirilmiştir.

Önerilen yöntemler; gerçek, yari sentetik ve sentetik verilerle doğrulanmıştır ve literatürdeki yöntemlerle karşılaştırılması, önerilen yöntemlerin büyük ağlara ölçeklenmede benzer ya da daha iyi biyolojik doğruluk elde ettiğini göstermiştir.

Anahtar kelimeler: Düzenleyici ağlar, ağ topolojisi, lineer opimizasyon, RNAi, gen ekspresyonu

dedicated to my husband, İstemi Barış Özsoy

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Dr. Tolga Can for being a tremendous mentor for me throughout my studies. His excellent guidance and enthusiasm not only in research but also in everyday life provided me everything I needed to accomplish this study. I am grateful for his availability as a supervisor regardless of time and location. I feel myself honored as he accepted me to be a Ph.D. student of him and hope to have a long-term collaboration with him thereafter as a colleague.

I would like to thank to Dr. Yeşim Aydın Son and Dr. Hasan Oğul for their suggestions, precious time and support. And also special thanks go to Dr. Aydın Son, who was participated in the study described in the last chapter of this dissertation with her findings on mouse-agouti data set. Without her comments, this chapter would not be materialized.

The studies presented in Chapter 2 and Chapter 3 of this dissertation are supported by the European Union in the 7th framework program through SysPatho (Grant No: 260429); and the study presented in Chapter 4 is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) 1002 Short Term R&D Funding Program, (Grant No: 113E323). I greatly acknowledge their valuable supports which helped me to accomplish these studies. And also, I acknowledge the High Performance Computing (HPC) facilities of the Department of Computer Engineering at Middle East Technical University were used to perform the experiments.

I really enjoyed working in the Bioinformatics and Computational Biology Laboratory, Department of Computer Engineering, METU, and would like to thank to my friends in the lab, Ayşe Gül Yaman and Ömer Tarı. I will miss our discussions.

I want to thank to my dear friend A. Burçak Sönmez. Without her, those years would not be enjoyable. I appreciate M.D. Alexis K. Okoh’s unforgettable friendship and his willingness to keep my motivation up.

I want to thank to my parents, Neriman and Abdullah Eren and and my brother, Çağlar Eren for their love and support during my whole life. Last but not least, to my lovely husband, Istemi Barış, who is always with me and makes me feel happy and confident, especially at the hectic times, and believing in me at all times; thank you for your endless love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Related Work

The human body is composed of countless number of cells, which are the smallest structural components of all organisms. Each cell contains thousands of genes and the interactions between them carry out specialized functions, understanding of which are crucial from medical and biological perspective. Yet, these functions, such as gene regulation, are not yet fully understood due to their complexity.

Systems biology deals with estimating the relationships between the genes in signal transduction networks and gene regulatory networks. The structure that the genes form and the interactions between them can be explained basically by networks. Each gene in the cell is represented by a node and the interactions between these genes are depicted by edges in the network. Although there are several studies in the literature that estimate networks from perturbation effects, they are either not accurate enough or cannot scale for large networks since it is a very challenging task. RNA interference (RNAi) is a perturbation technique used for finding the genes in a pathway (Fire et al., 1998). It opens new perspectives for network reconstruction methods. Although RNAi is a useful technique to identify genes associated with a particular phenotype, the temporal and spatial placement of these genes in the respective cellular pathways remains a problem (Moffat and Sabatini, 2006). Markowetz et al., 2007, proposed Nested Effect Models for the construction of signal transduction networks. Such models construct the signal transduction networks by using the nested structure of observed perturbation effects. Although they are suitable models for effect-result type of data, such as RNAi data, they require several types and relatively high number of readouts per knockdown. This prevents the usage of the results of RNAi experiments using one messenger gene.

It is possible to place genes in the respective pathways by interrogation of databases and literature (König et al., 2008) however, in the case when there is insufficient information about the genes in the pathways, automated approaches that place these genes in the network have to be developed. Kaderali et al., 2009, develop a probabilistic method that can reconstruct network topologies using single gene knockdown data. However, because of the computational complexity of this method, it can be applied to only small networks (8-10 genes). The method developed by Hashemikhabir et al., 2012, constructs relatively accurate networks satisfying RNAi data; however, the method does not always include all the genes in the constructed network. To overcome such difficulties, our first motivation is to develop methods that reconstruct signaling networks from RNAi data with high accuracy and for large networks.

We first propose an Integer Linear Programming (ILP) based approach that reconstructs the optimum network by applying minimum number of edit operations on a given reference network taken from the literature. The product network is ensured to satisfy the RNAi data by the formulation of the problem. Although this approach yields very accurate results, it fails to scale for large networks and can reconstruct networks with 10-12 genes, which constitutes the basis of our second motivation. Therefore, we extend our approach to deal with large networks employing the divide and conquer technique.

Despite their success, these methods are convenient only for the networks with a single receptor gene and a single reporter gene. Since there are multiple receptor and reporter genes in real biological networks these methods are not suitable for such networks. Our third motivation is to overcome this problem by developing another ILP based graph theoretical method using gene expression data. In this method, we use time-series data consisting of gene expression measurements at different time periods. Such data allows us to track the changes happening in the network, and thus to find the interactions between genes. To improve the results, we integrate perturbation data with time-series data. Since these data contain noise due to the nature of the experiments, we propose additional methods to reduce the effects of noise.

1.2 Biological background

This section provides the necessary biological and mathematical background for understanding the material and methods presented in this dissertation.

1.2.1 Biological networks

Understanding biological networks is essential from a medical or biological perspective. Network inference is crucial for interpreting the topology of the networks. Modeling the relationship between signal transduction networks, gene regulatory networks, and protein-protein interaction (PPI) networks is a very important problem in systems biology. This section gives brief information about some major biological networks, which were used in this study, signaling networks, gene regulatory networks and PPI networks.

A network can be defined as a collection of nodes and edges. In biological networks, nodes may be either a gene or a gene product and edges correspond to the interactions between these nodes. We use the terms protein and gene interchangeably in the rest of the study, sometimes using the term gene to indicate a gene product.

Signaling networks are represented as directed networks structurally. Each node of the network represents either a protein or a gene, and the interactions between these proteins/genes are represented by edges of the network. These interactions are directed edges and the direction of an interaction depicts the direction of the signal flow. The signal is usually received at a single receptor protein and is transduced to a target gene.

Signal transduction is a series of biochemical reactions involving proteins; therefore some of the physical interactions in PPI networks may indicate signaling processes. Exploiting this relationship, PPI networks can be used as reference networks, i.e., as initial skeleton networks, for reconstruction of the topology of signaling networks (Hashemikhabir et al. 2012). While signaling networks are modeled as directed graphs, high-throughput protein-protein interaction data is undirected. It is possible to transform this undirected data to directed pathways (Gitter et al., 2011).

Gene regulatory networks organize gene regulation which is simply the decision given by the cell to turn on or turn off the genes so as to control gene expression. They are similar to signal transduction networks in forming signaling cascades. Any gene can activate many others and large number of genes can be expressed as a result of an initial stimulus.

There are several models developed for reconstruction of signaling and gene regulatory networks using various types of high throughput data. The type and amount of data, prior knowledge for the network, experimental and computational resources play an important role in the development of these models. In this study, RNA interference data and microarray data are used to solve the network inference problem. Genes are represented by nodes and edges refer to the interactions between a gene product and its receptor gene.

1.2.2 RNA interference Data

RNA interference (RNAi) is a research tool that is used for a high-throughput analysis of gene function at the genome-scale. This powerful technique allows researchers to identify complex biological processes in the cells of the organisms ranging from plants to mammals. The introduction of the RNAi technique had a revolutionary effect on the studies of several biological processes, such as signal transduction, infection responses, cancer biology, etc (Moffat and Sabatini, 2006; Rao et al., 2013; Sarkies and Miska, 2013; Prados et al., 2013). It can be used as a therapeutic approach for several diseases where there is an abnormal problem in protein production and to inhibit the expression or replication of pathogenic viruses, such as HIV (Rossi, 2006) and hepatitis C virus (Seo et al., 2003).

Traditionally, double-stranded RNAs (dsRNAs) are used to knockdown, i.e. reduce the activity of the genes of particular interest and the change on the cellular activity is observed (Fire and Mello, 1998). The dsRNA is divided into small duplex RNAs which consist of approximately 21–23 nucleotides siRNAs (small interfering RNAs) by an enzyme which is called Dicer. Interaction of siRNAs with the RNA-induced silencing complex (RISC) which is a multiprotein complex, results in sequence specific association of the activated RISC complex with the messenger RNA transcript (Figure 1.1). This interaction results in sequence-specific cleavage of the target transcript and thus, the production of the unwanted protein is prevented by avoiding its translation.

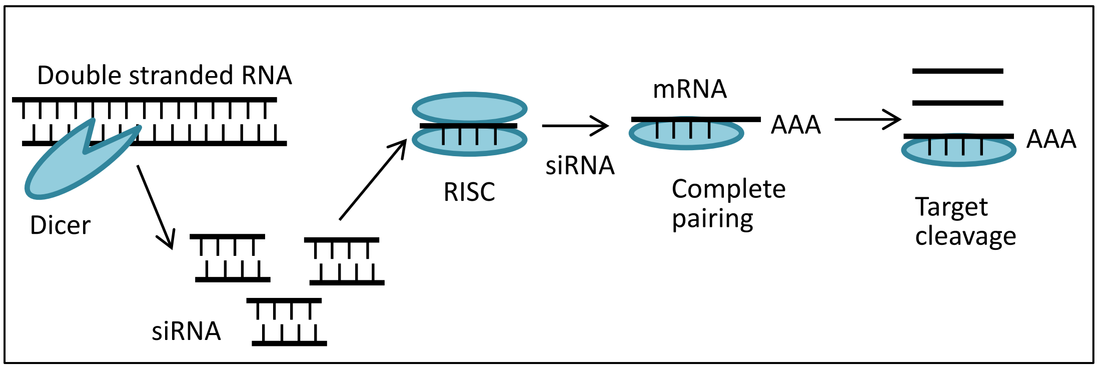


Figure 1.1 The mechanism of gene silencing by RNAi interference.

The genes between the receptor and reporter genes in a signaling network have varying levels of influence on the transduced signal. In this study, we model this influence in a binary manner, and label the intermediate genes as critical or non-critical. By using RNAi data, a gene can be resolved to be a critical gene or a non-critical one. RNAi data is generated from RNAi experiments which can be used to find the genes in a pathway (Fire et al., 1998). For RNAi screens at large-scale, the readouts are established upon single reporters (Brass et al., 2008). Genome-wide screens that have high contents are developing with an increasing rate with the advances in technology (Sacher et al., 2008). RNAi screens are used to investigate the downstream effects of a silenced gene. In these experiments, every other gene except the receptor and the reporter gene in the system are knocked down and the expression level of the reporter gene is measured. The genes which influence the reporter gene significantly are called “critical genes” and the rest are called “non-critical genes”. The result of a knock-down is usually measured as the level of differential expression (e.g., fold change) of the reporter gene (Friedman and Perrimon, 2006) and called the RNAi score. By using RNAi scores, the order of the genes in a signaling network can be predicted (Singh, 2011).

1.2.3 Microarray data

The functionality of a gene can be determined by measuring the amount of mRNA production during DNA transcription, in a cell. This can be done by using microarrays, which can detect the mRNA levels of thousands of genes in a single experiment. Microarrays, or gene-chips, consist of DNA spots attached to a surface consisting of specific DNA sequences. When a gene is activated, it produces mRNA which is complementary. Therefore, it is able to bind to the original portion of the DNA strand from which it was copied. To determine the activated or de-activated genes, the mRNA molecules are collected from a cell in consideration. Then, these mRNA molecules are labeled by using an enzyme which generates the so-called complementary DNA (cDNA) to the mRNA. Fluorescent agents are attached to the cDNA during this process. Then, these labeled cDNAs are placed onto the microarray, and they bind to their complementary DNAs on the microarray with fluorescence. Measuring the intensity of fluorescence at each spot on the microarray using a scanner, activity level of a gene can be obtained.

Depending on the type of experiment, microarray data can be either time-series or perturbation data. Time series data gives information about the time-change of gene expression levels. By using this data, the genes which are activated in a biological event can be discovered. The dynamics of the cellular system, as well as the order of the genes and their causal effects can be determined. Perturbation experiments are similar to RNAi experiments in that the activity of a gene is perturbed and its effects to the cellular activity are studied. Microarray data is available in terms of a gene expression matrix, which is a table of rows representing gene names and columns representing the individual sample. The columns can be either time points or treatments depending on the experiment type. The expression profile for a gene is accessed from the corresponding row on the matrix.

1.3 Mathematical background

Network inference problem is a kind of problem that figures out the placement of genes and the interactions between them. There are several models in the literature to solve this problem. In this study, the proposed model is formulated as a linear optimization problem where the objective function is defined as the minimization of applied changes on a given reference network.

1.3.1 Linear programming

Linear programming is a method to solve an optimization problem where the objective function and the constraints are linear functions of variables. A linear function *f* is defined by:

$$f(x_1, x_2, ..., x_n) = a_1 x_1 + a_2 x_2 + ... + a_n x_n = \sum_{j \in J} a_j x_j \qquad (1.1)$$

where a_j are real numbers and x_j are the variables. Linear constraints are the linear equalities or linear inequalities in the form

$$f(x_1, x_2, ..., x_n) = b \tag{1.2}$$
$$f(x_1, x_2, ..., x_n) \leq b \tag{1.3}$$
$$f(x_1, x_2, ..., x_n) \geq b \tag{1.4}$$

where *b* is a real number. Strict inequalities, i.e. "larger than" or "smaller than", are not allowed in linear programming.

Linear programs are represented in two forms; standard form and slack form. In standard form, a linear function is maximized subject to linear inequalities, whereas in slack form, it is maximized subject to linear equalities. One of the methods for solving linear programs is the simplex method. In order to solve a linear problem by simplex method, it is represented in slack form. A linear program subject to linear constraints can always be converted to standard form.

A linear program in standard form is shown as follows:

$$\text{Maximize } \sum_{j \in J} c_j x_j \tag{1.5}$$

Subject to

$$\sum_{j \in J} a_{ij} x_j \leq b_i \quad \text{for } \forall\, i \in I \tag{1.6}$$
$$x_j \geq 0 \quad \text{for } \forall\, j \in J \tag{1.7}$$

where c_j and b_i are real numbers with $I = \{1, 2, \ldots., m\}$ and $J = \{1, 2, \ldots., n\}$, and x_j are the real numbers that are to be found. Expression (1.5) is called the objective function, and (1.6) and (1.7) are called the constraints. Inequalities (1.7) are also called nonnegativity constraints. Not all linear programs (LPs) need nonnegativity constraints but they have to be shown if LP is represented in standard form. Any setting of x_j's satisfying all the constraints (1.6)-(1.7) is called a feasible solution to the LP.

In order to solve an LP by the simplex method, it is required to represent the LP in slack form, where all the constraints other than the nonnegativity constraints are converted to equalities.

Consider the inequality constraint

$$\sum_{j \in J} a_{ij} x_j \leq b_i \quad \text{for } \forall\, i \in I \tag{1.8}$$

Introducing new variables x_{n+i}, i.e. the slack variables, this constraint is converted to

$$x_{n+i} = b_i - \sum_{j \in J} a_{ij} x_j \tag{1.9}$$

$$x_{n+i} \geq 0 \tag{1.10}$$

By this conversion, we obtain an equivalent linear program in which the only inequality constraints are the nonnegativity constraints. The variables x_{n+i} are called the basic variables and the x_j are called the nonbasic variables. Removing the maximize and subject to words, adding an optional constant v to the objective function and denoting the objective function as z, the slack form can be expressed as

$$z = v + \sum_{j \in J} c_j x_j \tag{1.11}$$

$$x_i = b_i - \sum_{j \in J} a_{ij} x_j \text{ for } \forall\ i \in N \tag{1.12}$$

where N denotes the index set of nonbasic variables and all variables x are nonnegative.

1.4 Contributions

In this dissertation, we focus on the biological network inference problem. Reconstruction of gene regulatory networks and signaling networks from perturbation and gene expression assays is a challenging task. Finding the relationships between the genes in signaling networks and gene regulatory networks is very important to understand the whole system biologically in systems biology. However, the computational costs of solutions increase exponentially with the increasing number of genes in the system. Therefore construction of large scale networks is a challenging problem. Our contributions in this study can be grouped under three main categories; reconstruction of small size, single receptor single reporter networks optimally, large scale network reconstruction with single receptor and single reporter networks, and large scale network reconstruction with multiple receptor multiple and reporter networks.

We propose linear programming based solutions for the network inference problem. Our first approach is to formulate this problem as an integer linear optimization problem, which will provide a network satisfying the RNA interference (RNAi) data with a minimum change made on a given reference network. The genes in the network are known *a priori* according to our assumption. The given reference network may be taken from any protein-protein interaction (PPI) network database or it may be a literature curated network, where each node represents a protein or a gene and also we have RNAi data from RNAi experiment results related with the studied network topology. In fact, a reference network is not required but it forms a good skeleton to the reconstructed network. Due to the nature of RNAi screening experiments there may be some errors in the RNAi data. Moreover, there may be some errors in the reference network depending on the experiment it has been obtained. These errors may cause some contradictions between the given reference network and RNAi data. Using RNAi data and the reference network together as a complementary data may reduce such errors. For our approach, we show that as the number of nodes in a network increases, the number of constraints increases exponentially. We explicitly derive the number of constraints for several different scenarios. To handle the exponential growth problem, first we use raw RNAi scores to decide the placement of the genes. It helps to reduce the search space of the problem. In order to inspect how RNAi scores affect the number of constraints for a given problem, we considered a 12–node example and showed that it is possible to reduce the number of constraints by 97% depending on the number of critical genes and their ordering. We considered several problems with different reference networks, number of nodes and RNAi data. Our experiments show that, the topology of a network with 10 nodes can be constructed by our approach in a reasonable time. The integer linear programming (ILP) approach guarantees the optimal solution, however; it is not possible to solve the networks having more than 10 nodes in a reasonable time because of the complexity of the problem. Finding solutions for larger networks becomes difficult as the number of constraints increases exponentially.

To obtain solutions for large networks, we propose a heuristic based on the divide and conquer approach. This approach allows us to reduce the number of nodes for a network by dividing it into sub-networks up to 10 nodes, then solve each sub-network separately by our ILP approach, and finally combine the obtained resulting sub-networks at their respective positions to reconstruct the large network. Depending on the given reference network topology (i.e. ratio of the number of critical genes and network size) we develop two different division algorithms. We validate our divide and conquer approaches on real, semisynthetic and synthetic networks. These networks are varying to test our methods performance from sparser to denser, smaller to larger and having specific percentages of noises. Comparison with state of the art methods reveals that our methods can construct networks better in terms of recall and precision measures, scalability, robustness, and consistency with the RNAi data. The divide and conquer approaches are able to scale better for large networks while attaining similar or better biological accuracy compared with the state of the art methods. Moreover, while the state of the art methods cannot include all genes and the interactions related with these genes to the reconstructed network topology, our approach finds connected networks. A connected network can be defined as a network which includes every gene in the network topology and every gene is an element of a directed pathway from the receptor gene to the target gene. For dense and large networks, while the state of the art methods fail to construct networks in 1 hour, we are able to find solutions in minutes. By applying the divide and conquer strategy, we are able to scale our solution to large networks which makes the ILP solution a practical approach. The methods that we developed using RNA interference data are suitable for networks having single receptor and single reporter genes. It is not possible to find solutions for multiple receptor and reporter networks with these methods.

To deal with multiple receptor and reporter networks, we propose an ILP based graph theoretical model using time series gene expression data. Such data allows us to track the changes happening in the network, and thus to find the interactions between genes. The results show that time series data gives the underlying structure of the constructed network. In addition to the time series data, we show that integration of any biological information (such as perturbation data, RNAi data, and novel information about some interactions) to the proposed approach makes it accurate. We applied our approach on synthetic dataset along with different percentage of noise to validate its ability to construct networks on noisy data. Also the mouse agouti signaling network was reconstructed by the proposed approach and showed that several interactions and sub-networks can be found correctly. By using this approach we can solve large scale networks easily.

In summary, in this study, we show that the dimensionality of the network inference problem increases by the increase in the number of genes in the system. We developed an integer linear programming approach for reconstruction of signaling networks and prove that it finds the optimal solution, however; it is not able to scale large networks. To handle this problem we developed a divide and conquer based heuristic, which is a robust approach and guarantees to find a connected network topology. By using the proposed method, we reconstruct networks with hundreds of genes in minutes. It still has a drawback on multiple receptor and multiple reporter gene networks and we deal with this problem by proposing an integer linear programming based graph theoretical model which can solve large scale multiple receptor multiple reporter large scale networks. Our results show that our approach can reconstruct biologically significant networks.

1.5 Outline of the Dissertation

This study can be partitioned into three parts. In the first part we deal with the network inference problem by using linear programming approach along with the RNAi data and a given reference network for small size networks. This method provides solutions for networks up to 10 nodes. We give the proof of exponential growth of the dimensionality of the problem. The integer linear programming approach given in this chapter forms the backbone of our next approach for large scale network reconstruction, which is a divide and conquer based approach mentioned in the second part, it is able to scale for large networks in minutes, however; it reconstruct networks with a single receptor and a single reporter. To handle this drawback, third part presents the integer linear programming based graph theoretical model for reconstruction of multiple receptor multiple reporter gene networks.

In Chapter 2, we give the integer linear programming approach to construct signaling pathways from protein-protein interactions (PPI) and RNA interference (RNAi) data. This chapter details a general discussion on this approach and proves that the computational requirement of the problem grows exponentially by the increasing number of nodes; therefore it cannot scale for larger networks with more than 10 genes in reasonable time. Chapter 3 presents a method to handle this drawback by developing a divide and conquer based heuristic. In Chapter 3, we show that our approach is a robust approach and constructs biologically significant networks. And also we can deal with large networks with hundreds of nodes in minutes and construct the network topology. We compare our approach with the state of art methods and demonstrate our performance. In Chapter 4, we present another integer linear programming approach that uses time series gene expression data for reconstructing multiple receptor multiple reporter networks. We evaluated our study on a real network and demonstrate that our approach is able to reconstruct the sub networks and several interactions common in the actual network topology easily. And finally, in Chapter 5, we represent a brief summary of our study and give some future directions.

CHAPTER 2

RECONSTRUCTION OF SIGNALING NETWORKS FROM RNAi DATA

In this chapter, we present the method we have developed to construct signaling pathways from protein-protein interactions (PPI) and RNA interference (RNAi) data using Integer Linear Programming (ILP) (Eren Ozsoy and Can, 2013). By using this method, which we call thereafter Original Integer Linear Programming (oILP) to not confuse with other ILP based methods that we present in Chapter 3, we solve several example networks and give a comparison between findings of our method and state of the art methods (Eren Ozsoy and Can, 2013). Also, we derive the total number of constraints in the problem mathematically and show the result for a 12-gene network. The aim of the integer linear optimization model is to reconstruct the signaling network from the given PPI network satisfying RNAi data by making minimum number of changes on the given network. The corresponding integer linear program is solved by CPLEX v12.3 (64 bit). For evaluation of the method, 1000 reference PPI networks each with seven, eight, or nine proteins, and RNAi data for each of the regular proteins in the network were generated randomly. The solutions were examined to have a general overview about reconstruction of signaling networks from RNAi data by using the proposed method. We compare oILP with a state of art method on real and semi-synthetic small size networks. And lastly, we analyzed how the number of constraints grow for different network topologies and network sizes and conclude that the oILP formulation cannot scale up for large networks even with the integration of additional biological data that provide additional constraints.

2.1 Introduction

The characterization of interactions between genes in signal transduction and genetic regulatory networks is one of the main research fields in systems biology. The reconstruction of pathways in signaling networks from gene knockdown experiments is enabled by developments in RNA interference screening technology forms a basis in identifying genes related to a particular phenotype (Fire et al., 1998; Gao and Huang, 2013; Moffat and Sabatini, 2006). On the other hand, the placement of genes in the signaling networks is still a challenging task (Kaderali et al., 2009; Froehlich et al., 2007; Knapp and Kaderali, 2013).

In the approach that we developed for the reconstruction of regulatory networks from RNAi, we use RNAi hits and RNAi scores. RNAi hits determine whether a gene is a critical one or a non-critical one. Therefore, such data is Boolean, i.e., binary. In addition to this data, we use RNAi scores, which is the raw RNAi data. In our approach, RNAi scores determine the spatial ordering of the critical genes when the differences between these scores are high. Hence, the search space is reduced and more accurate biological networks can be constructed.

2.2 Related works

For the reconstruction of signaling networks from PPI data, several methods have been developed, such as the color coding algorithm (Scott et al., 2006) and Netsearch algorithm (Steffen et al., 2002). However, these methods can only search for a limited set of topologies such as linear paths.

When there is insufficient information about the genes in a signaling network, automated methods that place these genes in the network have to be developed. A survey on the methods developed for such purposes are performed by Kaderali and Radde, 2008. The developed methods use Boolean models, correlation based models and associative network approach, Bayesian networks, models based on differential equations and similar techniques. Several methods use microarray gene expression as input data and aim to generate gene regulation networks using time dependent or static data. Some methods like Bayesian networks allow the integration of biological prior information. Despite all these developed methods, the temporal and spatial placement of the genes in a signaling network is still a challenging problem.

For the construction of signaling networks using RNA interference, only a few methods are available. Markowetz et al., 2007 proposed Nested Effect Models for this problem. Such models construct the signal transduction networks by using the nested structure of observed perturbation effects. Although they are suitable models for effect-result kind of data, such as RNAi data, they require several kinds and relatively high number of readouts per knockdown. This prevents the usage of the results of RNAi experiments using one reporter gene.

Kaderali et al., 2009, developed a probabilistic method that can reconstruct network topologies using single reporter genes by generating topologies consistent with this data. However, because of the computational complexity of this method, only small networks can be solved in a reasonable time.

InfluenceFlow (Singh, 2011) combines PPI and RNAi data and finds consistent network topologies with RNAi data. It uses RNAi scores to determine the order of the genes, i.e., whether a gene is an upstream or downstream node in the signal flow. It uses linear programming to construct signaling network in tree topology. However, most signaling network topologies are more complicated than the tree topology; therefore, InfluenceFlow is not able to model such complex topologies.

Ruths et al., 2007, use PPI data and single knockout or inhibition experiments to

reconstruct signaling networks. They try to find a consistent network with a given set of single knockout or inhibition experiments. They use the given PPI network structure as a reference and make the network consistent with the gene knockdown experiments by adding missing interactions to the given PPI network. However, they do not consider false positive interactions with their approach.
Hashemikhabir et al., 2012, (SiNeC) use reference PPI networks and RNAi data together to reconstruct signaling networks. They use the given network as the initial topology and they perform a number of edge addition and edge deletion operations on this reference network while maintaining its consistency with the RNAi data. They try to keep the number of edit operations at minimum. They can handle both false positive and false negative interactions; however, they cannot deal with the genes if they are not RNAi hits and if there is no reference interaction related with this gene in the given PPI network. Therefore, their final output topology may not be a connected network in which every node is involved in the signaling process. In addition, the scalability of their proposed technique is limited to a couple of hundred nodes.

In this chapter, we propose an integer linear program based model for the reconstruction of signal transduction networks from RNAi data. We assume that the genes that comprise the signaling network are given and the related RNAi data with these genes are also provided initially. For the integer linear programming formulation, we define a binary state variable for each edge: the state variable is 1 if the edge is present in the network, otherwise it is 0. Then, each knockdown data is formulated as linear constraints after enumerating all possible paths from the source node *s* (receptor gene) to the sink node *t* (reporter/target gene). The major difference of our approach from Hashemikhabir et al.'s approach is that in addition to satisfying the RNAi constraints, we also impose a flow constraint which requires every node in the network to transduce all or part of the signal flow. The reference network can be either a PPI network or another signaling network from a different species or condition. For example, a network, which might be already constructed based on the experiments carried out in the past, may be used as a reference network. If there is novel biological information about the network, e.g., RNAi data is available, this network has to be updated with respect to this novel biological information.

Another difference of our approach from Hashemikhabir et al.'s approach is that while Hashemikhabir et al. treat RNAi data as Boolean data after thresholding, we make use of not only the same Boolean data but also the corresponding raw RNAi scores. If there is a distinctive difference between the RNAi scores of the genes, we use them in topological ordering of these genes in the network. This reduces the search space by eliminating some of the constraints in the ILP model. In addition, making use of the raw RNAi scores is beneficial in dividing the network into sub-networks while using topological features of the reference network. If some of the genes are not connected to the reference network, we cannot decide on the location of these genes. In such situations, we use the raw RNAi scores and place such a gene in the sub-network including genes with the closest RNAi scores. The details of our formulation are given in the next section of this dissertation.

2.3 Problem Formulation

Consider a given directed graph $G(V,E)$ where V represents the node (gene) set and E represents the edge (interaction) set, with a source node s (receptor gene), and a sink node t (target gene). This graph may be taken from any of the protein-protein interaction (PPI) network database with undirected PPI edges modeled as two directed twin edges. Assume that the RNAi data is available from RNAi experiment results. The aim is to reconstruct a new network from the given network satisfying RNAi data by making minimum changes on the given network. We formulate this problem as a linear optimization problem, which finds a connected network satisfying the RNAi data with a minimum change applied on the given network. A connected network is a network which includes every gene in the network topology and every gene is an element of a directed pathway from the receptor gene to the target gene.

Let x_{ij} be the binary variable representing the presence of the edge between nodes i and j which is from gene i to gene j in the given network. If the edge is present, then the value of x_{ij} is 1, otherwise it is 0. Similarly, let w_{ij} represent the edges in the final constructed network that satisfies the given RNAi data. The RNAi data consists of the information whether the signal is transferred from the source node s to the sink node t after the knockdown of a single node. We call these binary variables "the state variables". The goal is to reconstruct the given network with respect to the RNAi data by minimizing the changes that have to be applied to the given reference network. The objective function for this linear problem would be the sum of the absolute values of the differences between w_{ij} and x_{ij}, $\sum_{\forall ij} |x_{ij} - w_{ij}|$. If the difference is 1, it means that the edge is either taken out from the network or it is inserted into the network. Therefore, minimizing the sum of these differences results in a network that is obtained by applying minimum number of edit operations while satisfying the constraints obtained from the knockdowns.

The result of each knockdown can be formulated as a linear constraint after enumerating all possible paths from the source node s to the sink node t. If the signal is not observed at the sink node after knockdown of a node (gene), then any path from source s to sink t excluding the knockdown node should not be complete, i.e., the path has to be broken somewhere between the source and the sink. If it is observed, then at least one of the possible paths not including the knockdown node should be complete. If a path is not complete, then at least one of the edges on the path should not be present. Therefore, the state variable corresponding to that edge must be 0. If a path is complete, then all edges on the path must be present and the corresponding state variables take the value of 1.

To visualize the discussion above, consider a network consisting of 5 nodes, two of which are the source node s and the sink node t, as in Figure 2.1. Note that, there are no edges going into the source node *s* and no edges coming out of the sink node *t*. Also, self-edges are not allowed and there is no direct edge from source to sink. Even with these assumptions, if we disregard the sign of an edge (whether it activates or inactivates its target node) the number of possible network topologies would be close to 2^{nxn} for a network with *n* nodes. The topology shown in part (a) includes all possible paths from the source to the sink.

Let (*s*-3-*t*) be a given path on this network which is taken from a PPI network database, as in Figure 2.1. All other edges are missing in the reference network and it is known that this network consists of 5 nodes. Let the knockdown data be as given in Table 2.1. According to this data, knockdown of node 1 causes the sink node *t* to be not activated. This result can be written as mathematical constraints considering all possible paths which do not include node 1. Since no activation of the sink node is observed, none of these paths can transduce the signal, i.e., they must all be broken at one or more edges. Such a constraint can be satisfied by only setting at least one of the state variables of the edges on these paths to 0. Therefore, for a non-transducing path, the product of the state variables must be 0. We call node 1 a “critical gene” since it affects the signal transduction, i.e., the signal is not transduced to the sink node. We call the other nodes “non-critical gene(s)” since they have no effect on the signal transduction.

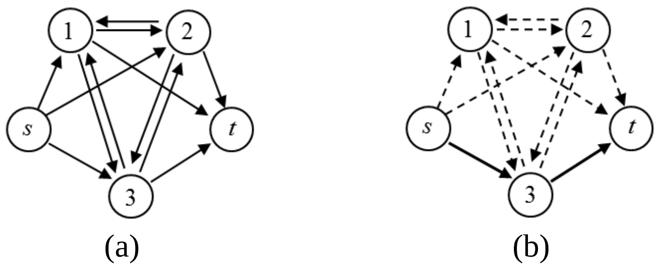


Figure 2.1 (a) A 5-node network with all possible edges, (b) a given reference network: solid lines show the edges in the reference network, dashed lines show the non-existent edges.

Table 2.1 Artificial knockdown data

Knockdown	**Effect on sink node *t***
Node 1	Not activated
Node 2	Activated
Node 3	Activated

Only source gene s is activated, while the remaining are inactive. Effect on gene t is observed.

Now, all the paths which do not include an edge connected to node 1 should be determined. For our problem with the knockdown of node 1, these non-transducing paths are (*s*-2-*t*), (*s*-3-*t*), (*s*-2-3-*t*), and (*s*-3-2-*t*). All of these paths must be broken, i.e., one of w_{ij}'s in these paths must be zero. This condition can be formulated for our problem as follows:

$$w_{s2} + w_{2t} \leq 1 \qquad (2.1)$$
$$w_{s3} + w_{3t} \leq 1 \qquad (2.2)$$
$$w_{s2} + w_{23} + w_{3t} \leq 2 \qquad (2.3)$$
$$w_{s2} + w_{23} + w_{3t} \leq 2 \qquad (2.4)$$

Since at least one of the state variables at the left hand sides of the inequalities is zero, the corresponding sum must be less than the number of terms in the inequality. There is a logical "AND" relationship between all of these constraints. They must all be satisfied at the same time; otherwise the signal would be transduced from the source node to the sink node.

Next, the second knockdown data is to be written as a constraint for our linear programming problem. Knockdown of node 2 results in activation of the sink node *t*. This observation means that at least one path that does not include node 2 must be complete, i.e., not broken, so that it is possible to transduce the signal from source to the sink. If a path transduces the signal, then all of the state variables of the edges on that path must be 1. For our problem with the knockdown of node 2, at least one path that does not include an edge connected to node 2 must be present in the network. These paths are (*s*-1-*t*), (*s*-3-*t*), (*s*-1-3-*t*), and (*s*-3-1-*t*). At least one of these paths has its w_{ij}s entirely equal to 1. This condition can be formulated for this problem as follows:

$$w_{s1} + w_{1t} \geq 2 \text{ or} \qquad (2.5)$$
$$w_{s3} + w_{3t} \geq 2 \text{ or} \qquad (2.6)$$
$$w_{s1} + w_{13} + w_{3t} \geq 3 \text{ or} \qquad (2.7)$$
$$w_{s3} + w_{31} + w_{1t} \geq 3 \qquad (2.8)$$

Note that the inequality signs (greater than or equal to) can be replaced by equalities, since the state variables are binary variables. Between these constraints, there is a logical "OR" condition, i.e., at least one of them must be satisfied. Note that constraint (2.6) cannot be satisfied because of constraint (2.2); therefore it can be omitted from the formulation.

Similarly, the knockdown of node 3 implies that at least one of the paths (*s*-1-*t*), (*s*-2-*t*), (*s*-1-2-*t*), and (*s*-2-1-*t*) must be present in the network. The formulation of this constraint is as follows:

$$w_{s1} + w_{1t} \geq 2 \text{ or} \qquad (2.9)$$
$$w_{s2} + w_{2t} \geq 2 \text{ or} \qquad (2.10)$$

$w_{s1} + w_{12} + w_{2t} \geq 3$ or (2.11)
$w_{s2} + w_{21} + w_{1t} \geq 3$ (2.12)

Here, constraint (2.10) cannot be satisfied because of constraint (2.1), therefore it can be excluded from the formulation.

After combining the objective function and all these constraints, the problem is stated as follows:

Minimize $\sum|x_{ij} - w_{ij}|$ (2.13)

Subject to
$w_{s2} + w_{2t} \leq 1$ (2.14)
$w_{s3} + w_{3t} \leq 1$ (2.15)
$w_{s2} + w_{23} + w_{3t} \leq 2$ (2.16)
$w_{s2} + w_{23} + w_{3t} \leq 2$ (2.17)

$w_{s1} + w_{1t} \geq 2$ or (2.18)
$w_{s3} + w_{3t} \geq 2$ or (2.19)
$w_{s1} + w_{13} + w_{3t} \geq 3$ or (2.20)
$w_{s3} + w_{31} + w_{1t} \geq 3$ (2.21)

$w_{s1} + w_{1t} \geq 2$ or (2.22)
$w_{s2} + w_{2t} \geq 2$ or (2.23)
$w_{s1} + w_{12} + w_{2t} \geq 3$ or (2.24)
$w_{s2} + w_{21} + w_{1t} \geq 3$ (2.25)

Now, it is necessary to convert the constraints related with “OR” (2.18)-(2.25) into linear form. We apply the strategy described in Hillier and Lieberman, 2001, to convert the OR constraints into linear form:

Either/Or constraints
Suppose that at least one of the following equalities must hold:

Either $f_1(x_1, x_2, \dots, x_n) \leq b_1$ (2.26)
or $f_2(x_1, x_2, \dots, x_n) \leq b_2$ (2.27)

Using a sufficiently large positive number *M*, an equivalent set of constraints can be formulated as

$f_1(x_1, x_2, \dots, x_n) \leq b_1 + My$ (2.28)
$f_2(x_1, x_2, \dots, x_n) \leq b_2 + M(1 - y)$ (2.29)

where *y* is a binary variable. Since *y* must be either 1 or 0, both (2.28) and (2.29) are satisfied if at least one of (2.26) and (2.27) is satisfied. A more general case when *K* out of N constraints must hold is explained below (Hillier and Lieberman, 2001).

K out of N Constraints must hold:
If there are several constraints only some of which must hold, formulation of either/or constraint explained above can be expanded to account for such requirement. Assume that there are N constraints and K of them must hold where $K < N$. The formulation is stated as

$$\begin{aligned}
&f_1(x_1, x_2, \dots, x_n) \leq b_1 + My_1\\
&f_2(x_1, x_2, \dots, x_n) \leq b_2 + My_2\\
&\qquad\qquad\qquad\cdot\\
&\qquad\qquad\qquad\cdot\\
&\qquad\qquad\qquad\cdot\\
&f_N(x_1, x_2, \dots, x_n) \leq b_N + My_N\\
&y_1 + y_2 + \cdots + y_1 = N - K\\
&y_i = 0 \text{ or } 1.
\end{aligned}$$

The equality in $y_1 + y_2 + \cdots + y_1 = N - K$ can be changed by an inequality ≤ if at least K constraints are required to be satisfied.

If y_i = 0, then the original constraint is obtained. However, if y_i = 1, because of the large positive number M, even if the original constraint is not satisfied, the new constraint is satisfied. Since K out of N constraints must hold, the summation of y_i's should be equal to N-K.

Now, we can reformulate the constraints (2.18)–(2.25) using the method described above to transform "OR" conditions into "AND" conditions. The constraints (2.18)-(2.21) and (2.22)-(2.25) then take the form

$$w_{s1} + w_{1t} + My_1 \geq 2 \tag{2.30}$$
$$w_{s3} + w_{3t} + My_2 \geq 2 \tag{2.31}$$
$$w_{s1} + w_{13} + w_{3t} + My_3 \geq 3 \tag{2.32}$$
$$w_{s3} + w_{31} + w_{1t} + My_4 \geq 3 \tag{2.33}$$
$$y_1 + y_2 + y_3 + y_4 \leq 3 \tag{2.34}$$
$$w_{s1} + w_{1t} + My_5 \geq 2 \tag{2.35}$$
$$w_{s2} + w_{2t} + My_6 \geq 2 \tag{2.36}$$
$$w_{s1} + w_{12} + w_{2t} + My_7 \geq 3 \tag{2.37}$$
$$w_{s2} + w_{21} + w_{1t} + My_8 \geq 3 \tag{2.38}$$
$$y_1 + y_2 + y_3 + y_4 \leq 3 \tag{2.39}$$

Since it is required that at least one of the constraints (2.30)-(2.33) and (2.35)-(2.38) must hold, (2.34) and (2.39) are written as inequalities respectively and N - K = 4 – 1 = 3. Eliminating the constraints that cannot be satisfied due to the constraints obtained from knockdown of node 1, the problem can be finally stated as

Minimize $\sum|x_{ij} - w_{ij}|$ (2.40)

Subject to

$w_{s2} + w_{2t} \leq 1$ (2.41)

$w_{s3} + w_{3t} \leq 1$ (2.42)

$w_{s2} + w_{23} + w_{3t} \leq 2$ (2.43)

$w_{s2} + w_{23} + w_{3t} \leq 2$ (2.44)

$w_{s1} + w_{1t} + My_1 \geq 2$ (2.45)

$w_{s1} + w_{13} + w_{3t} + My_2 \geq 3$ (2.46)

$w_{s3} + w_{31} + w_{1t} + My_3 \geq 3$ (2.47)

$y_1 + y_2 + y_3 \leq 3$ (2.48)

$w_{s1} + w_{1t} + My_4 \geq 2$ (2.49)

$w_{s1} + w_{12} + w_{2t} + My_5 \geq 3$ (2.50)

$w_{s2} + w_{21} + w_{1t} + My_6 \geq 3$ (2.51)

$y_4 + y_5 + y_6 \leq 3$ (2.52)

The optimum solution to this linear program gives 3 as the objective function value and the state variables are found as w_{3t}=1, w_{s1}=1, w_{1t}=1, and the remaining are 0. Therefore, totally 3 changes are applied to satisfy the given constraints: edge *s*-3 is removed and edges *s*-1 and 1-*t* are added to the network. The final network structure is given in Figure 2.2.

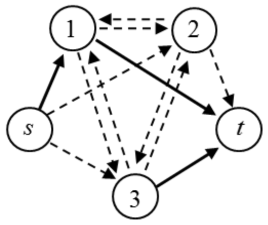


Figure 2.2 The resulting network satisfying the RNAi data as given in (Table 2.1) and therefore the constraints (2.41)-(2.52).

The solution makes sense considering the given constraints. In order to prevent signal transduction with the knockdown of node 1, either edge *s*-3 or edge 3-*t* must be removed from the network. Also, in order for the signal to be transduced after the knockdown of node 2 or 3, the path *s*-1-*t* must be present in the network. There may be more than one optimum solution for this problem. Removing edge 3-*t* and keeping edge *s*-3 results in another optimum solution. We report one of these possible solutions as output by the tool we use, CPLEX v12.3 (64 bit).

In this formulation, we assume that there is at least one non-critical gene, i.e., there is at least one OR-condition. This formulation works even if all the nodes are non-critical; however, it does not work if they are all critical genes. When there is at least one non-critical gene, our formulation guarantees that there is a path from the source to the sink due to the OR-conditions we impose. However, when there is no non-critical gene, we state in our formulation that only some of the pathways must be incomplete so that the signal cannot be transduced. This formulation results in an incomplete network for such specific problem, therefore we need to consider a specific solution. The solution to this problem is simple; if all nodes are critical, it means that all of them are connected end to end. Thus, we include the corresponding constraints which state that at least one of them should be satisfied (as in OR-conditions described above). Assume that in the previous 5-node example, all nodes are critical. Then, the solution network must be one of the following: *s*-1-2-3-*t*, *s*-1-3-2-*t*, *s*-2-1-3-*t*, *s*-2-3-1-*t*, *s*-3-1-2-*t*, *s*-3-2-1-*t*. The constraints for this case are

$$w_{s1} + w_{12} + w_{23} + w_{3t} + My_1 \geq 4 \tag{2.53}$$
$$w_{s1} + w_{13} + w_{32} + w_{2t} + My_2 \geq 4 \tag{2.54}$$
$$w_{s2} + w_{21} + w_{13} + w_{3t} + My_3 \geq 4 \tag{2.55}$$
$$w_{s2} + w_{23} + w_{31} + w_{1t} + My_4 \geq 4 \tag{2.56}$$
$$w_{s3} + w_{31} + w_{12} + w_{2t} + My_5 \geq 4 \tag{2.57}$$
$$w_{s3} + w_{32} + w_{21} + w_{1t} + My_6 \geq 4 \tag{2.58}$$
$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 5 \tag{2.59}$$

The formulation described above as a whole may result in a network in which some of the nodes are not connected to the network or they can act as a sink node (no edge is going out of them) or as a source node (no edge is going into them). Such situation can occur for a non-critical gene. For non-critical genes, we consider the paths that do not include the non-critical gene, which gives no information about the non-critical gene. Therefore, this node may not be connected to the network after solving the problem, because the constraints are satisfied. Since this node is a non-critical gene, i.e., the signal is transduced after it is knocked down, it does not make any difference whether it is connected to the network or not. In order for this node to be included in the solution network, there must be at least one path from source node to sink node containing this particular node. As an example, consider the 5-node network given in Figure 2.1. For this network, node 2 is a non-critical gene, so in addition to the constraints (2.30)-(2.34), new constraints that include the path passing through node-2 must be written and connected by ORs. These additional constraints are given as

$$w_{s2} + w_{2t} + My_1 \geq 2 \tag{2.60}$$
$$w_{s1} + w_{12} + w_{2t} + My_2 \geq 3 \tag{2.61}$$
$$w_{s2} + w_{23} + w_{3t} + My_3 \geq 3 \tag{2.62}$$
$$w_{s2} + w_{21} + w_{1t} + My_4 \geq 3 \tag{2.63}$$
$$w_{s3} + w_{32} + w_{2t} + My_5 \geq 3 \tag{2.64}$$
$$y_1 + y_2 + y_3 + y_4 + y_5 \leq 4 \tag{2.65}$$

Adding such constraints for all non-critical genes will yield these nodes to be included in the network. In other words, we force our method to create a network for which every node is a member of a pathway that is directed from source node to the sink node. With this approach, we can construct networks with up to 10 nodes in a reasonable time. We name this method as oILP (original ILP formulation).

2.4 Automatic Generations of Constraints

In order to solve the network problem described in the previous section by using the optimization studio IBM CPLEX, it is necessary to generate the constraints automatically as the number of constraints becomes very large even for pathways with small number of genes. An LP-format file which can be read by CPLEX is created by a code written in C. The code generates the objective function with the input data, i.e. a given reference PPI network. The objective function consists of all the edge values w_{ij} that are to be found and the values x_{ij} which are the edge values for the given initial network. Then, for RNA interference data, the constraints are generated. Firstly, the constraints for knockdown of the genes which do not activate the receptor gene *t* are generated. The constraints include all possible paths from the reporter gene *s* to the receptor gene *t* which do not include the knockdown gene and there exists an "AND" relation between them as explained above. These paths have a minimum of 2 edges since a direct edge from source to sink and paths with loop are not allowed and a maximum of *n*-1 edges for a network consisting of *n* genes. For all configuration of paths, i.e. paths with different number of edges (e.g. paths consisting of 2 edges, 3 edges… *n*-1 edges), the constraints are written in the LP-file. Next, the constraints for the knockdown of the genes which activate the receptor gene *t* are generated. The constraints include all possible paths from the reporter gene *s* to the receptor gene *t* which do not include the knockdown gene and there exists an "OR" relation between them as explained above. They also have a minimum of 2 edges and a maximum of *n*-1 edges. Since they are related with "OR" condition, y_i and M values are added to the constraints and the number of y_i values is counted to write the additional constraints as in (2.18) and (2.22). Some of these paths are already dealt with when considering the genes which are not activating the receptor gene, therefore the corresponding constraint cannot be satisfied and they are excluded. In fact, this is the reason why the constraints for the genes which do not activate the receptor gene are generated first. Lastly, the variables w_{ij} and y_i are defined as binary variables in the LP-file.

2.5 Data Sets

To evaluate the method described above, we use the following data sets in our experiments.

Real data sets:
We use type I IFN stimulated Janus Kinases and Signal Transducers and Activators of Transcription (JAK/STAT) network (Platanias, 2005) as real data set. JAK/STAT network is a small network with nine components. The signaling protein in this 9 node network is IFNA2 (Interferon alpha-2) and the reporter protein is Luciferase.

The genes IFNAR1, IFNAR2, JAK1, TYK2, STAT2 and IRF9 are reported to be the critical genes and STAT1 is reported to be a noncritical gene (Kaderali et al., 2009).

Semisynthetic data set:
For generating semisynthetic data set, we use KEGG database (Kanehisa and Goto, 2000). Fruit fly MAPK signaling network is selected from KEGG database. MAPK fruit fly is a sub pathway with the tailless (tll) and huckebein (hkb) genes. It has 10 genes and 10 edges, which is the number of interaction between genes. Using this network, we have generated 25 semisynthetic data sets for this actual network. This is done by inserting edges to the actual networks at a certain rate and removing edges at the same or at a different rate; i.e. inserting or removing different number of edges from each of them. Therefore, both dense and sparse networks have been generated. These rates are; 10% insertion-10% deletion for the 1st network, 10% insertion-20% deletion for the 2nd network, ..., 10% insertion-50% deletion for the 5th network, 20% insertion-10% deletion for the 6th network, ..., 20% insertion-50% deletion for the 10th network, ..., 30% insertion-40% deletion for the 14th network, ..., 50% insertion-50% deletion for the 25th network. First, edges are deleted randomly at a rate of p%; meaning p x |E| edges are randomly deleted from the actual network. Then, addition of edges is performed. For addition, we first generate the set of all edges as if the actual sinaling network is a fully connected network. After that, we take the difference of two networks, i.e. the fully connected network of the actual signaling network and the actual signaling network, and name it “the set of extraction”. Then, we add edges at a rate of p%, meaning p x |E| edges are randomly added to the network from the set of extraction. At the end, we have 25 unique reference networks with different number of edges generated by this method and named as ADDRIP (“addition/deletion rates in percentages”).

Synthetic data set:
We have randomly generated 1,000 reference networks each with seven, eight, or nine genes, including the receptor and target genes. Each edge in a network is an obtained by Bernoulli trial with probability 0.5. In addition, RNAi constraints have been generated for each of the genes with *p*(critical gene=1)=0.5.

RNAi score generation: The real data sets we use in this study have RNAi data from RNAi knockdown experiments. However, synthetic and semi-synthetic data sets do not have RNAi data. Therefore, we have generated synthetic RNAi scores for our synthetic and semi-synthetic data sets. In order to generate RNAi scores, we first generate every possible path from the receptor gene to the target gene by a depth first search traversal of the network. After finding all possible paths between the receptor and the target gene, we analyze the genes in these paths. If a gene is included in most of the paths, than this gene has a higher influence in signal transduction. We assign an RNAi score to each gene based on the ratio of paths that the gene is observed. We simulate noise by a Gaussian error function.

2.6 Evaluation of the oILP approach

We analyzed the performance of the original ILP formulation on 1000 synthetic small networks of sizes 7 to 9 components. Since the RNAi constraints for each of the regular genes in the network is generated with *p*(critical gene=1)=0.5, most of the initial PPI networks are dense networks. Then, for each PPI network we generate RNAi data. The knockdown results are simulated by randomly assigning a value of 0 or 1 to the knockdown data for each gene between the receptor and the target genes. These values represent the observable results at the target gene after each knockdown. If the observable value is 0 at the target gene, the knockdown gene is a critical gene. If it is 1, then the corresponding gene is a noncritical gene. From this RNAi data, the constraints for our ILP are constructed. If the knockdown result is 0, "AND" constraints are created; if it is 1, "OR" constraints are written down.

Given a reference PPI network and a set of RNAi constraints, the corresponding integer linear program is solved by CPLEX_12.3 (64 bit). For each network, we inspect six different criteria, namely, (1) number of critical genes vs. average solution time, (2) number of critical genes vs. number of edit operations applied on the reference network, (3) number of edges in the reference network vs. average solution time, (4) number of edges in the reference network vs. number of edit operations applied on the reference network, (5) number of constraints vs. number of critical genes, (6) difference in number of "AND" and "OR" constraints vs. number of critical genes.

The number of critical genes for an *n*-gene network can be at most *n*-2, i.e. all the genes between the receptor gene and the target gene are critical genes. If all of the genes are critical genes, we do not solve the problem, since the formulation described above is valid when there is at least one complete path from receptor gene to the target gene. The presence of a complete path in this formulation is guaranteed by having a noncritical gene, because at least one of the paths that do not include the noncritical gene must be complete.

First, 7-gene networks are considered. The number of 7-gene networks created randomly are shown in Figure 2.3 with respect to the number of critical genes. The figure shows a normal distribution since the networks are created randomly with a 0.5 probability of a gene being a critical gene. In some of the networks, all genes (i.e. 5 genes) are critical genes. In that case, the problem is not solved. Similar graphs for 8 and 9 gene networks are shown in Figure 2.10 and Figure 2.17. Figure 2.4 shows the boxplot of solution times of oILP for different number of critical genes for 7-gene networks. The black circles represent the outliers and they make the standard deviation very high. This is because the solution time depends not only on the number of critical genes but also on many other variables, such as number of constraints and number of edges in initial network. The solution time shows a normal distribution over the number of critical genes. This can be explained by the number of "AND" and "OR" constraints. Although the number of constraints decreases with the increasing number of critical genes, as shown in Figure 2.8 the difference between them shows a similar but reverse behavior with the solution time, as shown in Figure 2.9. If the difference between the number of “AND” and “OR” constraints are small, the average solution time is high; if the difference is high, then the average solution time is low. From this observation, we can conclude that as the “and” and “OR” constraints mix more, the solution time gets higher since finding the edge values minimizing the objective function and at the same time satisfying both the “AND” and “OR” constraints are getting harder.

In Figure 2.5, the number of changes made on the initial network with respect to the number of critical genes are shown. As the number of critical genes increases, the number of changes made on the initial network also increases. This is due to the increase in the number of “AND” constraints with the number of critical genes. Satisfying all the “AND” constraints is more difficult than satisfying only one of the “OR” constraints, because while satisfying only one “OR” constraint is enough, all the “AND” constraints must be satisfied at the same time. Therefore, more changes are made on the initial network as the number of critical genes increases.

As mentioned before, the solution time may also depend on the number of edges in the initial network. Such a dependence is shown in Figure 2.6. Here, the number of edges in initial networks span from 7 to 23. This is because these networks are created randomly and each edge has a 0.5 probability of being present in the initial network. Figure 2.6 shows that the solution time slightly increases with the number of edges in the initial network. This is as expected because as the number of edges in initial network increases, more edges should be removed from the network to satisfy the constraints and therefore the time required to find such edges increases. Consequently, as shown in Figure 2.7, the number of changes made in the initial network also increases with the increasing number of edges in the initial network. In these two figures, while the solution time deviates too much, the standard deviation of the number of the changes in initial network is smaller and proportinal to the number of edges in the initial network.

The above discussion is also valid for 8 gene networks, Figure 2.10 to Figure 2.16; and for 9 gene networks, Figure 2.17 to Figure 2.23. The created networks have a normal distribution. As the number of genes increase, the total number of constraints increase exponentially, which in turn results in an exponential increase in the solution time. Similarly, the number of edges in initial network and the number of changes imposed on the initial network increases with number of genes in the network.

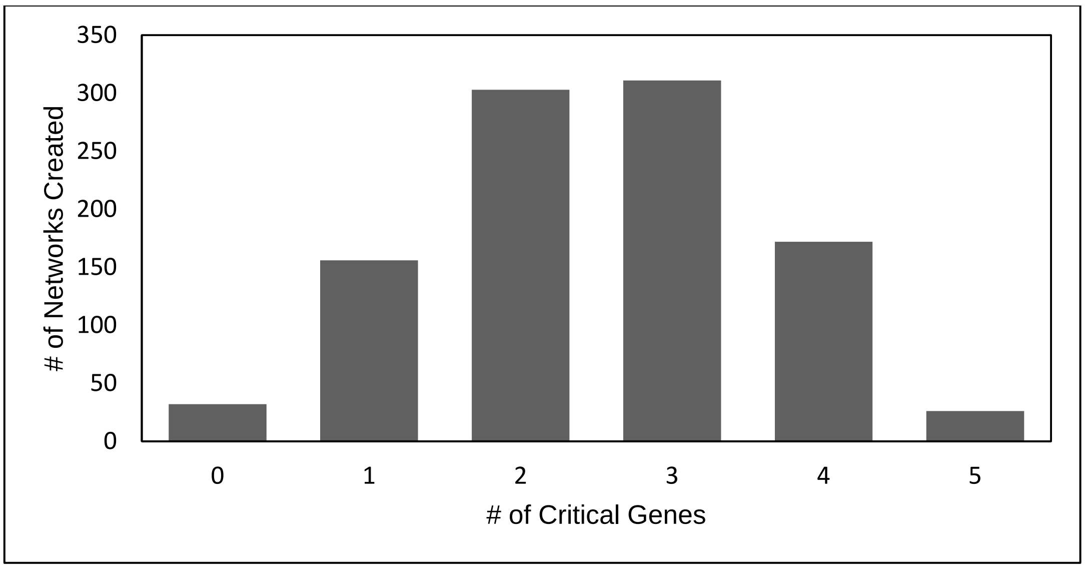


Figure 2.3 Number of 7-gene networks that are created randomly for different number of critical genes.

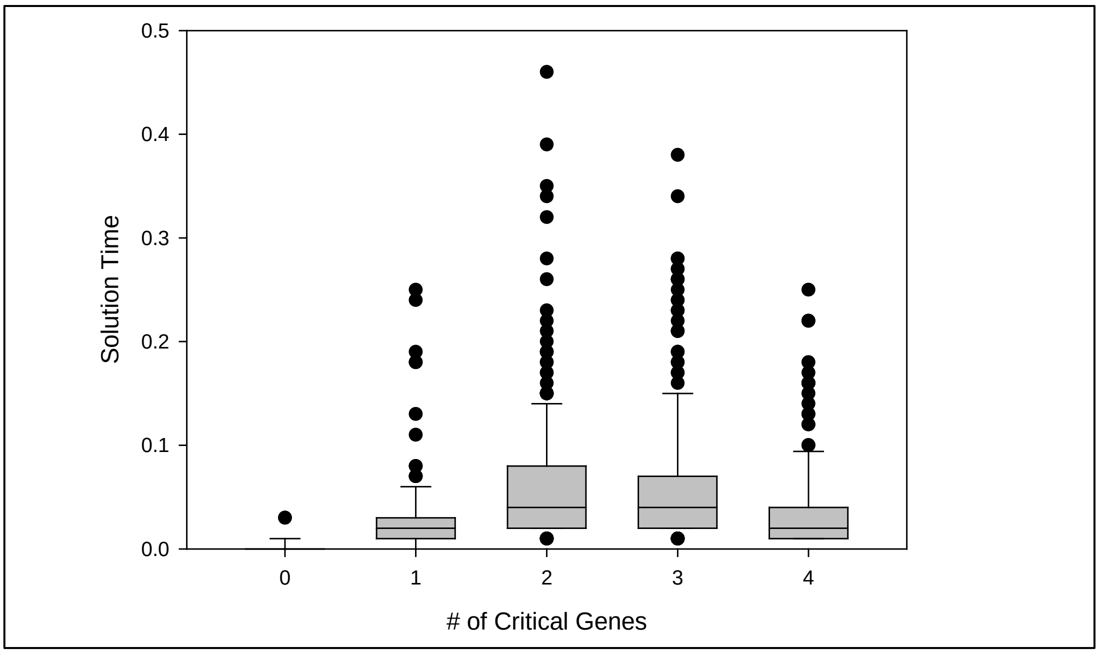


Figure 2.4 Solution times for the 7-gene networks with different number of critical genes.

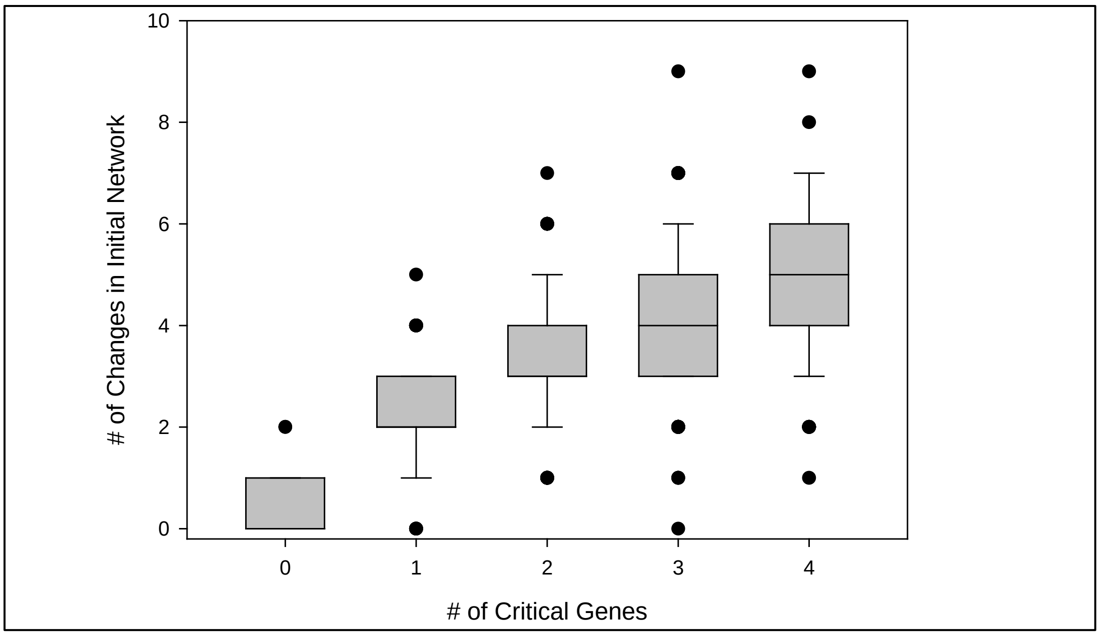


Figure 2.5 Number of changes applied on the reference 7-gene networks with different number of critical genes.

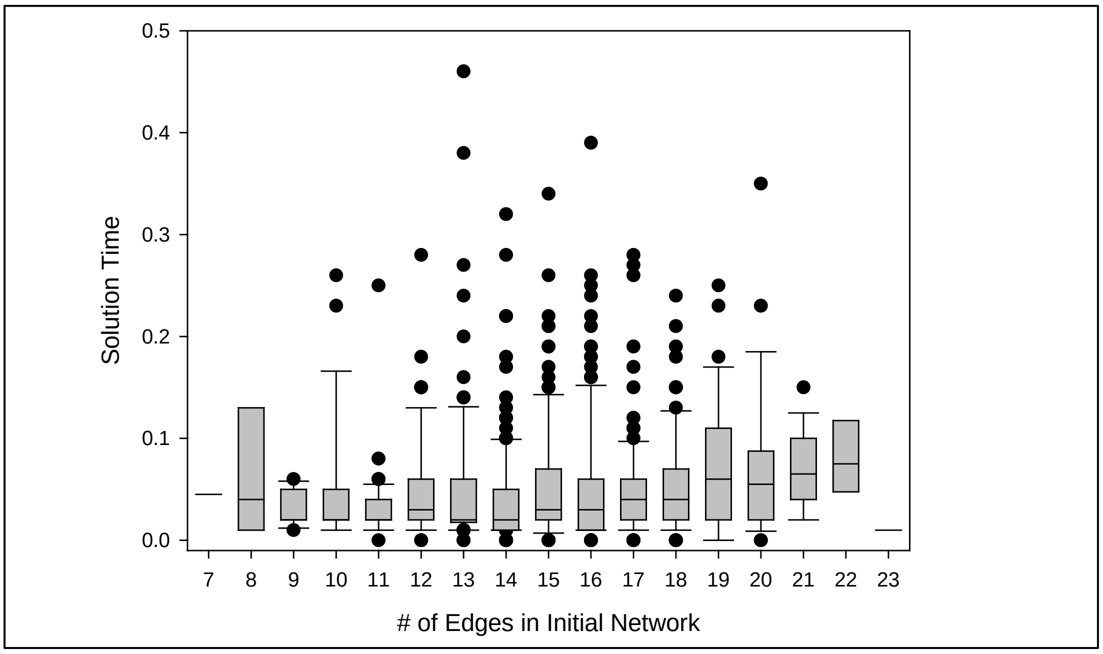


Figure 2.6 Solution times for the 7-gene networks with different number of edges in the reference networks.

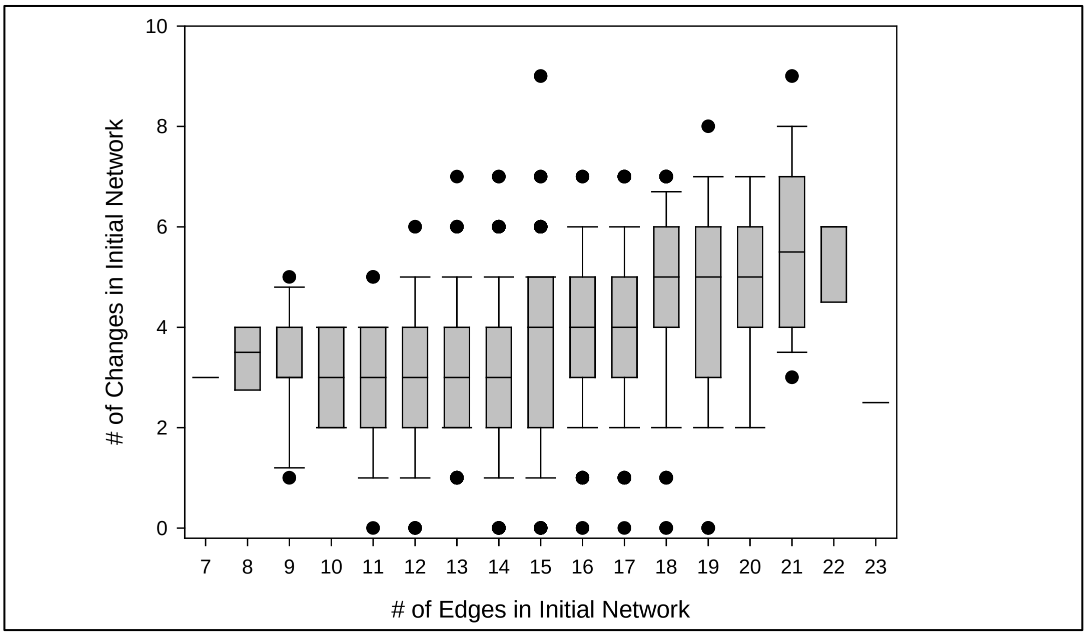


Figure 2.7 Number of changes applied on the reference 7-gene networks with different number of edges in the reference networks.

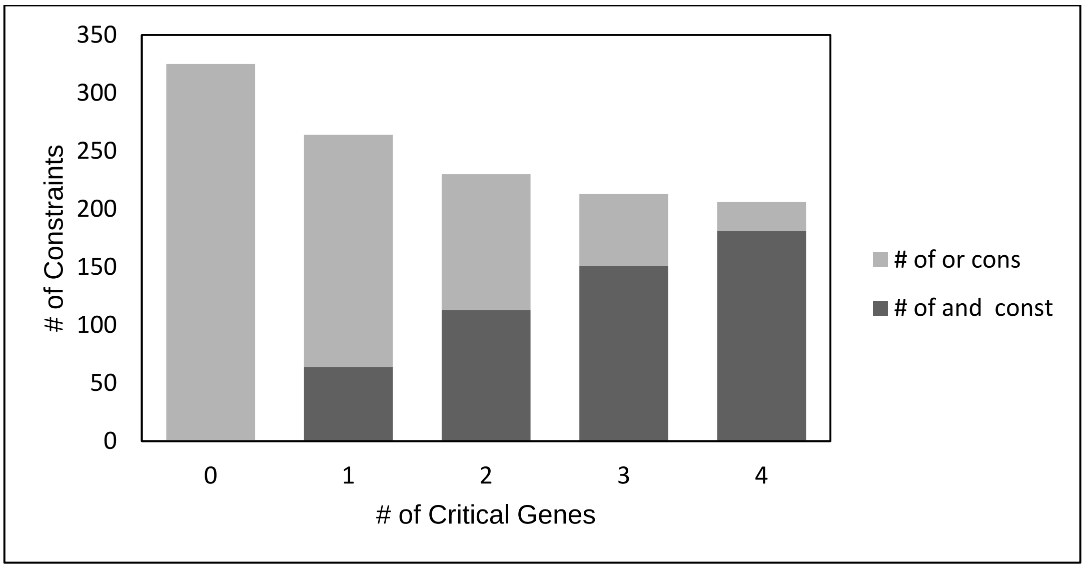


Figure 2.8 Number of “AND” and “OR” constraints for 7-gene networks with different number of critical genes.

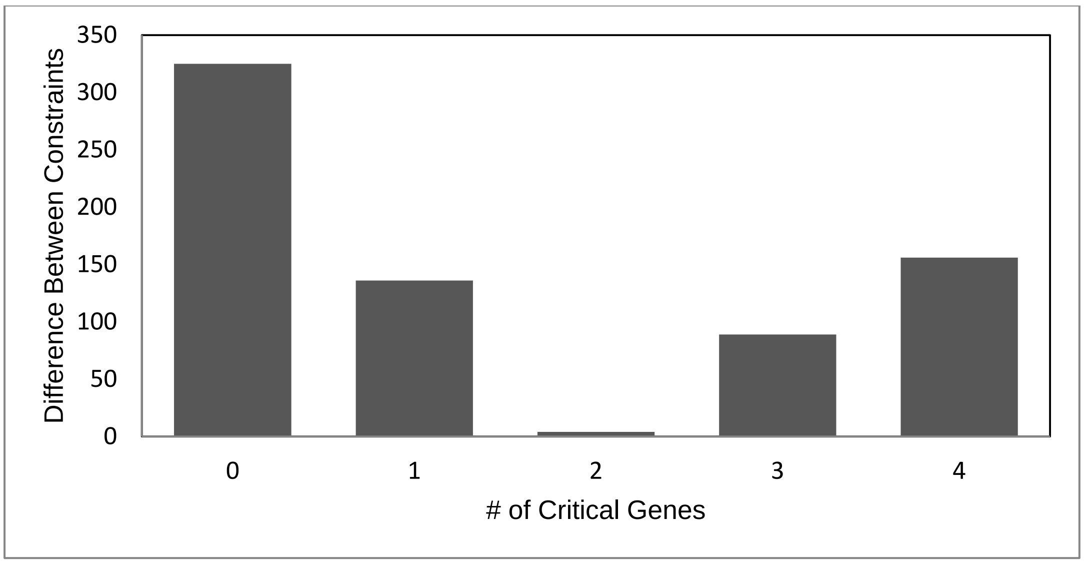


Figure 2.9 Difference between “AND” and “OR” constraints for 7-gene networks with different number of critical genes.

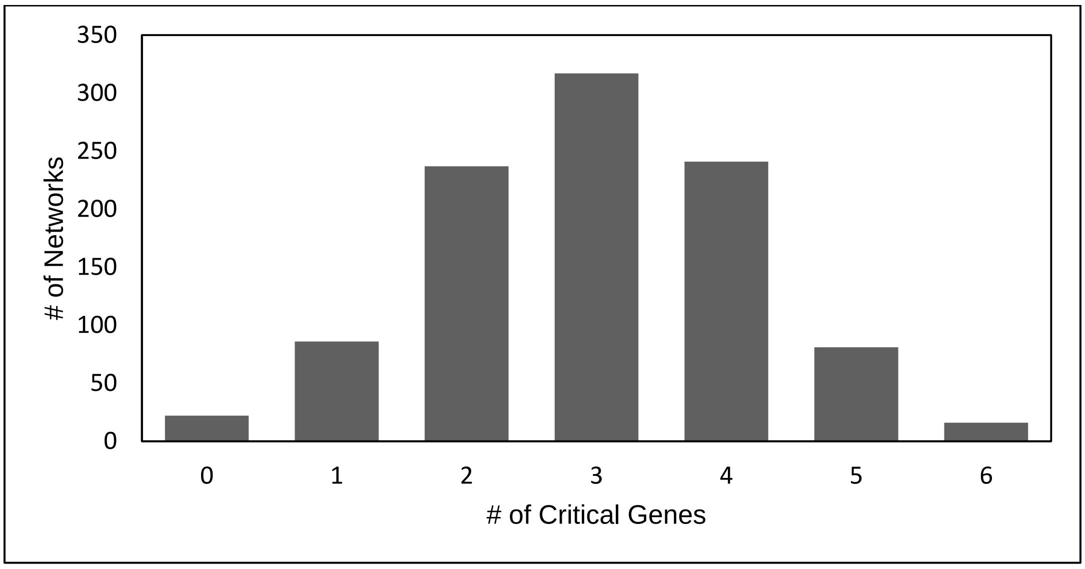


Figure 2.10 Number of 8-gene networks that are created randomly for different number of critical genes

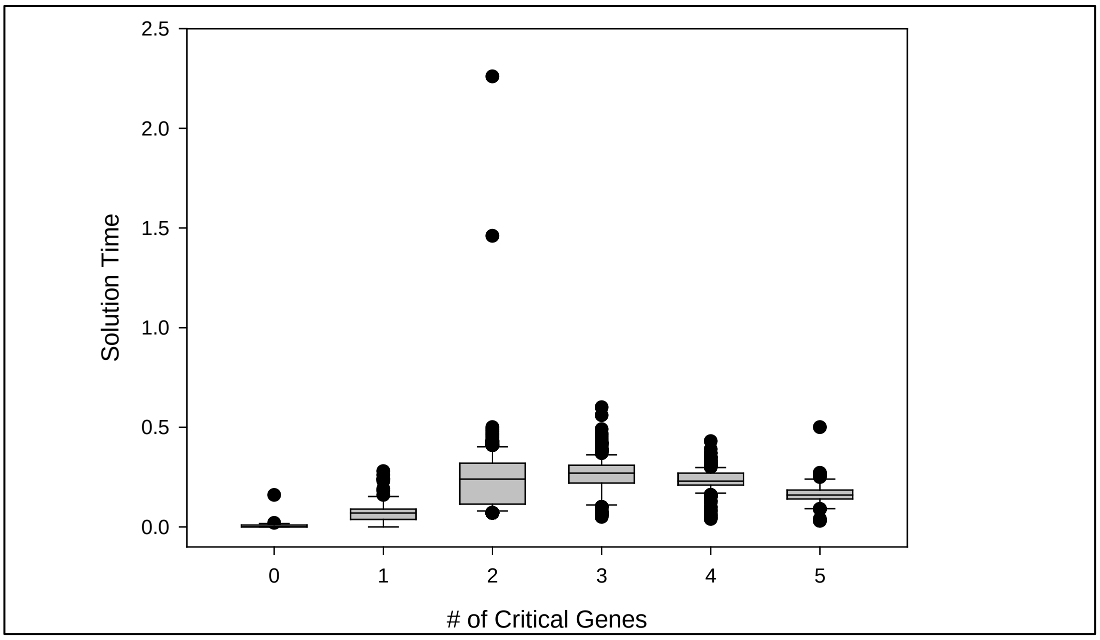


Figure 2.11 Solution times for the 8-gene networks with different number of critical genes.

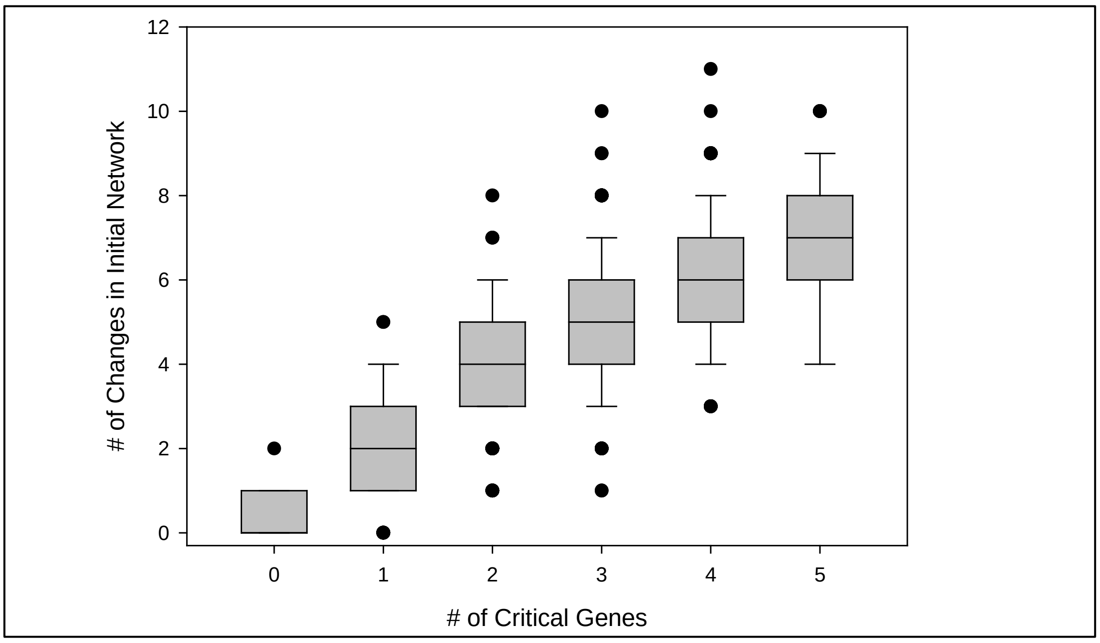


Figure 2.12 Number of changes applied on the reference 8-gene networks with different number of critical genes.

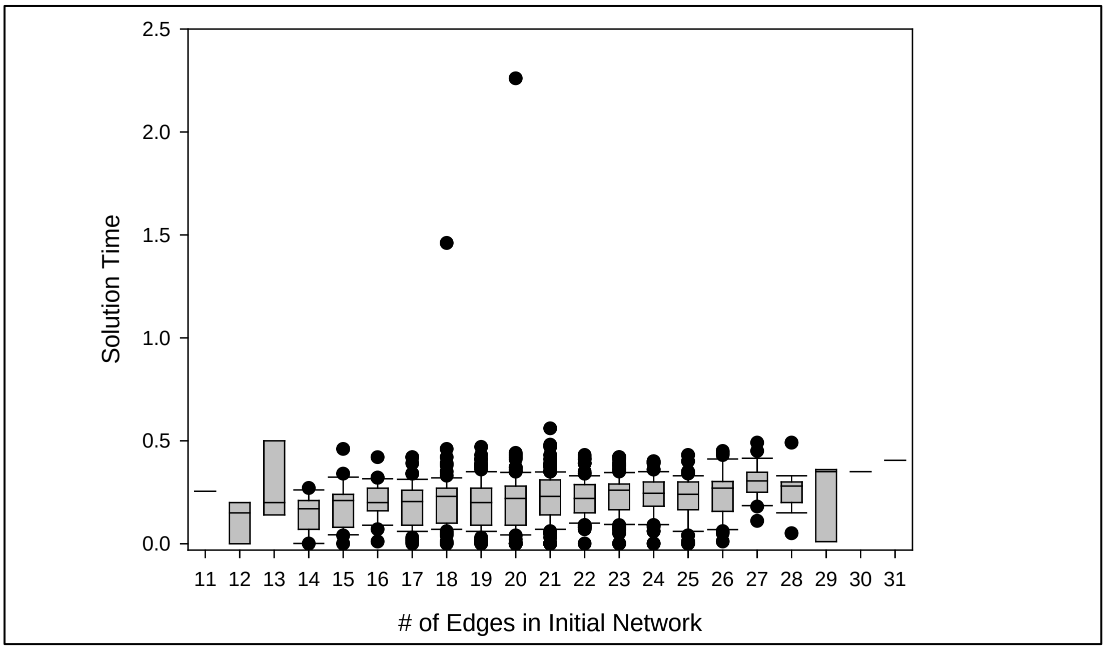


Figure 2.13 Solution times for the 8-gene networks with different number of edges in the reference networks.

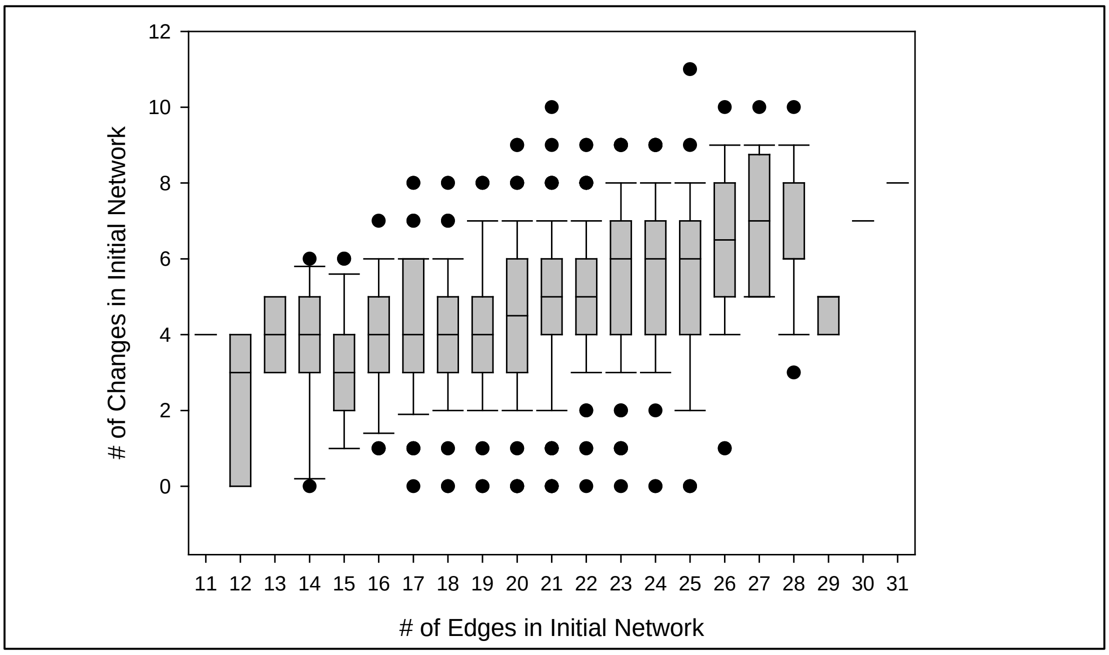


Figure 2.14 Number of changes applied on the reference 8-gene networks with different number of edges in the reference networks.

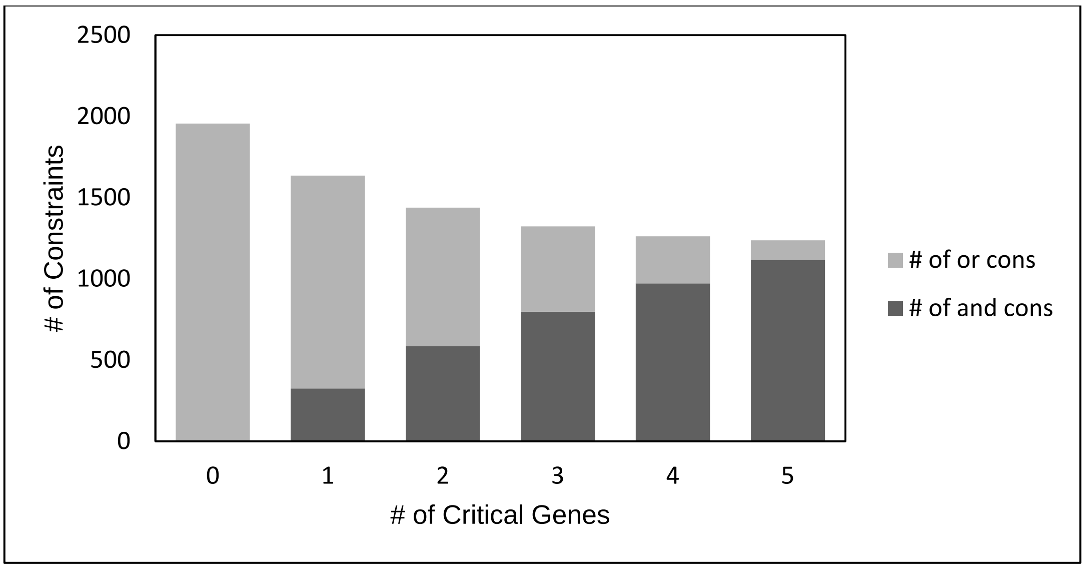


Figure 2.15 Number of “AND” and “OR” constraints for 8-gene networks with different number of critical genes.

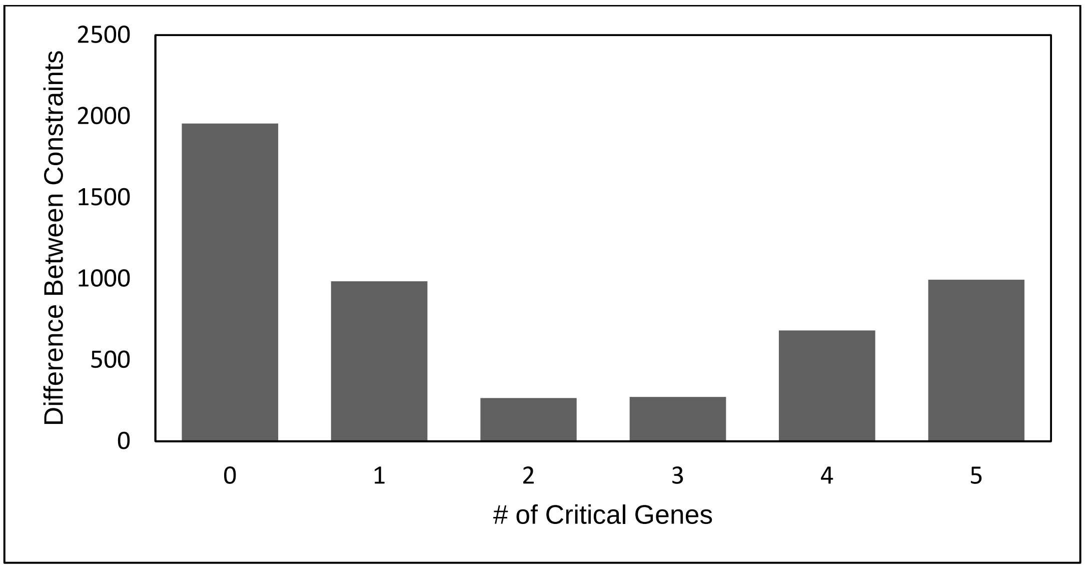


Figure 2.16 Difference between “AND” and “OR” constraints for 8-gene networks with different number of critical genes.

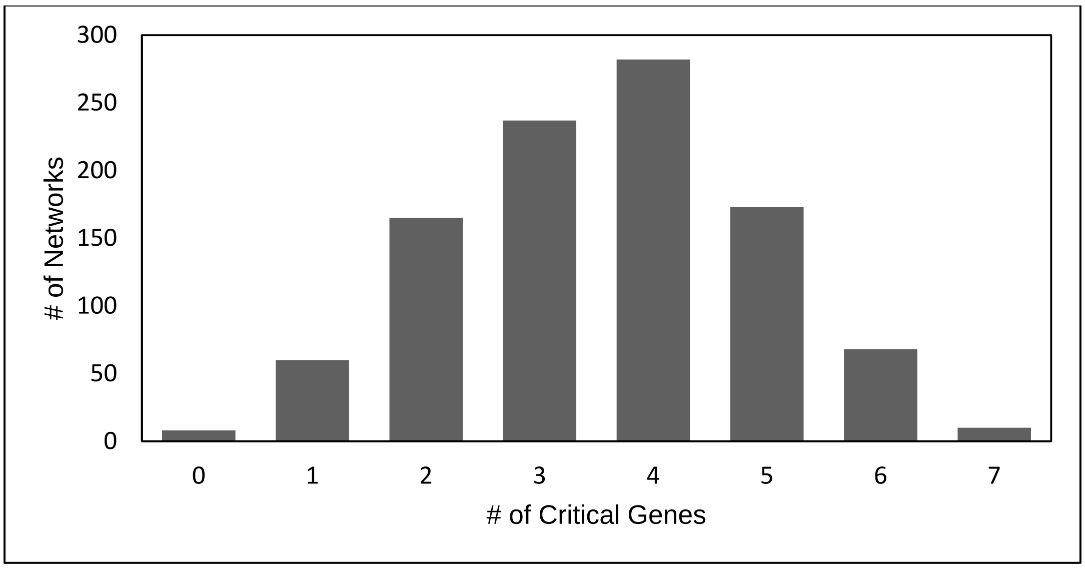


Figure 2.17 Number of 9-gene networks that are created randomly for different number of critical genes.

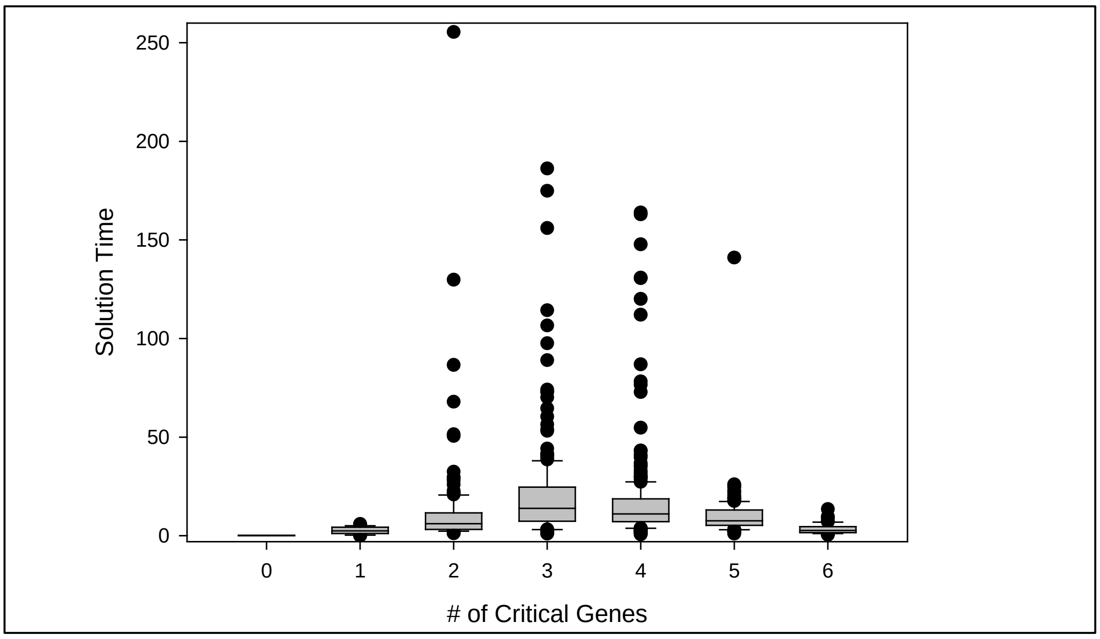


Figure 2.18 Solution times for the 9-gene networks with different number of critical genes.

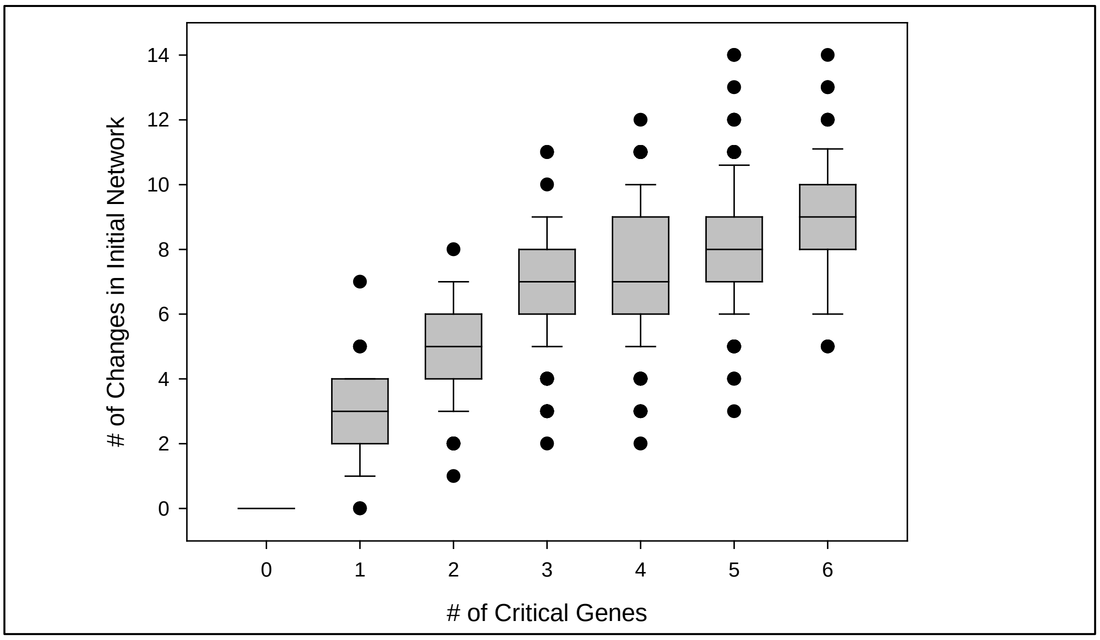


Figure 2.19 Number of changes applied on the reference 9-gene networks with different number of critical genes.

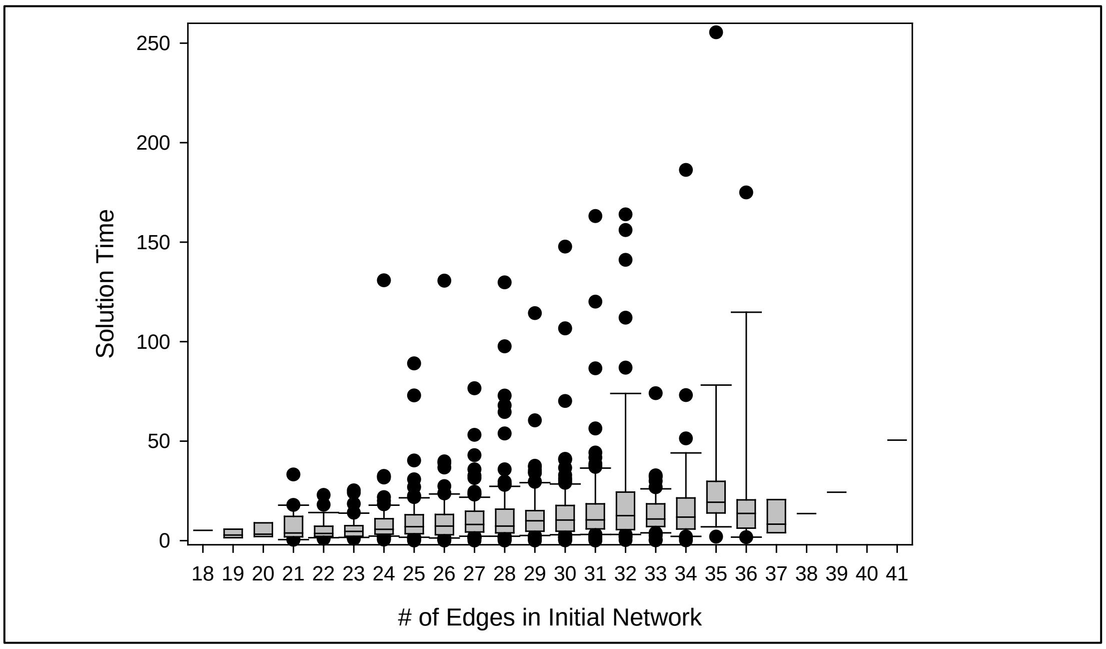


Figure 2.20 Solution times for the 9-gene networks with different number of edges in the reference networks.

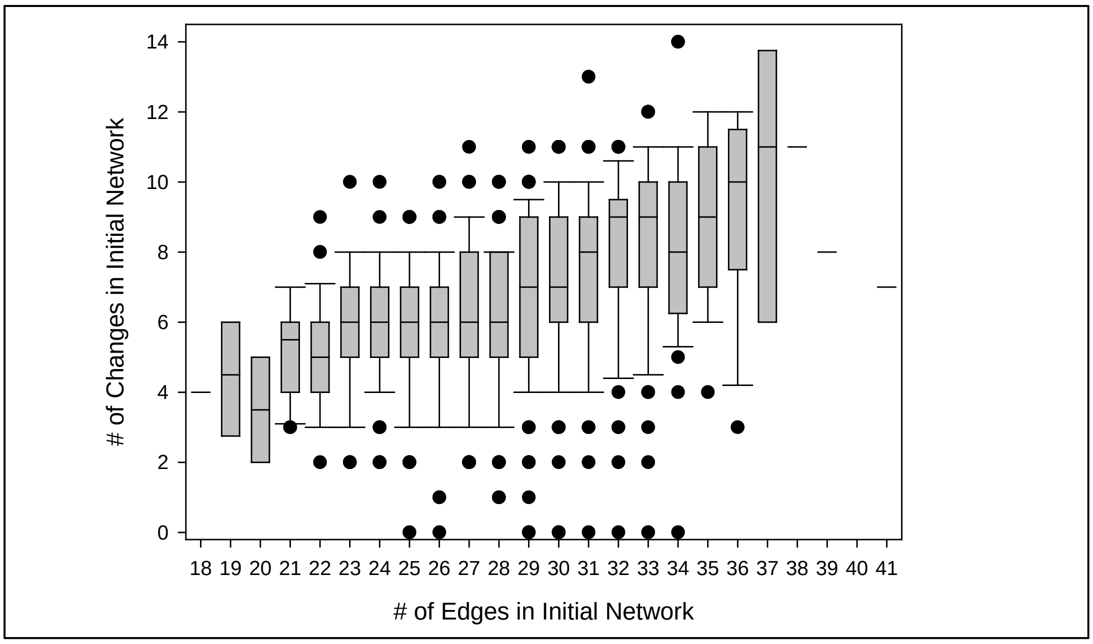


Figure 2.21 Number of changes applied on the reference 9-gene networks with different number of edges in the reference networks.

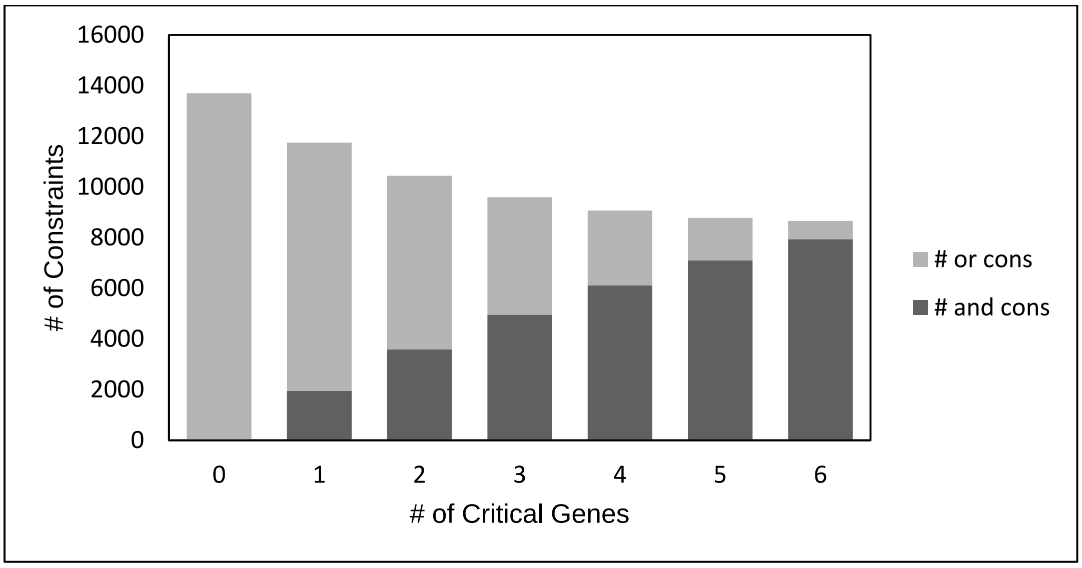


Figure 2.22 Number of “AND” and “OR” constraints for 9-gene networks with different number of critical genes.

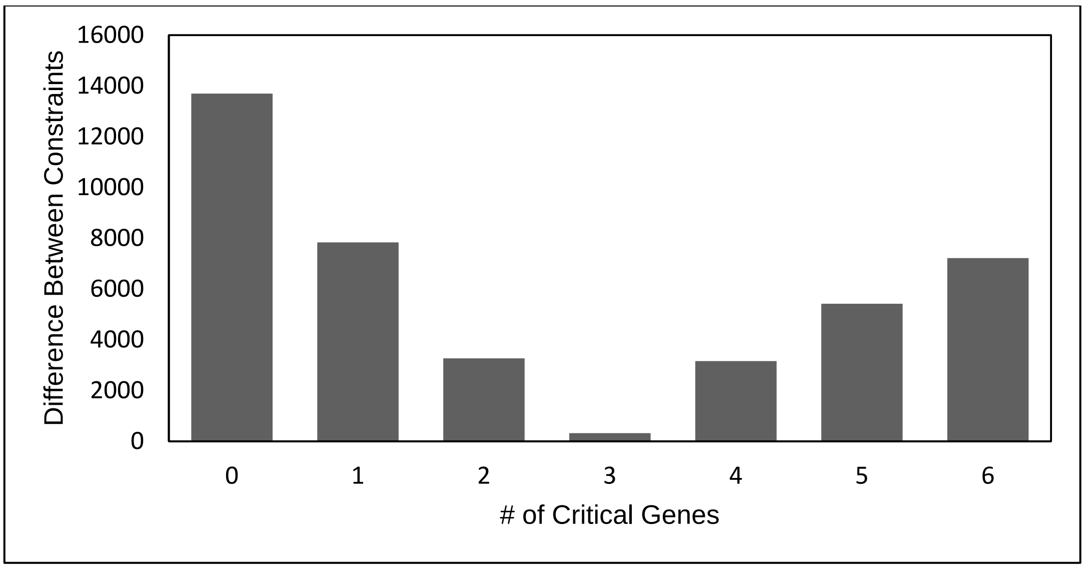


Figure 2.23 Difference in number of constraints between “AND” and “OR” constraints for 9-gene networks with different number of critical genes.

2.7 Comparison of oILP with state of the art

We evaluated oILP first by applying it on the semisynthetic data (MAPK fruit fly) and than on the real data (JAK/STAT) which are described in Section 2.4. We compare the precision and recall values with the state of the art methods. these two accuracy measures are described below.

Let G and G_c be the actual and constructed networks.

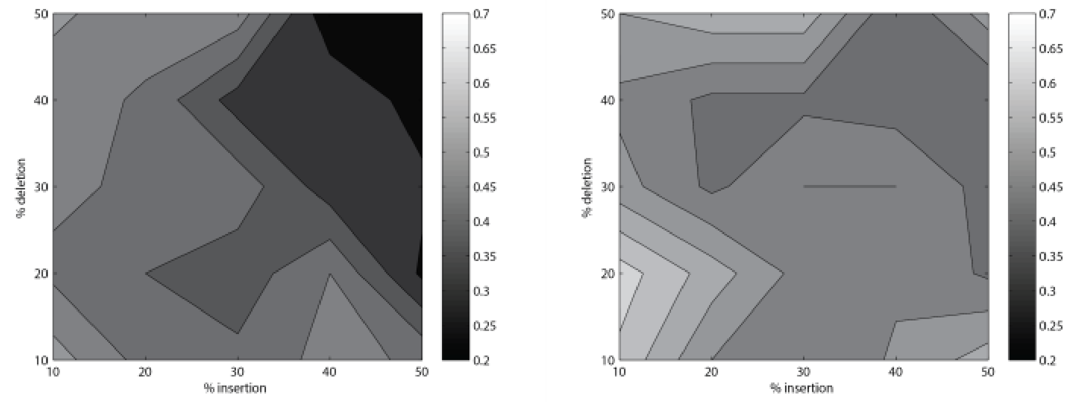
- *Precision* is defined as the ratio of the number of interactions common to G and G_c to the interactions in G_c.
- *Recall* is defined as the ratio of the number of interactions common to G and G_c to the interactions in G.

2.7.1 Comparison of the results on semisynthetic MAPK network

We first compare the results of oILP with SiNeC (Hashemikhabir et al., 2012). Although SiNeC and oILP have the same basic fundamentals dealing with the network inference problem, oILP has two essential differences; the major difference is that oILP results in a complete network structure, and the other one is that oILP use raw RNAi scores along with the information obtained from RNAi data which figures out whether a gene is a critical gene or not. Therefore, to point out that our method provides more accurate network topologies, we compare the results of oILP and SiNeC on small size networks of up to 10 nodes. This comparison is performed on the 25 networks that are mutated from the MAPK signaling network of Drosophila melanogaster (fruit fly) taken from the KEGG database. As we mentioned above, we use RNAi scores related with the RNAi hits to find the farthest gene from the sink node as described in Singh, 2011, and assume all of the reference networks are undirected networks. On the other hand, while SiNeC uses RNAi hits (meaning that the critical genes are known), it does not use the corresponding RNAi scores. Also, SiNeC uses the same reference networks as oILP does. According to the RNAi scores, the gene TOR is determined to be the farthest gene to the target gene. The results from SiNeC and oILP are compared in Figure 2.24 and Figure 2.25 by means of recall and precision, respectively. The results used in obtaining the contour plots are also given in Table 3.2 in Chapter 3, along with the results for the comparison of SiNeC and our methods for large scale networks, for clarity. On these contour plots, x-axis represents the percent insertion and y-axis represents the percent deletion. The observed variable (third dimension) is indicated as a grayscale color on the plot. As the color gets lighter, the corresponding value becomes higher in these figures. The upper left corners show results for the reference networks which are sparse networks since there is 10% insertion but 50% deletion as described earlier in this section. As we move from upper parts of the plots to the lower parts, the reference networks get denser. As we move from left to right on the plots, we obtain results for denser reference networks. The densest network is at the lower right corner, since there is 50% insertion and 10% deletion.

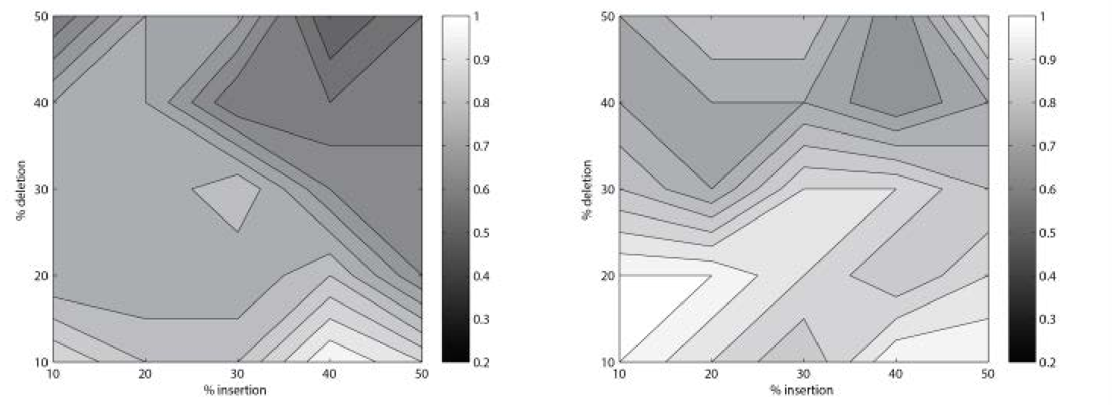
These figures show that the constructed networks by oILP are more accurate than the ones constructed by SiNeC. For all different noise rates, the minimum recall value attained by oILP is 0.6; however, SiNeC's recall value is 0.4. In addition, oILP's recall value is 0.9 at the upper right part of the figure which corresponds to the %50 insertion and %50 deletion network, i.e., the most mutated network. The recall for SiNeC is 0.5 for the same network.

The comparison between oILP and SiNeC for these different levels of noisy reference networks shows that oILP is more robust compared to SiNeC. When SiNeC and oILP are compared by means of precision, we see that oILP has higher precision values. There is a great difference in precision values, especially when the noise rates are high. oILP handles the false positive interactions which are due to using undirected PPI networks, as well as false negative interactions. When some of the constructed networks are examined in detail by means of interactions and genes, it can be seen that if a gene is not a critical gene and if the reference PPI network that is used as a reference network does not have an interaction with this gene (however, it is known that this gene is a member of this signaling network), SiNeC cannot include such a gene in the final network. On the other hand, oILP constructs the network by including all the genes which are known to be a member of the network independent of their type, i.e., whether they are critical or non-critical. Moreover, it constructs a connected network as output. Every node is connected to the network and must contribute to signal transduction. On the other hand, oILP has a drawback that it cannot scale for large networks which have more than 10 genes.



(a) SiNeC, (b) oILP

Figure 2.24 Contour plot for precision values for MAPK fruit fly network.



(a) SiNeC, (b) oILP

Figure 2.25 Contour plot for recall values for MAPK fruit fly network.

2.7.2 Comparison of the results on JAK/STAT Network

In order to make an evaluation about our method, we compared oILP with the 4 methods in the literature, namely SiNeC (Hashemikhabir et al., 2012), InfluenceFlow (Singh, 2011), Ruths et al.`s method (Ruths et al., 2007), and Kaderali et al`s method (Kaderali et al., 2009), using JAK/STAT network. The true JAK/STAT network is shown in Figure 2.26a (Platanias, 2005). The protein-protein interactions are obtained from the EBI IntAct database as in Hashemikhabir et al., 2012 (Figure 2.26b).

Comparison with SiNeC: The constructed network by oILP is exactly the same as the one constructed by SiNeC. This is because both methods use similar idea while constructing of the network. SINEC’s transforms the reference network into a new one which has consistency with the RNAi constraints while providing a minimum distance to the reference network. oILP’s aim is to reconstruct the connected signaling network from the given protein-protein interaction (PPI) network satisfying RNAi data by applying minimum edit operations on the given network using ILP. The main difference between the final outputs is that while oILP is forced to find a complete network, meaning all the genes are a member of a directed pathway from source to sink, SINEC finds the topology consistent with RNAi scores but does not guarantee finding a complete network and cannot handle the noncritical genes which are not connected to the network in the reference network. For JAK/STAT network, because of the reference network topology, both methods find the same final output because of the similarity between the main ideas of the methods. The constructed network is given in Figure 2.26f. oILP constructs the network that is consistent with the RNAi data, meaning that it determines all the critical genes correctly.

Comparison with Ruths et al.`s method: This method can be considered as a simplified version of oILP. It modifies a given reference network to satisfy the given RNAi constraints by inserting new edges only. The JAK/STAT network obtained by Hashemikhabir et al. 2012, using the method of Ruths et al., 2007, is shown in Figure 2.26d. As seen from the constructed network, genes JAK1 and TYK2 are predicted to be two other source nodes for JAK/STAT network and therefore, they do not belong to any signaling pathway. Since these two genes are found to be critical, the method modifies the network such that there is a path from the knocked-down gene to the receptor gene. Also, since the method does not consider a path in the reverse direction, it cannot modify the undirected edges to make them directed. On the other hand, oILP gave a direction to the undirected edge STAT1-IRF9, and added JAK1 and TYK2 genes to the signaling pathway. Although Ruths et al.’s method has a higher recall value, since it does not remove any edges; its precision value is lowered as a consequence.

Comparison with InfluenceFlow: InfluenceFlow constructs a core network that includes only the RNAi hits and ensures satisfaction of the knock-down results. It cannot handle the missing interactions and it gives a network in spanning tree format. For the experiments with InfluenceFlow, raw RNAi scores obtained from Lars Kaderali is used and the genes STAT1, STAT2 and IRF9 are included in JAK/STAT network. As shown in Figure 2.26c, the constructed network by InfluenceFlow does not include the gene STAT1 since it is not an RNAi hit. It also excludes the genes JAK1 and TYK2 since their interactions with other genes are missing in IntAct. It also has two genes, IFNAR1 and IRF9, which are not a member of the signaling pathway from IFNA2 to Luciferase. However, oILP includes STAT1, JAK1 and TYK2 genes in the final network.

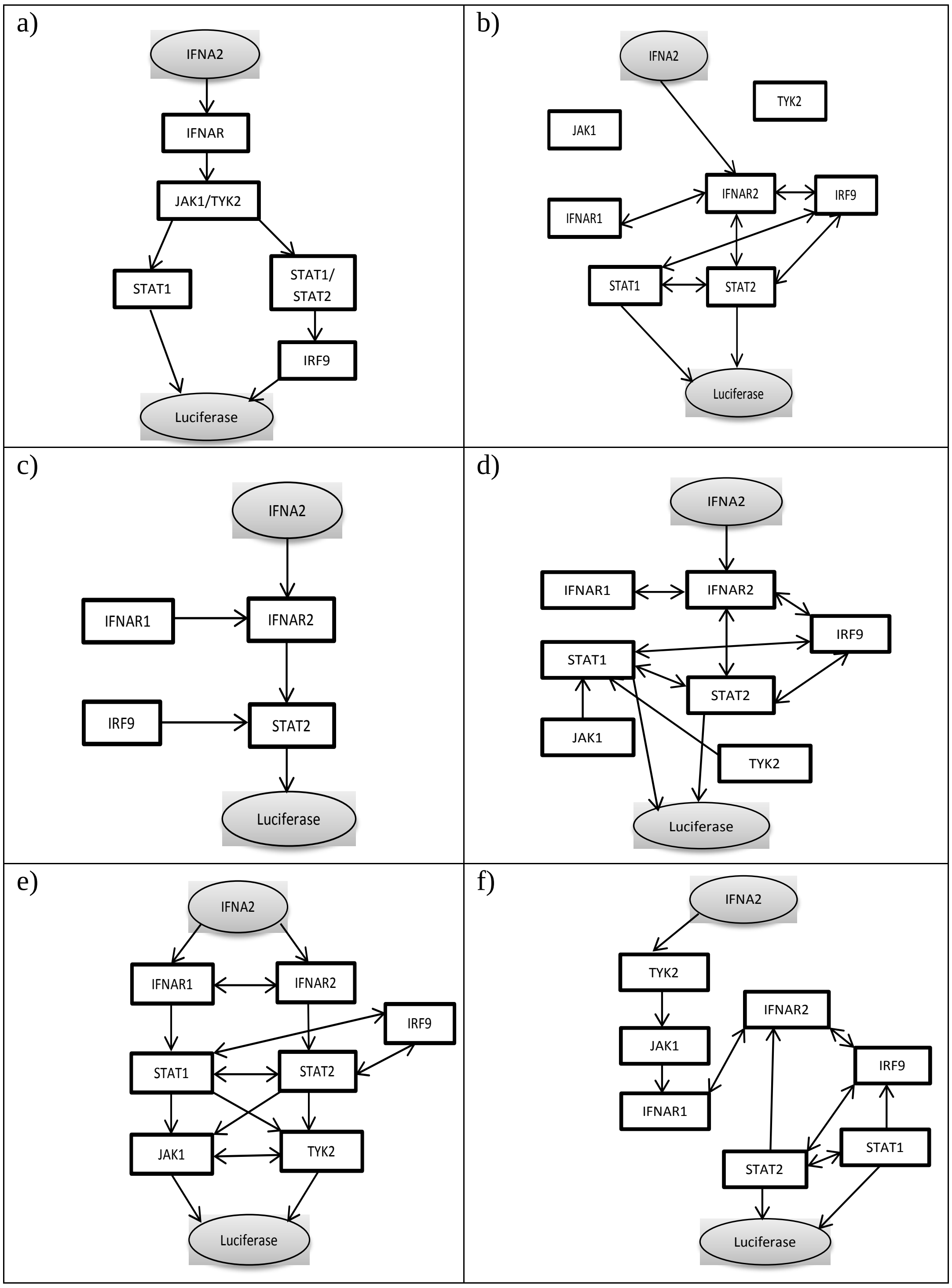


Figure 2.26 Schematic of JAK/STAT network. (a) True network: Type I IFN stimulated JAK/STAT network (Platanias, 2005), (b) Intact network; (Kerrien et al., 2012). The constructed networks by (c) InfluenceFlow, (Singh, 2011), (d) Ruths et al., 2007, (e) Kaderali et al., 2009, (f) SiNeC (Hashemikhabir et al., 2012) and oILP.

Comparison with Kaderali et al.`s method: This method constructs the signaling network from RNAi data alone without using any network as a reference. For comparison of this method and oILP, the network having the highest probability that is reported by Kaderali et al. is used. To simplify the problem, Kaderali et al. uses combined nodes (a single node instead of two nodes) and these nodes are reported as separate nodes by Hashemikhabir et al., 2012. In their result (Figure 2.26e), the most probable topology determines JAK1 and TYK2 as the genes that come just before the reporter, although they interact with the type I IFN receptors, i.e. they have to be located at the topmost of the network. However oILP constructs the network that is consistent with the RNAi data, meaning that it determines all the critical genes correctly.

2.8 Calculation of Number of Constraints

Using the oILP approach described in Section 2.2 of the chapter, it is possible to reconstruct networks with up to 10genes in reasonable time. As the number of genes in a network increases, the number of constraints increases exponentially. Therefore, finding solutions for larger networks gets more and more difficult. In order to solve the problems for larger networks, we need to consider different practical methods. One such method is reducing the number of constraints by using more biological data. For this purpose, in addition to the currently used RNAi data, we use the corresponding RNAi scores. This will help us put the genes in an order with respect to their RNAi scores. These scores provide us the information about the closeness of the genes to the receptor gene. Therefore, using such information reduces the number of constraints. Using this information, we are able to solve larger networks of up to 12-13 genes. In order to understand how much reduction in number of constraints is obtained with such information, we first derive the equations for the number of constraints in a given problem. We start with the most general formulation in which we do not eliminate the common constraints in AND and OR constraints.

Let the network in consideration has n+2 genes including the source gene s and the target gene t. Therefore, the network has n intermediate genes. We will find the number of constraints for pathways including different number of edges, i.e. pathways with 2 edges, pathways with 3 edges, etc. Pathway with only 1 edge is not considered because it is a direct pathway from source to sink (target gene). In each case, one of the genes is knocked down, therefore the number of intermediate genes to be considered becomes n-1. Then, we find the number of constraints when we eliminate them, and finally the number of constraints when we include the RNAi scores, as explained below.

2.8.1 Number of constraints without elimination of common constraints in AND and OR conditions:

For this case, the number of constraints does not change with the type of the gene, i.e. whether the knockdown of the gene affects (critical gene - AND conditions) or does not affect (noncritical gene -OR conditions) the target gene. For example, for the 5-gene network example considered in Section 2.3, there are 4 constraints for each knockdown. However, when the OR constraints are linearized, one additional constraint is added. Therefore, for noncritical genes, one constraint is added to the calculated number of constraints. The derivations for number of constraints are given in the following.

Paths with 2 edges: A 2-edge pathway can be constructed with 3 genes; by putting a gene between the source and the target genes (s – gene – t). As one of the genes is the knockdown gene, n-1 genes are possible to be put as the intermediate gene. Thus, there are n-1 possible 2-edge pathways. For convenience, the number of constraints for this case can be written as

$$\underline{(s)}.\,\underline{(gene)}.\,\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\,\underline{(n-1)}.\,\underline{(1)} = \frac{(n-1)!}{(n-2)!}$$

Paths with 3 edges: Similarly, for a 3-edge pathway 4 genes are required, 2 of which are the source and the target genes (s – gene 1 – gene 2 – t). Therefore, there are 2 intermediate genes for this case. There are n-1 possibilities for the first intermediate gene and n-2 possibilities (since one of the n-1 genes is put in the first place, n-2 genes remain for the second place) for the second intermediate gene. The number of constraints can be found as

$$\underline{(s)}.\,\underline{(gene\ 1)}.\,\underline{(gene\ 2)}.\,\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\,\underline{(n-1)}.\,\underline{(n-2)}.\,\underline{(1)} = \frac{(n-1)!}{(n-3)!}$$

Number of paths with 4 edges, 5 edges, …, n edges can be found accordingly. In summary, the number of paths, i.e. the number of constraints for each critical gene (AND condition) is the summation of number of 2-edge, 3-edge, … , n-edge paths, which are given below:

$$2 - \text{edge paths:} \quad \frac{(n-1)!}{(n-2)!} \tag{2.66}$$

$$3 - \text{edge paths:} \quad \frac{(n-1)!}{(n-3)!} \tag{2.67}$$

$$4 - \text{edge paths:} \quad \frac{(n-1)!}{(n-4)!} \tag{2.68}$$

$$\vdots$$

$$n - \text{edge paths:} \quad \frac{(n-1)!}{0!} = (n-1)! \tag{2.69}$$

The total number of constraints for each critical gene is found by adding all these numbers up;

$$\sum_{i=2}^{n} \frac{(n-1)!}{(n-i)!} = \sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} \qquad (2.70)$$

For a noncritical gene, i.e. OR constraints, one more constraint is added to this number. Therefore, the total number of constraints for a problem with m number of noncritical genes can be found as;

$$(n-m)\sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} + (m)\left(\sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} + 1\right) = m + \sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} \qquad (2.71)$$

Now, we need to add the additional constraints that have to be considered for the noncritical genes to be included in the network, as given by constraints (2.60)-(2.65).

<u>Paths with 2 edges</u>: Only one 2-edge path including the knockdown gene can be constructed:

$$\underline{(s)}.\underline{(KD\ gene)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(1)} = 1$$

<u>Paths with 3 edges</u>: There are 2 possibilities for this case: the knockdown gene can be put in the first or the second place. Then, one of the remaining *n*-1 genes can be put in the other place.

$$\left.\begin{array}{l} \underline{(s)}.\underline{(KD\ gene)}.\underline{(gene\ 2)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(n-1)}.\underline{(1)} = (n-1) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(KD\ gene)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-1)}.\underline{(1)}.\underline{(1)} = (n-1) \end{array}\right\}$$
$$= 2(n-1) = 2\frac{(n-1)!}{(n-2)!}$$

<u>Paths with 4 edges</u>: Similarly, there are 3 possibilities;

$$\left.\begin{array}{l} \underline{(s)}.\underline{(KD\ gene)}.\underline{(gene\ 2)}.\underline{(gene\ 3)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(n-1)}.\underline{(n-2)} = (n-1)(n-2) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(KD\ gene)}.\underline{gene}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-1)}.\underline{(1)}.\underline{(n-2)} = (n-1)(n-2) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(gene\ 3)}.\underline{(KD\ gene)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-1)}.\underline{(n-2)}.\underline{(1)} = (n-1)(n-2) \end{array}\right\}$$
$$= 3(n-1)(n-2) = 3\frac{(n-1)!}{(n-3)!}$$

Number of paths with 5 edges, 6 edges, …, *n* edges for this case can be found accordingly and are summarized below:

$$2-\text{edge paths:} \quad 1 \qquad (2.72)$$
$$3-\text{edge paths:} \quad 2\frac{(n-1)!}{(n-2)!} \qquad (2.73)$$
$$4-\text{edge paths:} \quad 3\frac{(n-1)!}{(n-3)!} \qquad (2.74)$$

.

.

.

.

$$n - \text{edge paths:} \qquad (n-1)\frac{(n-1)!}{[n-(n-1)]!} = (n-1)(n-1)! \qquad (2.75)$$

Adding all these numbers up, we obtain the number of constraints that includes the knockdown gene for each noncritical gene as;

$$\sum_{i=1}^{n-1} i\,\frac{(n-1)!}{(n-i)!} \qquad (2.76)$$

The total number of such constraints for a problem with m number of noncritical genes can be found as;

$$m\left[\sum_{i=1}^{n-1} i\,\frac{(n-1)!}{(n-i)!}\right] \qquad (2.77)$$

In these calculations, the common constraints which appear between the AND-conditions and between the AND- and OR-conditions are not eliminated. As the number of constraints increases exponentially with an increase in number of genes in a network, eliminating such common constraints will reduce the calculation time. In the following section, the number of constraints for such elimination will be derived.

2.8.2 Number of constraints with elimination of common constraints in AND- and OR- conditions:

In this section, we will find the number of constraints after elimination of the common constraints in AND and OR constraints. Assume that first m genes of an n-gene network are critical, i.e. the corresponding constraints are AND-constraints. If the corresponding constraints are written for these m genes, it is seen that there are common ones. For example, the constraint corresponding to the path *s*-2-*t* appears in the constraints written for 1st knockdown, 3rd knockdown,…, *m*th knockdown, which are unnecessary. Only one of them is enough for the formulation. Also, some of the constraints corresponding to noncritical genes, i.e. OR constraints, cannot be satisfied since they are already included in the AND constraints, as explained before (constraint (2.6) cannot be satisfied because of the constraint stated in (2.2), therefore it can be omitted from the formulation). Therefore, such constraints should be eliminated from the formulation of the problem. Note that no elimination should be done between two critical genes because they are separate OR-conditions and should be satisfied independently.

***Elimination between AND-conditions*:**

Let the network has *n*-genes with the first *m*-genes being critical (AND constraints). In order to calculate the number of constraints by eliminating the common constraints, the critical genes are handled one-by-one.

1st critical gene:
For the 1st AND constraint, the number of constraints is the same as given in (2.66)-(2.70). So the number of constraints for the 1st AND-condition is given by

$$\sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} \qquad (2.78)$$

Note that none of these constraints include this gene since it is the knockdown gene.

2nd critical gene:
If the constraints corresponding to this case are written, it is seen that the constraints that do not include 1st critical gene are common with the constraints in the case above. Of course these constraints do not include the 2nd critical gene but such constraints are also present in the 1st case. This means; in this case, all the constraints must include the 1st critical gene, and therefore all others are eliminated. Now, we can obtain the number of constraints again by considering the paths with different number of edges.

Paths with 2 edges: Since the 1st critical gene (1st CG) must be present in the constraints, there is only one such path with 1 edge.

$$\underline{(s)}.\underline{(1^{st}\text{ CG})}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(1)} = 1 = \frac{(n-2)!}{(n-2)!}$$

Paths with 3 edges: In this case, there are two places to put the 1st CG, and *n*-2 different possibilities to put in the other place.

$$\left.\begin{array}{l} \underline{(s)}.\underline{(1^{st}\text{ CG})}.\underline{(gene\ 2)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(n-2)}.\underline{(1)} = (n-2) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(1^{st}\text{ CG})}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-2)}.\underline{(1)}.\underline{(1)} = (n-2) \end{array}\right\} = 2(n-2)$$

$$= 2\frac{(n-2)!}{(n-3)!}$$

Paths with 4 edges: There are 3 possibilities for this case,

$$\left.\begin{array}{l} \underline{(s)}.\underline{(1^{st}\text{ CG})}.\underline{(gene\ 2)}.\underline{(gene\ 3)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(1)}.\underline{(n-2)}.\underline{(n-3)} = (n-2)(n-3) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(1^{st}\text{ CG})}.\underline{(gene\ 3)}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-2)}.\underline{(1)}.\underline{(n-3)} = (n-2)(n-3) \\ \underline{(s)}.\underline{(gene\ 2)}.\underline{(gene\ 3)}.\underline{(1^{st}\text{ CG})}.\underline{(t)} \quad \rightarrow \quad \underline{(1)}.\underline{(n-2)}.\underline{(n-3)}.\underline{(1)} = (n-2)(n-3) \end{array}\right\}$$

$$= 3(n-2)(n-3)$$

$$= 3\frac{(n-2)!}{(n-4)!}$$

Number of paths with 5 edges, 6 edges, …, *n* edges for this case can be found accordingly and are summarized below:

$$2-\text{edge paths:}\quad \frac{(n-2)!}{(n-2)!}=1 \tag{2.79}$$

$$3-\text{edge paths:}\quad 2\frac{(n-2)!}{(n-3)!} \tag{2.80}$$

$$4-\text{edge paths:}\quad 3\frac{(n-2)!}{(n-4)!} \tag{2.81}$$

.

.

.

$$n-\text{edge paths:}\quad (n-1)\frac{(n-2)!}{(n-n)!}=(n-1)! \tag{2.82}$$

Summation of all these numbers gives the number of constraints for the 2nd critical gene, and for convenience, we will use permutation notation *P(k,m)*, i.e. ***m*-permutations of *k*;**

$$\sum_{i=1}^{n-1} P(i,1)\frac{(n-2)!}{(n-(i+1))!} \tag{2.83}$$

3rd critical gene: When 3rd critical gene is considered, it can be seen that the 1st and the 2nd critical genes must appear in all of the constraints. This is because all other constraints are already written in the constraints written for the 1st and the 2nd critical genes. Since all the constraints must include 1st and the 2nd critical genes, then there is no path with 2 edges. So, we start with the paths with 3 edges,

Paths with 3 edges: There are only 2 possibilities for this case;

$$\left.\begin{array}{lcl} \underline{(s)}.\underline{(1^{st}\text{ CG})}.\underline{(2^{nd}\text{ CG})}.\underline{(t)} & \rightarrow & \underline{(1)}.\underline{(1)}.\underline{(1)}.\underline{(1)}=1 \\ \underline{(s)}.\underline{(2^{nd}\text{ CG})}.\underline{(1^{st}\text{ CG})}.\underline{(t)} & \rightarrow & \underline{(1)}.\underline{(1)}.\underline{(1)}.\underline{(1)}=1 \end{array}\right\} = 2 = P(2,2)\frac{(n-3)!}{(n-3)!}$$

Paths with 4 edges: Similar calculations can be done for this case with permutation of 1st and the 2nd critical genes. An example of 4-edge path is $\underline{(s)}.\underline{(1^{st}\text{ CG})}.\underline{(2^{nd}\text{ CG})}.\underline{(gene\ 3)}.\underline{(t)}$. Other possible paths are obtained by permuting the intermediate genes and 6 different paths are found. The number of constraints for this case is expressed as

$$P(3,2)\frac{(n-3)!}{(n-4)!}$$

Number of paths with 5 edges, 6 edges, …, *n* edges for this case can be found accordingly and are summarized below:

$$2 - \text{edge paths:} \qquad 0 \tag{2.84}$$

$$3 - \text{edge paths:} \qquad P(2,2)\frac{(n-3)!}{(n-3)!} \tag{2.85}$$

$$4 - \text{edge paths:} \qquad P(3,2)\frac{(n-3)!}{(n-4)!} \tag{2.86}$$

.

.

.

$$n - \text{edge paths:} \qquad P(n-1,2)\frac{(n-3)!}{(n-n)!} \tag{2.87}$$

Adding all these numbers up, we obtain the number of constraints that includes the critical genes as

$$\sum_{i=2}^{n-1} P(i,2)\frac{(n-3)!}{(n-(i+1))!} \tag{2.88}$$

The index *i* starts from 2 since all of the constraints must include the 1st and the 2nd critical genes.

mth critical gene: Similarly, for *m*th critical gene, the number of constraints can be expressed as

$$\sum_{i=m-1}^{n-1} P(i,2)\frac{(n-m)!}{(n-(i+1))!} \tag{2.89}$$

In order to obtain a general expression for the total number of constraints for critical genes, we need to express Eqn. (2.78) similar to Eqns. (2.83) and (2.88) with permutation notation, which is given below:

$$\sum_{i=1}^{n-1} \frac{(n-1)!}{(n-(i+1))!} = \sum_{i=0}^{n-1}\left[P(i,0)\frac{(n-1)!}{(n-(i+1))!}\right] - 1 \tag{2.90}$$

A summary of the derivations for the number of constraints for each critical gene with the elimination of common constraints is given below:

$$\text{1st critical gene:} \quad \sum_{i=0}^{n-1}\left[P(i,0)\frac{(n-1)!}{(n-(i+1))!}\right] - 1 \tag{2.91}$$

$$\text{2nd critical gene:} \quad \sum_{i=1}^{n-1} P(i,1)\frac{(n-2)!}{(n-(i+1))!} \tag{2.92}$$

$$\text{3rd critical gene:} \quad \sum_{i=2}^{n-1} P(i,2)\frac{(n-3)!}{(n-(i+1))!} \tag{2.93}$$

.
.
.

mth critical gene: $\sum_{i=m-1}^{n-1} P(i, m-1) \frac{(n-m)!}{(n-(i+1))!}$ (2.94)

Summing all these constraints up, we obtain the total number of the constraints for critical genes with the elimination of common constraints as;

$$\sum_{j=1}^{m} \sum_{i=j-1}^{n-1} \left[P(i, j-1) \frac{(n-j)!}{(n-(i+1))!}\right] - 1 \qquad (2.95)$$

***Elimination between AND and OR-conditions*:**

Elimination of the common constraints between AND and OR-constraints are similar to the elimination done above, however, this time, we will find the number of constraints only for the 1st noncritical gene as the number of constraints for the other noncritical genes is the same as the number of constraints for the 1st noncritical gene. We will now explain in detail why the numbers of constraints for noncritical genes are the same. Assume that for an n-gene network (excluding the source and the target genes), there are m critical genes, and we wrote down all the corresponding constraints for the knockdowns of these m-genes. Expressing the constraints for the (m+1)th gene, which is a noncritical gene, is no different than expressing the constraints as if it is a critical gene. This is because the common constraints between this gene and the previous ones should be eliminated as it is done before between the critical genes. The difference starts with the (m+2)th gene. This time we cannot eliminate the common genes between the (m+1)th gene and (m+2)th gene, because both are noncritical genes and their constraints are connected by logical OR and they have to be treated separate from each other. The common constraints between the m critical genes and the (m+1)th gene are eliminated first and then, the same is done for (m+2)th gene. Therefore, although the eliminated constraints are different, the numbers of them are the same and this is valid for all noncritical genes.

Derivation of the number of constraints for noncritical genes with elimination of the common genes can be done similar to the one for critical genes. Assuming that (m+1)th gene is also a critical gene and similar to the Eqn. (2.94), we obtain

$$\sum_{i=m}^{n-1} P(i, m) \frac{(n-m-1)!}{(n-(i+1))!} \qquad (2.96)$$

Since this number is the same for the rest of the noncritical genes, it is multiplied by (n-m) to obtain the total number of constraints for noncritical genes, which is expressed as

$$(n-m) \sum_{i=m}^{n-1} P(i, m) \frac{(n-m-1)!}{(n-(i+1))!} \qquad (2.97)$$

As it is stated before, we need additional constraints to have all the genes connected to the network. Otherwise, some of the genes may just rest in the space; they may have no connection or they may act as a target gene. We input these additional constraints only for the noncritical genes. These constraints consist of the paths which pass through the knockdown gene and they are all connected by logical ORs; i.e., at least one constraint must be satisfied. To linearize them, we again use the procedure explained before by adding one more constraint to the each group of noncritical gene constraints. Since we have now the elimination of common genes, each group of constraints must include the critical genes and the knockdown gene.

Let us consider this situation with an example and let our network has only one critical gene. Again, we first write down the constraints for critical gene(s). When writing down the constraints for the noncritical genes, we need to consider paths with different edge numbers as before. There is no path with 2 edges since the paths must include both the noncritical genes and the critical gene. We need at least 3 edges for such paths. Since both the noncritical genes and critical gene should be included in the constraints, there are only 2 paths with 3 edges. As we continue counting the number of constraints similarly for other paths with different number of edges, we see that the formulation is exactly the same with Eqns. (2.84)-(2.88). We can generalize the expression for the number of constraints for *m* critical genes with elimination of the common genes as

$$\left[\sum_{i=m+1}^{n-1} \frac{(n-m-1)!}{(n-i-1)!} P(i, m+1)\right](n-m) \qquad (2.98)$$

where the term in brackets is the number of constraints for each noncritical gene and (*n*-*m*) is the number of such constraints.

A 12-gene example:

In this section, we discuss the resulting number of constraints on a 12-gene network as an example. The number of constraints for different number of critical genes is given in Table 2.2 for such a network. The calculations follow the formulations given above. Each column in the first part of the table shows the number of constraints for the corresponding critical gene. Since the common constraints are eliminated, the number of constraints for each critical gene is different. For example, if there are 3 critical genes in the network, then the numbers of constraints for these genes are 986409 for the 1st critical gene, 1863218 for the 2nd gene, and 780908 for the 3rd gene. Their sum is given in the CUMULATIVE (A) row as 2644116. The rest of the genes are noncritical genes, therefore for the number of constraints of each of these genes (OR-conditions), we need to look at the number given in the TOTAL row of the 4th critical gene which is 696750. Since there are 7 noncritical genes, this number should be multiplied by 7. And also, since we add one more constraint to make each of them linear, we add 7 to this number. We also add additional constraints to the OR-conditions to connect all genes to the network as explained before and the numbers of these additional constraints are given in the second part of the table. For our example (3 critical genes, 7 noncritical genes), this number is given in TOTAL (C) row as 20260975. The GRAND-TOTAL row shows the sum of all the constraints.

As it is seen from the results, the number of constraints decreases with the increasing number of critical genes in the network, i.e. with the decreasing number of noncritical genes. The reason for such reduction in the constraints is that additional constraints are required for each noncritical gene. For example, for a network having two critical genes, 1863218 constraints are required for critical genes, and 780908 constraints are required for the rest of the genes (which are noncritical genes) and an additional 3591174 constraints. This much additional constraints for such problem is approximately 4.5 times the number of constraints for each noncritical gene, therefore the total number of constraints decreases with decreasing number of noncritical genes.

2.8.3 Effect of RNAi Scores

As the number of genes in a network increases, the number of constraints increases exponentially. In order to be able to construct larger networks, we may reduce the number of constraints by using RNAi scores. RNAi scores provide us the information about the closeness of the genes to the reporter or the receptor gene. In order to understand how much reduction in number of constraints is obtained with such information, we can derive the equations for the number of constraints in a given problem as we did above. However, the derivation gets harder and impractical when RNAi scores are used. Instead, we count the number of constraints using the computer program we developed. To show how RNAi scores affect the number of constraints for a given problem, we considered the 12–gene example given above for 3 different cases depending on the RNAi scores.

Case 1: The first case is the one we already considered above which has no RNAi scores.

Case 2: In the second one, only one of the gene is decided to be closer to the reporter gene depending on the RNAi scores. Note that this gene is a critical gene.

Case 3: In the third example, two genes are decided to be closer to the reporter gene, however they both are considered to have the similar RNAi scores, therefore no judgment between them about their closeness to the reporter gene is made.

Table 2.2 Number of constraints for a 12-gene network consisting of different number of critical genes

	1st CG	2nd CG	3rd CG	4th CG	5th CG	6th CG	7th CG	8th CG	9th CG	10th CG
2-edge paths	9	1	0	0	0	0	0	0	0	0
3-edge paths	72	16	2	0	0	0	0	0	0	0
4-edge paths	504	168	42	6	0	0	0	0	0	0
5-edge paths	3024	1344	504	144	24	0	0	0	0	0
6-edge paths	15120	8400	4200	1800	600	120	0	0	0	0
7-edge paths	60480	40320	25200	14400	7200	2880	720	0	0	0
8-edge paths	181440	141120	105840	75600	50400	30240	15120	5040	0	0
9-edge paths	362880	322560	282240	241920	201600	161280	120960	80640	40320	0
10-edge paths	362880	362880	362880	362880	362880	362880	362880	362880	362880	362880
TOTAL	986409	876809	780908	696750	622704	557400	499680	448560	403200	362880
CUMULATIVE **(A)**	986409	1863218	2644126	3340876	3963580	4520980	5020660	5469220	5872420	
Corresponding NCG const. # (B)	876809*9+9 =7891290	780908*8+8 =6247272	696750*7+7 =4877257	622704*6+6 =3736230	557400*5+5 =2787005	499680*4+4 =1998724	448560*3+3= 1345683	403200*2+2= 806402	362880*1+1= 362881	
CG: critical gene, NCG: noncritical gene										
Additional constraints to connect all genes to the network: (only for critical genes)										
2-edge paths	0	0	0	0	0	0	0	0	0	
3-edge paths	2	0	0	0	0	0	0	0	0	
4-edge paths	48	6	0	0	0	0	0	0	0	
5-edge paths	672	168	24	0	0	0	0	0	0	
6-edge paths	6720	2520	720	120	0	0	0	0	0	
7-edge paths	50400	25200	10800	3600	720	0	0	0	0	
8-edge paths	282240	176400	100800	50400	20160	5040	0	0	0	
9-edge paths	1128960	846720	604800	403200	241920	120960	40320	0	0	
10-edge paths	2903040	2540160	2177280	1814400	1451520	1088640	725760	362880	0	
SUBTOTAL	4372082	3591174	2894424	2271720	1714320	1214640	766080	362880	0	
TOTAL **(C)**	4372082*9+9= 39348747	3591174*8+8= 28729400	2894424*7+7= 20260975	2271720*6+6= 13630326	1714320*5+5= 8571605	1214640*4+4= 4858564	766080*3+3= 2298243	362880*2+2= 725762	0	
GRAND TOTAL **(A)+(B)+(C)**	48226446	36839890	27782358	20707432	15322190	11378268	8664586	7001384	6235301	

The number of constraints for these 3 cases and the corresponding % reduction in constraints are given in Table 2.3. In order to visualize the differences, the corresponding graphs are also given in Figure 2.27. The first row of Case 2 in Table 2.3 is empty because of the fact that if there is only one critical gene, then there is no need for an RNAi score, i.e. no ordering between the genes can be made. The first two rows of Case 3 in Table 2.3 are also empty because as stated before, we considered that the closeness of the two critical genes is the same in all situations. Therefore, when the network has two critical genes, then no ordering can be made between these genes.

Table 2.3 Number of constraints for 12-gene networks with different number of critical genes

	Number of constraints				
# of critical genes	**Case 1**	Case 2	% reduction	Case 3	% reduction
1	**48226446**	-	-	-	-
2	**36839890**	19351562	47.5	-	-
3	**27782358**	10242638	63.1	10242638	63.1
4	**20707432**	6162787	70.2	4208996	79.7
5	**15322190**	4050790	73.6	2134022	86.1
6	**11378268**	2882783	74.7	1262588	88.9
7	**8664586**	2224210	74.3	850986	90.2
8	**7001384**	1861585	73.4	641480	90.8
9	**6235301**	1679221	73.1	529405	91.5

When we compare Case 2 with Case 1, we see that at least 47.5% reduction is obtained for a 12-gene network consisting of two critical genes with knowledge of RNAi scores for these two genes. If the network has more than 5 critical genes, more than 73% reduction is possible.

When two genes are determined to be closer to the reporter gene as in Case 3, the reduction in number of constraints becomes more, as expected. At least 63.1% reduction in the number of constraints occurs, and even a reduction of 91.5% is possible as it is seen from the table.

The reason why 63.1% reduction is the same for Case 2 and Case 3 can be explained as follows: In both cases, there are 3 critical genes. According to the RNAi scores, 1 gene is stated to be closer to the reporter gene in Case 1. However, in Case 2, it is stated that 2 genes are closer to the reporter gene, which can alternatively be stated as 1 gene is closer to the receptor gene. In terms of number of constraints, there is no difference in stating that one gene is closer to the reporter gene or closer to the receptor gene and that's why the number of constraints for Case 2 and Case 3 are the same.

We also studied on some examples to see the effect of using more information (RNAi scores) on the number of constraints. We considered again a 12-gene network with 5 critical genes and numbered these 5 genes from 1 to 5. Depending on the RNAi scores, we assumed several different orderings between the critical genes and calculated the corresponding number of constraints. The following orders of genes are assumed: gene 1 is closer to the source gene than (a) gene 2, (b) genes 2 and 3, (c) genes 2, 3 and 4, (d) genes 2, 3, 4, and 5. Additionally, we considered the following situations: (e) gene 1 is closer to the reporter gene than gene 2, and gene 2 is closer to the reporter gene than gene 3, (f) gene 1 is closer to the reporter gene than gene 2 and gene 2 is closer to the reporter gene than both genes 3 and 4, (g) and lastly, all genes are ordered from smallest id to largest id, e.g. 1-2-3-4-5.

The numbers of constraints for these 7 examples are given in Table 2.4 together with the reduction percentage in the constraints compared to the case when no RNAi score is available. A minus (-) symbol means that ordering is made between those genes and comma (,) means the reverse, i.e. no ordering is made between the genes separated by a comma. For example, 1-2-3, 4 means that gene 1 is closer to the reporter gene than gene 2, and gene 2 is closer to the reporter gene than genes 3 and 4, therefore gene 1 is closer than genes 3 and 4 also. The table shows that one more gene is added into the ordering as we proceed from example (a) to example (d) and its order is compared with the order of gene 1. Since more information is added to the system, the number of constraints decreases. While the reduction is 43.9% for example (a), it is 73.6% for example (d), which is a significant improvement. However, more significant reduction is also possible by ordering the genes between each other, e.g. compare examples (b) and (e). In both examples, there is ordering between 3 genes, but in addition to the information of example (b), an ordering is made also between genes 2 and 3 in example (e). This additional information results in an additional 76.6%-60.3%=16.3% reduction in the number of constraints. In example (g), we see a 97.3% reduction in the number of constraints since all critical genes are in an order with respect to their RNAi scores.

2.9 Conclusion

In this chapter, we present a formulation for the network reconstruction problem as a linear optimization problem which provides a network satisfying the RNAi data with minimum edit operations applied on the given reference PPI network. By using our approach we guarantee to find the optimal solution and a connected network topology as a final network. We show that the scalability of the problem exponentially increases as the number of genes in the network increases. We tried to decrease the dimensionality of the problem by adding some additional constraints obtained from RNAi scores; however, it allows addition of only a few genes in the final network topology. We validate our proposed approach on real and synthetic data sets, and comparison with the state of the art shows that our proposed approach is able to scale better for small networks of size up to 10 genes while attaining similar or better biological accuracy.

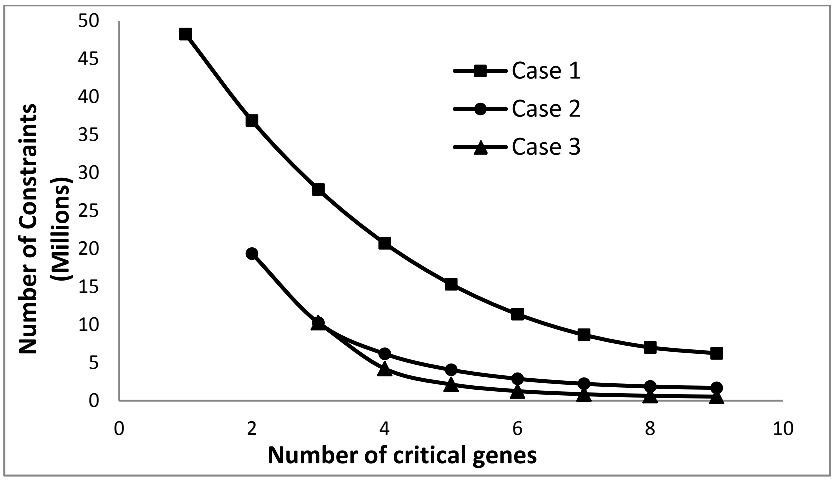


Figure 2.27 Number of critical genes vs. number of constraints for 3 cases

Table 2.4 Examples with different RNAi scores

Order of genes	Number of constraints	% reduction
a) 1-2	8592709	43.9
b) 1-2,3	6089246	60.3
c) 1-2,3,4	4816475	68.6
d) 1-2,3,4,5	4050790	73.6
e) 1-2-3	3585783	76.6
f) 1-2-3,4	1265133	91.7
g) 1-2-3-4-5	406845	97.3

CHAPTER 3

CONSTRUCTION OF LARGE-SCALE NETWORKS

In this chapter, we present the methods we have developed to construct large-scale biological networks (Eren Ozsoy and Can, 2013). As shown in the previous chapter, as the number of nodes increases in a network, the number of constraints increases exponentially. Since the maximum number of nodes that can be handled in reasonable time with a standard desktop with our approach is 10, it is not possible to solve the networks that have larger number of nodes because of the complexity of the problem. To solve such problems, we propose the divide and conquer technique which allows us to reduce the number of nodes for a large network by dividing it into sub-networks that have up to 10 nodes, then to solve each sub-network separately, and finally to combine the obtained results. Since there is a limitation for the maximum number of nodes in a single sub-network for the previously developed method, the original network has to be divided into sub-networks in such a way that no more than 10 nodes should be allowed.

We apply our method on several networks, which can be sparse or dense, have small or large number of nodes, and contain noisy data. The results show that our methods can construct networks better than the state of the art methods in terms of recall and precision measures, scalability, robustness, and consistency with the RNAi data. For some cases, the state of the art methods cannot place a non-critical gene in the constructed network if the connection of this gene to the reference network is missing, although it is known that this gene is part of a signal transducing path in the network. On the other hand, our method is able to include such genes in the reconstructed network resulting in a more biologically correct network. For dense and large networks, while the state of the art methods fail to construct networks in 1 hour, we are able to find solutions in minutes.

3.1 General divide and conquer algorithm

The basic principle of divide and conquer algorithm is partitioning the complex problem into subproblems and then solve each of them. It is assumed that these subproblems, which are themselves smaller instances of the main problem, are easily solvable. If all these subproblems are solved correctly and integrated together, the solution of the complex problem is said to be found out.

Therefore, a divide and conquer algorithm has three main steps;

1- Divide the main problem into subproblems

2- Solve these subproblems recursively to conquer.
3- Integrate the solution of subproblems to find the solution of the main problem.

This is a promising approach for the solution of large network reconstruction problems. We use different strategies for different number of gene to number of critical genes ratios as explained in the next section.

3.2 The Divide and conquer approach used for construction of large-scale networks

As the number of genes increases in a network, the number of constraints increases exponentially (see Chapter 2). Since the maximum number of genes that can be handled in reasonable time with a standard desktop with our approach is 10, it is not possible to solve large networks. In order to scale for larger networks, we propose using a divide and conquer approach which allows us to employ the original ILP formulation on small sub-networks up to 10 genes and combine the obtained results.

In this approach, partitioning the whole network into sub-networks is the most important process. We use RNAi data not only to find the RNAi hits by treating the data as Boolean after thresholding to find out the critical genes but also we use the raw RNAi scores to decide the placement of the gene mentioned in Chapter 2. RNAi scores and the reference network (PPI) play an important role in deciding which sub-network includes which genes. First, the critical genes are determined according to RNAi data. Then, two different approaches are applied to divide the original network to sub-networks depending on the ratio of the number of total genes to critical genes:

i) ratio of the number of total genes to critical genes is greater than 10. We propose two different division algorithms for this case which are detailed below.
ii) ratio of the number of total genes to critical genes is less than 10.

Since the largest network that can be solved by the oILP method is a 10 node network, the original network has to be divided into sub-networks in such a way that each sub-network has at most 10 nodes.

In the combine phase of this approach, the reconstructed sub-networks are simply combined together from their common nodes, which are the articulation points (critical genes) to obtain the wholly reconstructed network.

i) Horizontal Divide (# of genes/# of critical genes > 10):
(a) The first approach, named as RNAiDivide, does not take the structure of the reference network into account and divides the network into sub-networks using only the RNAi scores. For this case *(i)*, since we have a large network with small number of critical genes, the number of critical genes is not enough to divide the network into sub-networks which are smaller than 10 gene networks. Therefore, due to the small number of critical genes, all the critical genes are included in all sub-networks.

Then, the remaining genes are ordered with respect to their RNAi score values. After ordering, all genes are grouped into sub-networks having the same source (the receptor gene of the main network), the same target (the target gene of the main network) and the same critical genes. Then, according to the ordering, the rest of the non-critical genes are included in the sub-networks equally. Note that the last sub-network may have smaller number of genes than the others if the network cannot be divided equally.

A schematic of the structure of such a divided network is given in Figure 3.1. The sub-networks obtained by such division may have genes between all critical genes. Therefore, for example, sub-network-A includes the receptor gene *s*, the target gene *t*, all the critical genes 2-3-4-…, and some non-critical genes between these critical genes.

This procedure has a drawback since the division of the network causes removal of some of the original edges (interactions) in the reference network. These edges are directed from a sub-network gene to another sub-network gene (upstream or downstream) and a remedy is possible with a difference: since some of the genes belonging to separate sub-networks can be in-between two consequent articulation points (critical genes) (e.g., between gene 1 and gene 2, from sub-network A to sub-network B as in Figure 3.1), such edges can be inserted into the constructed network without any violations in our ILP approach. Algorithm 1 presents the pseudocode for RNAiDivide.

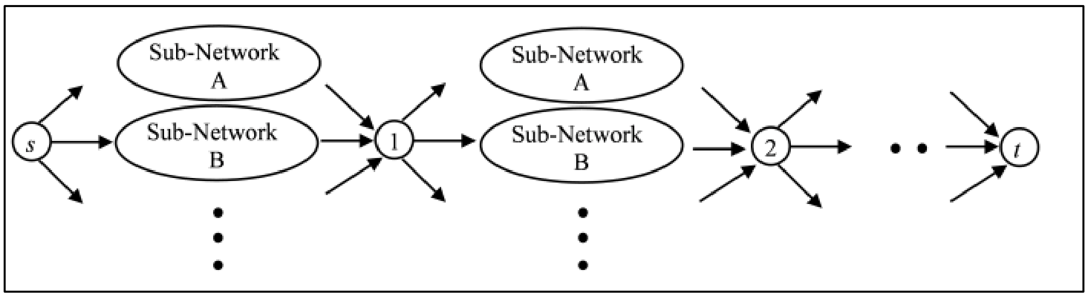


Figure 3.1 Structure of the divided network produced by RNAiDivide. Each sub-network has the same source and sink (*s*, *t*), and includes all the critical genes (1, 2, ... etc.) and also the genes between the critical genes which belong to itself. The combined network has the same structure with changes applied in the paths between the critical genes.

Our experimental results show that as the total number of genes increases, the resulting constructed network deviates too much from the reference network with the RNAiDivide approach.

Algorithm 1. RNAiDivide

```
1: If (# of genes/# of critical genes > 10) then
2:     Put the critical and non-critical genes in ascending order with respect to their
       RNAi scores.
3:         While (# of genes in non-critical gene set) + (# of critical genes) > 10 do
4:             initialize a sub-network by assigning all the critical genes as members
               of the sub-network
5:             insert first x genes from ordered non-critical gene set to the sub-
               network such that total genes in the sub-network=10
6:             remove inserted genes from the non-critical gene set
7:         end while
8:     call oILP for each sub-network
9:             for all consecutive critical gene pairs do
10:                    if there are deleted interactions between sub-networks then
11:                    insert deleted interactions to the oILP solution
12:                    end if
13:            end for
14: end if
15: combine all solutions together from their critical genes.
```

(b) In order to increase the accuracy of our division method for case *(i)*, we propose an alternative approach which takes into account the reference network topology along with the RNAi scores and name this method as TopologyRNAiDivide. The RNAi scores of the critical genes alone are not enough to make a decision about their locations –whether they are close to the source or to the sink- in the network. Therefore, the order of the critical genes should be determined by considering the given reference network structure and in some situations by utilizing the RNAi scores. We also use RNAi scores together with reference network structure when dividing the non-critical genes, if the sub-network of these genes cannot be determined from the reference network structure.

The order of the critical genes can be determined easily if all of the genes are connected to the reference network. This can be done by a breadth-first-search (BFS) of the network. BFS starts with the receptor gene and continues with the neighboring genes. Then it searches the neighbors of these genes in turn. By inspecting the final search output, the order of the critical genes in the reference network can be determined since all the critical genes are known initially. If the reference network is a sparse network with some missing interactions we cannot place some of the genes using this BFS approach. In such situations, we utilize the RNAi scores. We place such genes by using the location of a previously placed gene with the closest RNAi score.

After finding the order of the critical genes, we determine the genes in each sub-network. Each gene belongs to a specific sub-network between two critical genes, or between the receptor gene and the first critical gene, or between the last critical gene and the target gene. In order to assign genes to sub-networks, we start by finding the genes which are neighbors of the receptor gene. If an edge is directed towards a non-critical gene from the receptor, then it is put into the first sub-network. Next, we continue assigning immediate neighbors of first sub-network to that sub-network, until a critical gene is confronted. At this point, since all the genes which belong to the first network are determined, we continue with the first non-critical gene that comes after the receptor gene. Then, applying the same procedure, the genes belonging to the second sub-network (i.e., which are downstream neighbors of the second critical gene) are determined. The genes belonging to the remaining sub-networks can be determined in the same way until no genes remain to assign to a sub-network.

In some cases, the sub-network may contain too many genes which is a problem for the ILP approach. For these cases, the sub-network has to be divided again into two or more sub-networks including the common receptor and target genes. Here, instead of dividing arbitrarily, this division has to be made in such a way that the resulting secondary sub-networks should include as much information as possible from the reference network. This way, the constructed network will be closer to the actual network, i.e., biologically more accurate, since the reference network includes several edges from the actual network. The order of genes can be determined by using breadth-first-search on sub-networks. The first and the last nodes in such sub-networks are common. Since only a limited number of critical genes are included in the sub-networks, the required number of sub-networks is less than that of the RNAiDivide approach. Therefore, the problem can be solved with less computational effort with higher accuracy. In Figure 3.2, a schematic of the structure of a network divided by TopologyRNAiDivide is given.

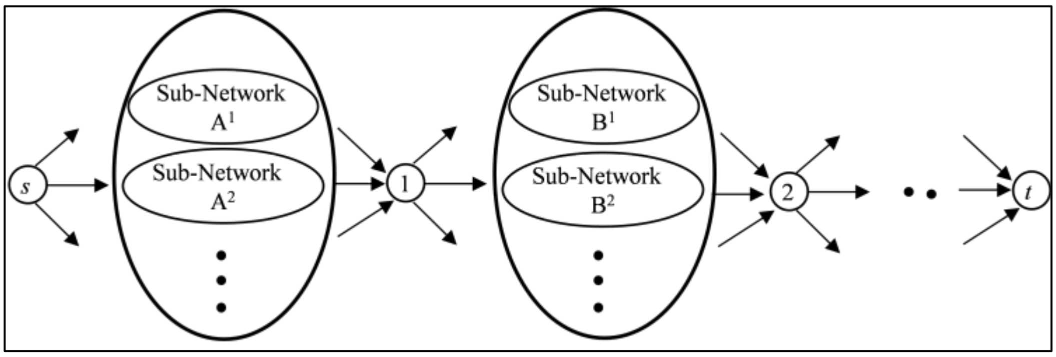


Figure 3.2 Structure of the divided network produced by TopologyRNAiDivide. Each group of sub-networks (sub-network A, sub-network B, ...) has the same source and sink (for e.g. the sink and source of group B sub-networks are gene 1 and 2.). After these sub-networks are solved, they are put in their respective places in this structure.

The algorithm for TopologyRNAiDivide is given below.

Algorithm 2. TopologyRNAiDivide

```
1: If (# of genes/# of critical genes > 10) then
2:     run BFS on the given reference network to order the critical genes
3:         if (# of critical genes < # of critical genes from BFS final output)
4:             Place the missing critical genes by using the location of a
               previously placed gene with the closest RNAi score.
5:         end if
6:     for all non-critical genes do
7:         Locate non-critical genes by labeling them according to their
           downstream neighbors
8:     end for
9:     for all sub-networks do
10:        Count the # of genes in all sub-networks
11:            if (# of genes in the sub-network > 10)
12:                run BFS on the sub-network
13:                divide again into two or more 10-node sub-networks
                   including the common receptor and target genes by
                   BFS final output.
14:            end if
15:    end for
16:    run oILP for each sub-network
17:    Combine all solutions together from their critical genes
18:    for all consecutive critical gene pairs do
19:        if there are deleted interactions between sub-networks then
20:            insert deleted interactions to the oILP solution
21:        end if
22:    end for
23: end if
```

ii) *VerticalDivide (# of genes/# of critical genes < 10)*:
We perform this division technique when ratio of the number of total genes to critical genes is less than 10. For this case, we both use the RNAi scores and the reference network to partition the whole network into sub-networks as in TopologyRNAiDivide described above.

The main idea is similar to that of TopologyRNAiDivide. However, this time, the sub-networks may contain more than one critical gene. If the RNAi scores are sufficient to decide the location of critical genes with respect to each other, such as when some of the critical genes have higher or lower scores than the other critical genes, we utilize these scores and their relative position in the reference network to order the genes. If the RNAi scores are sufficient to order all critical genes in the sub-network, genes are put in the order from the one having the lowest RNAi score to the one having the highest RNAi score. Such kind of division decreases the solution time according to the lowest number of constraints due to ordering of genes and make the final topology close to the true signaling network.

Also, we assume that these sub-networks are connected to each other at some common genes which are in fact the articulation points (critical genes) of the main network. Therefore, while such genes act as a source for a sub-network, they act as a target gene for an upstream sub-network. Note that for the first sub-network, the receptor gene coincides with the receptor gene (*s*) of the main network, and for the last sub-network, the target gene coincides with the target gene (*t*) of the main network. The genes between the networks are the critical genes or non-critical genes and the critical genes can be ordered with respect to their RNAi scores. A schematic of the structure of such a divided network is given in Figure 3.3. However, the last sub-network may have smaller number of genes than the others if the network cannot be divided equally. This situation does not create a problem since we solve the problem by considering each sub-network separately.

After the reference network is divided into sub-networks, we use the induced sub-graphs to create smaller ILP instances. Removal of inter sub-network edges from the original reference network is a drawback of the divide and conquer approach since it causes the constructed network to be sparser. However some of these edges can be included in the final constructed network. If the removed edge is an upstream edge which is directed from a gene of a downstream sub-network to the gene of an upstream sub-network, then we can insert such an edge to the solution without violating the RNAi constraints. Therefore, we obtain a network more similar to the reference network. Algorithm 3 presents the pseudocode for VerticalDivide.

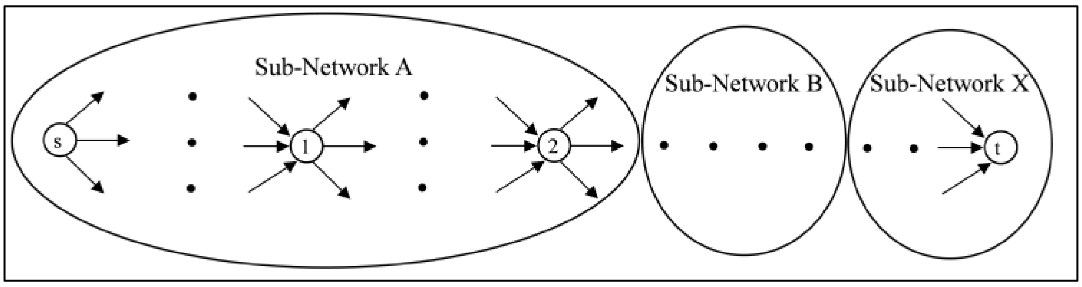


Figure 3.3 Structure of the divided network produced by VerticalDivide. Each sub-network includes a couple of critical genes, and they are connected end to end. Therefore, 2 consecutive sub-networks share a common critical node, which is the sink for the first network and the source for the second network. After the solution, these sub-networks are connected back to each other from these common genes.

Algorithm 3. VerticalDivide

```
1: If (# of genes/# of critical genes < 10) then
2:     run BFS on the given reference network to order the critical genes
3:         if (# of critical genes < # of critical genes from BFS final output)
4:             Place the missing critical genes by using the location of a
               previously placed gene with the closest RNAi score.
5:         end if
6:     for all non-critical genes do
7:         Locate non-critical genes by labeling them according to their
           downstream neighbors
8:     end for
9:     while # of genes in sub-network < 10 do
10:        add 2 consecutive critical genes to the sub-network together with the
           non-critical genes between them
11:    end do
12:    run oILP for each sub-network
13:    Combine all solutions together from their common critical genes
14:    for all sub-networks do
15:        if there are deleted upstream interactions between sub-networks then
16:            insert deleted interactions to the combined oILP solution
17:        end if
18:    end for
19: end if
```

3.3 Data sets

To evaluate the performance of the divide and conquer approach we apply our method on several networks, which are classified as synthetic, semisynthetic and real networks. As the real network, we use ERBB network taken from the literature (Sahin et al., 2009). Semi-synthetic networks are obtained by adding noise to several real networks at different noise rates. The synthetic and semi-synthetic networks vary by sparseness or denseness, number of nodes they include and the noise that they have. The details about these data sets are given below.

Real data set: We use the ERBB receptor-regulated G1/S transition network (Sahin et al., 2009)) given in Figure 3.4a as the real data set. The ERBB network is a large network with 20 nodes, each of which can be either a gene or a complex of proteins.

Semisynthetic data sets: In order to compare the performances of our Horizontal Divide methods TopologyRNAiDivide and RNAiDivide on a semisynthetic dataset, we generate 20 reference networks for the VEGF_PGI2 network by the edge shuffling method (Milo et al., 2003) with the noise rates of $r = 0.05$, $r = 0.1$, $r = 0.2$, and $r = 0.4$. For each network, $r \times |E|$ edges are deleted or inserted to generate random networks. The first 5 of these 20 networks are generated with a noise rate of 0.05, the second 5 networks with a noise rate of 0.1, the third 5 networks with a noise rate of 0.2, and the last 5 networks with a noise rate of 0.4.

In addition, to compare our methods with the state of the art methods, we use a selection of five signaling networks from the KEGG database (Kanehisa and Goto, 2000). One of them is HSA2 network that is used by Hashemikhabir et al 0. HSA2 network has 388 genes and 615 interactions. We named the rest of the networks as VEGF_PGI2, p53_apoptosis, neutrotrophin and FceRI (descriptions, number of genes, |V|, and number of edges, |E|, are given in Table 3.1). Using these five networks, we have generated 25 semisynthetic data sets for each actual network (a total of 125 networks). This is done by inserting edges to the actual networks at a certain rate and removing edges at the same or at a different rate, i.e., inserting or removing different number of edges from each of them. Therefore, both dense and sparse networks have been generated. The insertion/deletion rates range from 10% to 50%, each. In this data set, we have 25 unique reference networks with different number of edges and we name this data set generation method as ADDRIP ("addition/deletion rates in percentages").

We also generated 200 reference networks from HSA2 data set 0 by modifying the original network using the degree preserving edge shuffling method (Milo et al., 2003) with varying noise rates, i.e., 0.05, 0.1, 0.2, 0.4 to compare the running time performance of our method with the state of the art methods.

Synthetic data set: We have generated two artificial signaling networks, which simulate signaling network topologies, with 24 and 72 genes named as Synthetic24 and Synthetic72, respectively (as described in Table 3.1). Synthetic24 network has two critical genes and Synthetic72 has four critical genes. We also randomly generated RNAi scores for each of the critical genes as described below in the following section.

For both Synthetic24 and Synthetic72 networks, we generated 25 networks by using the "ADDRIP" method as described above.

To compare the performances of TopologyRNAiDivide and RNAiDivide on a synthetic dataset, we also use Synthetic24 network by generating 20 reference networks using the edge shuffling method (Milo et al., 2003).

RNAi score generation and Accuracy Measures: Since the synthetic and semi-synthetic data we use in this study do not have RNAi data, we generate synthetic RNAi scores for these data. We follow the same procedure for the generation of synthetic RNAi scores as given in Chapter 2. Precision and recall values are the accuracy measures chosen for the evaluation of the results, which are also defined in Chapter 2.

Table 3.1 Datasets used in the experiments

Dataset	Description	|V|	|E|
VEGF_PGI2	VEGF sub pathway containing the PGI2 gene (human)	14	15
p53_apoptosis	p53 apoptotic response sub pathway (human)	20	24
Neutrotrophin	neurotrophin sub pathway signaling with Bcl-2 (human)	24	31
FceRI	Fc epsilon RI subpathway containing the genes JNK and p38 (human)	13	16
HSA2	The HSA2 network from Sasan et al.	388	615
Synthetic 72	72 gene synthetic signaling network	72	102
Synthetic 24	24 gene synthetic signaling network	24	35

3.4 Results

We applied our methods on the given data sets and compared them with the state of the art methods. We first compare the results with SiNeC on ERBB network, then we compare two of our methods RNAiDivide and TopologyRNAiDivide with each other on synthetic and semi-synthetic networks, and finally compare SiNeC with TopologyRNAiDivide, which yields better results than RNAiDivide, on both synthetic and semi-synthetic datasets.

3.4.1 Comparison on ERBB Network with the state-of-art methods

It has been shown in Hashemikhabir et al. (2012) that SiNeC outperforms other state of the art methods, (Ruths et al., 2007; Singh, 2011; Kaderali et al., 2009) in the real data set, ERBB network (Hashemikhabir et al., 2012); therefore, in this section, we report a comparison of our method with SiNeC on the ERBB network only.

ERBB Network:
The ERBB network is a network in which the ratio of the number of components to the number of critical genes is small. Therefore, we use VerticalDivide to reconstruct the network for this dataset. The RNAi scores were taken from the Sahin et al.2009, and the critical genes are identified as ERBB1, ERBB1_2, IGF1R, ER-alpha, c-MYC, CyclinD1, CyclinE1, CDK4, and CDK6 with respect to their RNAi data. We compare VerticalDivide with SiNeC on the ERBB receptor-regulated G1/S transition network. We use an undirected literature curated reference network (Figure 3.4a) to reconstruct the signaling network.

Comparison with SiNeC: For both methods, the reference network and critical genes are taken the same. The constructed networks are given in Figure 3.4b and Figure

3.4c. In order to make a comparison between them, the precision and recall values are calculated for the constructed networks. While SiNeC has a precision value of 0.28, we obtain a precision of 0.39 with our method. Also, recall value of our method is 0.55, while it is 0.52 for SiNeC. Both recall and precision values are greater for VerticalDivide, which indicates that VerticalDivide is able to produce biologically more accurate results on real data sets.

3.4.2 Comparison between RNAiDivide and TopologyRNAiDivide

In this section, we analyze the results of RNAiDivide and TopologyRNAiDivide on the datasets VEGF_PGI2 and Synthetic24. The results are shown in Figure 3.5 - Figure 3.8.

For the VEGF_PGI2 network, we see that TopologyRNAiDivide is 10% more accurate on the average and produces less number of false positive interactions compared to RNAiDivide as the noise rate increases. Moreover, while RNAiDivide reconstructs the whole network by dividing the reference network into three sub-networks, TopologyRNAiDivide reconstructs the signaling network with two sub-networks. Therefore, TopologyRNAiDivide reduces the probability of deletion of the true edges due to the division of the network, which is a drawback of the division procedure.

Depending on the increase in the number of genes in a network, RNAiDivide performs poorly compared to TopologyRNAiDivide in both running time and in finding accurate solutions. When we analyze the results for the Synthetic24 gene network, we see that TopologyRNAiDivide increases the recall values by 20% on average.

In addition, an average of 25% increase in the precision values is observed. Due to the much better handling of false positive and false negative edges compared to RNAiDivide, TopologyRNAiDivide gives more accurate results.
In conclusion, TopologyRNAiDivide copes with the problems of the division of large networks into smaller sub-networks much better than RNAiDivide and produces more robust solutions. We observe that TopologyRNAiDivide is more powerful in determining the true positive interactions while eliminating the false positive edges.

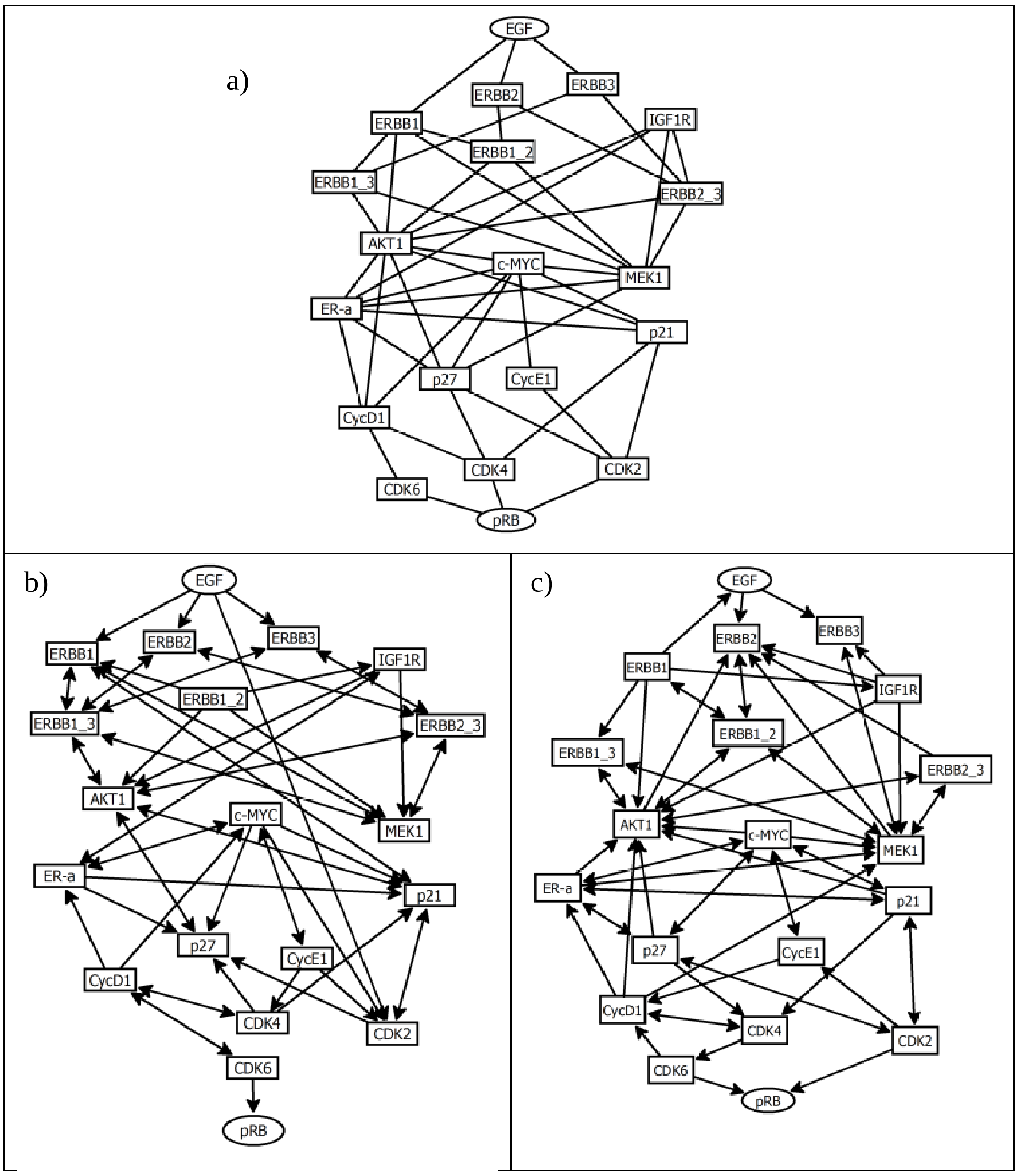


Figure 3.4 Schematic of ERBB network. (a) Reference network (Sahin et al., 2009), (b) constructed network by VerticalDivide, (c) SiNeC (Hashemikhabir et al., 2012)

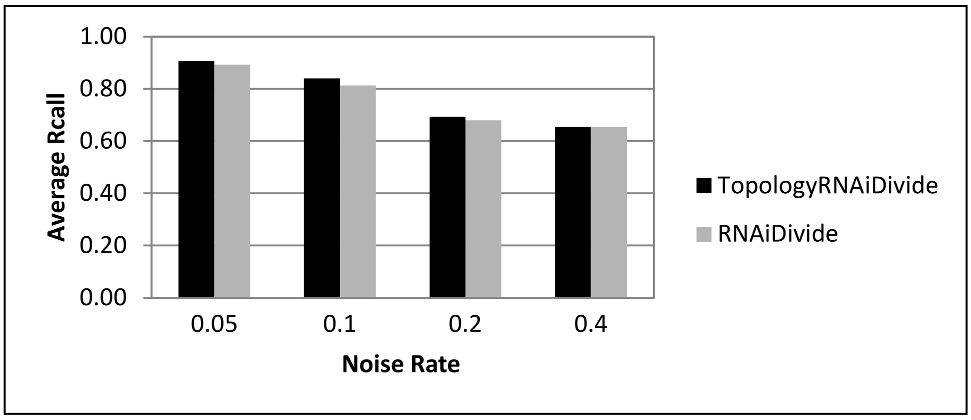


Figure 3.5 Average recall values for VEGF_PGI2 data set.

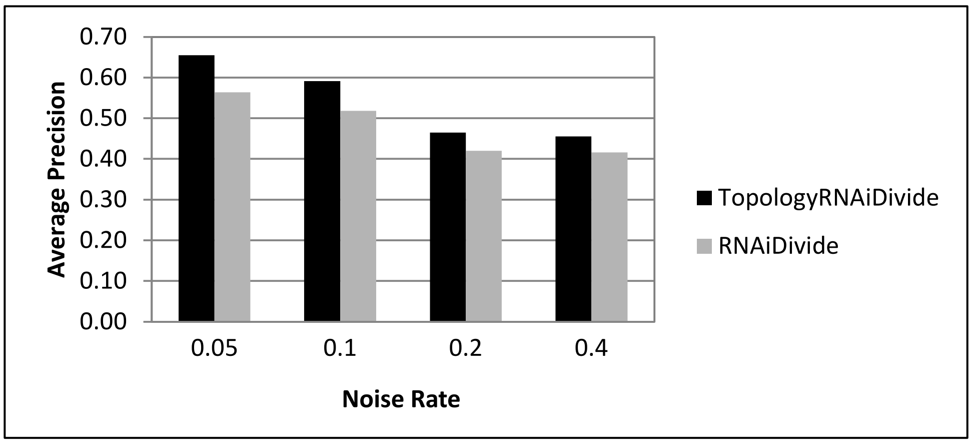


Figure 3.6 Average precision values for VEGF_PGI2 data set.

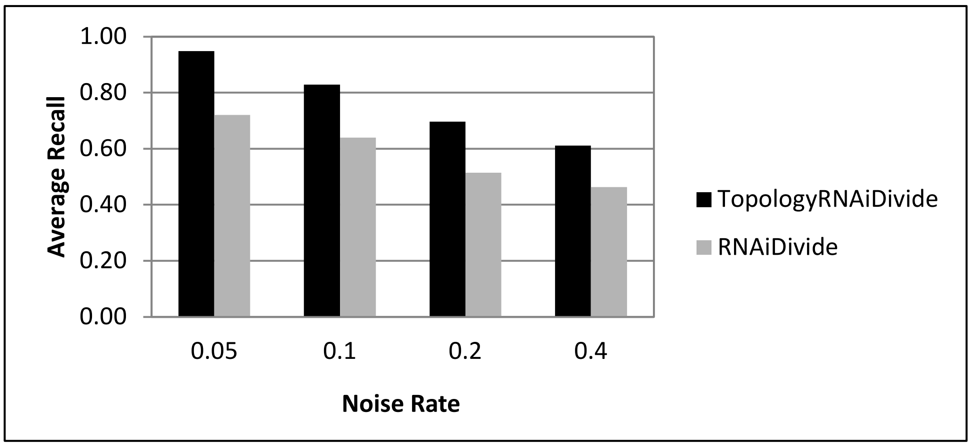


Figure 3.7 Average recall values for Synthetic24 data set.

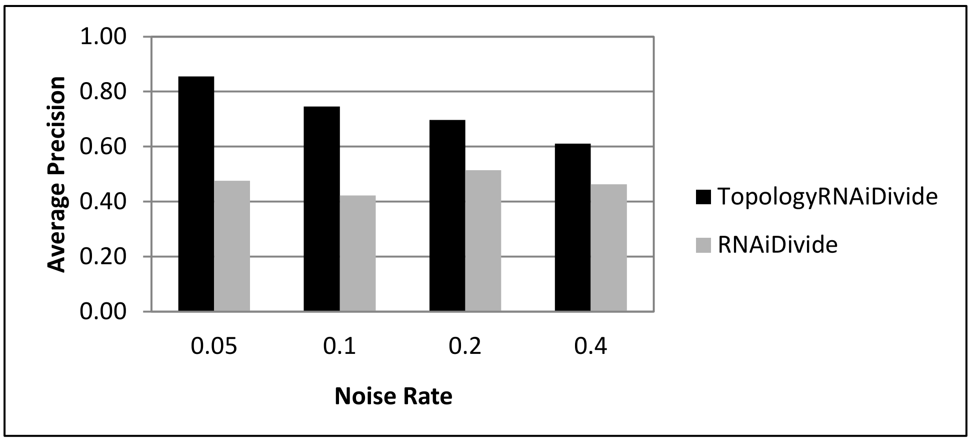


Figure 3.8 Average precision values for Synthetic24 data set.

3.4.3 Comparison between TopologyRNAiDivide and SiNeC on large networks

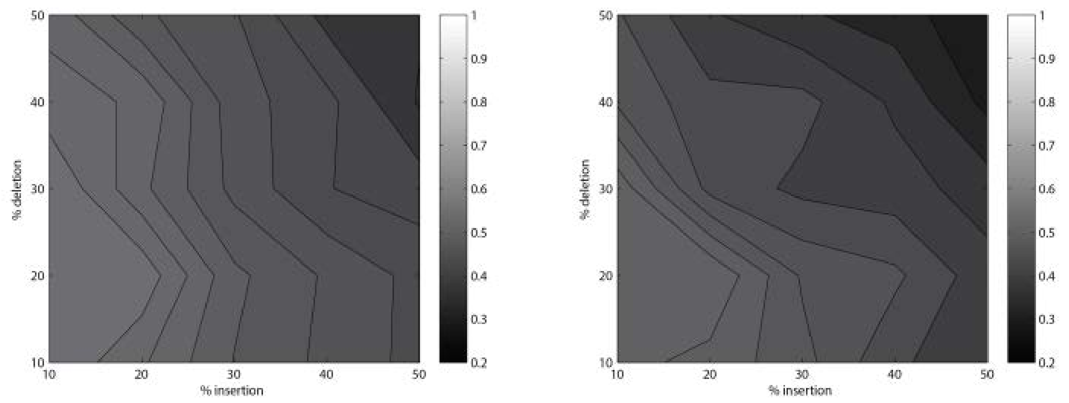
In the previous section, we have shown that TopologyRNAiDivide is able to construct signaling networks more accurately compared to RNAiDivide on large size networks. In this section, we compare TopologyRNAiDivide with SiNeC.

To assess the performance of TopologyRNAiDivide on large networks, we compare the final topologies constructed by TopologyRNAiDivide and SiNeC by means of recall and precision on six different data sets. Four of these data sets are semi-synthetic data sets, VEGF-PGI2, p53-apoptosis, neutrotrophin, and FceRI. The other two data sets we used in the comparison experiments are the Synthetic24 and Synthetic72 networks. p53-apoptosis and neutrotrophin networks have independent multiple parallel pathways structurally. The other four networks have bow-tie topology. We ran both SiNeC and TopologyRNAiDivide on these networks and calculated the corresponding recall and precision values. Contour plots which are described in Section 2.7.1 are used to represent these values for comparison.

If we compare the precision and recall values for p53-apoptosis and neutrotrophin networks, the results of which are given in Figure 3.9 - Figure 3.12, it is seen that the robustness of the two methods are approximately same on the dense networks. However, while the reference networks become sparse, TopologyRNAiDivide is able to construct more accurate networks compared to SiNeC. This is because SiNeC cannot handle some non-critical genes if there is no interaction related with this gene in the reference network. Although TopologyRNAiDivide has better accuracy on sparse networks, it cannot handle false positive edges as much as SiNeC can do, due to the structural nature of the network. Since these two networks have independent multiple parallel pathways, TopologyRNAiDivide adds some false positive edges on the final network to satisfy the constraints because of the division method. Moreover, higher precision values obtained by SiNeC are also because of the different structures of the constructed network topologies. While TopologyRNAiDivide is forced to construct a connected network topology, the topology of the constructed network by SiNeC may have genes which do not transduce the signal. Therefore, if all the genes are required to be involved in signal transduction, TopologyRNAiDivide should be used; if not, SiNeC may be preferred.

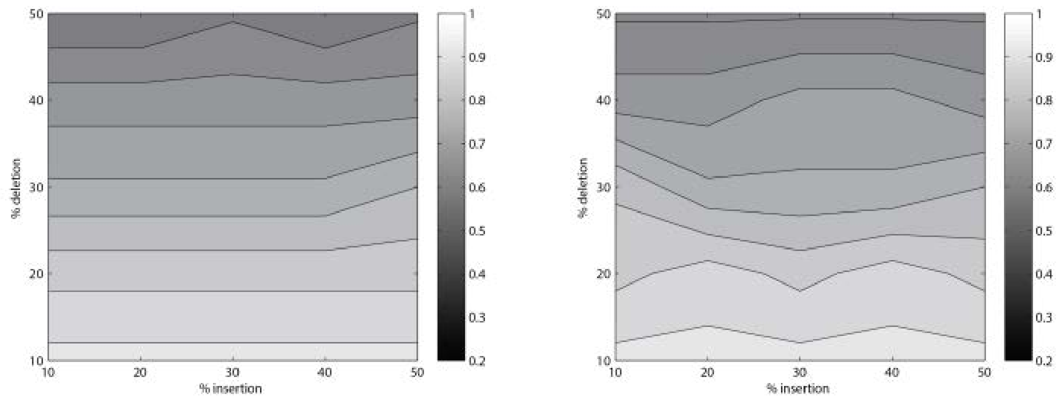
Next we consider bow-tie topologies, which are theVEGF-PGI2, FceRI, Synthetic24, and Synthetic72 networks. The comparison between TopologyRNAiDivide and SiNeC is depicted in Figure 3.13 - Figure 3.20. The results reveal that TopologyRNAiDivide is more robust and results in less number of false positive and false negative edges compared to SiNeC in these four networks with varying number of genes. For such kind of network structures, TopologyRNAiDivide achieves better accuracy on both sparse and dense reference networks compared to SiNeC. TopologyRNAiDivide constructs more accurate results which are closer to the true network topology and always gives a connected network as a final output. The main difference between the final outputs is that while TopologyRNAiDivide is forced to find a connected network, with all the genes contributing to the flow, SINEC finds a topology consistent with the RNAi scores but does not impose a connectivity constraint; hence, cannot handle non-critical genes which are not connected to the signaling network in the reference network.

In general, as the amount of deletion increases, TopologyRNAiDivide gives better results than SiNeC (the shaded area colors are lighter for TopologyRNAiDivide). This means that even if we have very little biological information at the beginning, i.e., high amount of deletions, our proposed approach is able to construct a network which has a closer topology to the actual network than the one SiNeC constructs.



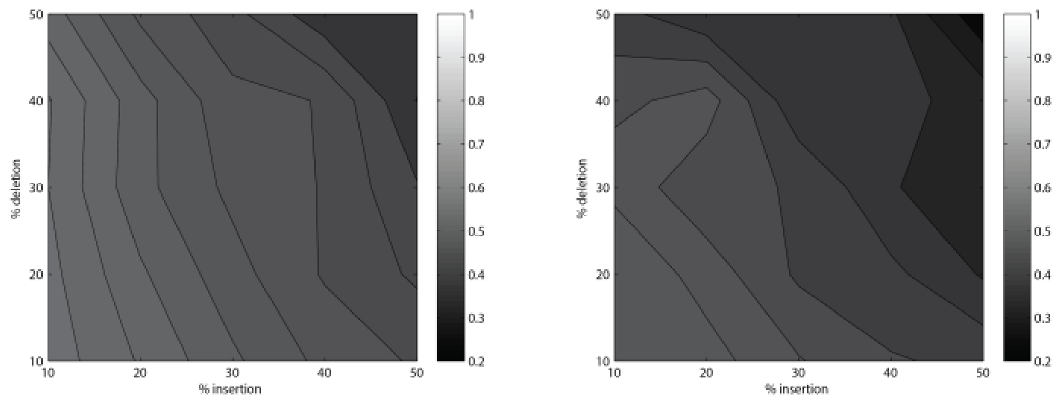
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.9 Contour plot for precision values for the p53apoptosis network.



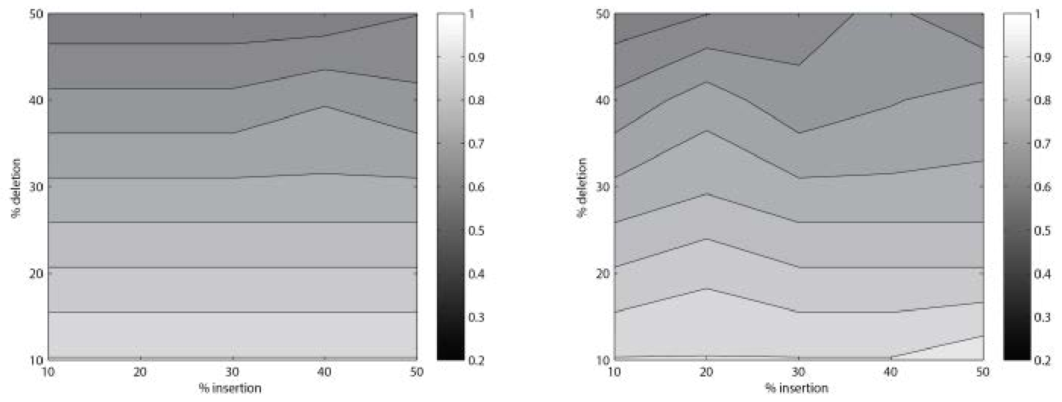
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.10 Contour plot for recall values for the p53apoptosis network.



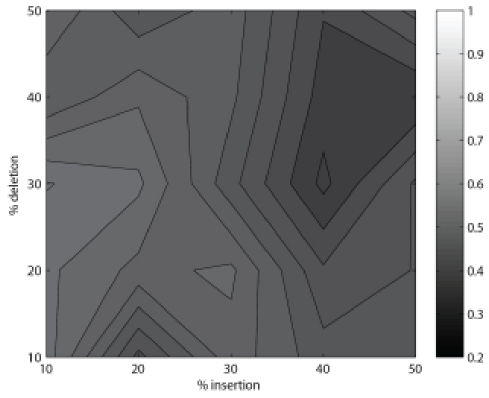
(a) SiNeC, (b) TopologyRNAiDivide

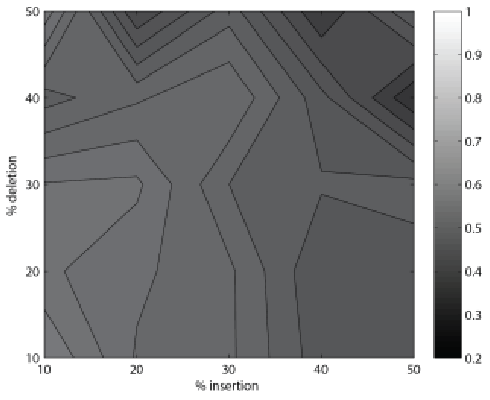
Figure 3.11 Contour plot for precision values for the neutroprophin network.



(a) SiNeC, (b) TopologyRNAiDivide

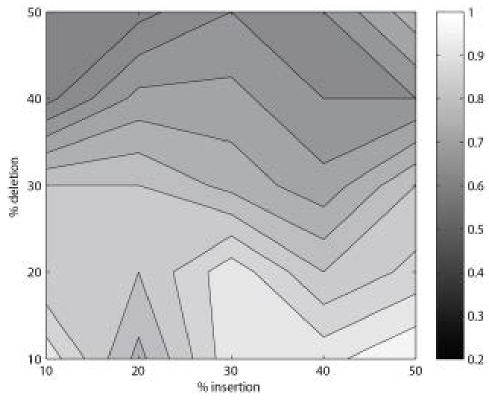
Figure 3.12 Contour plot for recall values for the neutroprophin network.

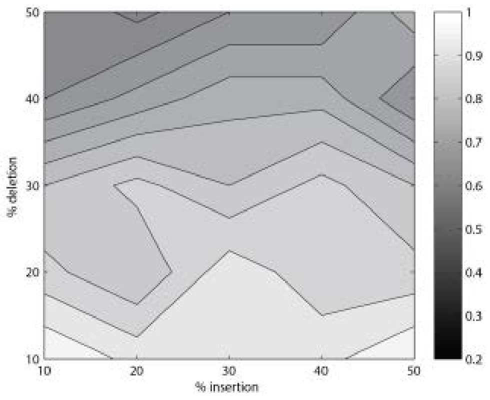




(a) SiNeC, (b) TopologyRNAiDivide

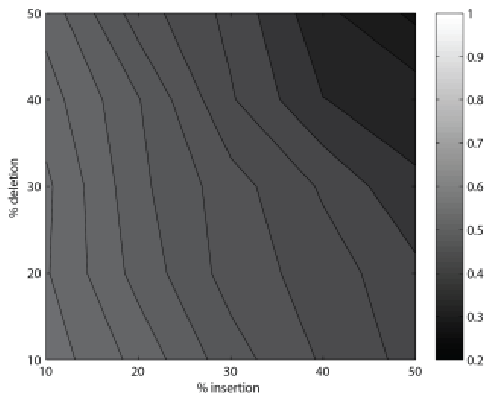
Figure 3.13 Contour plot for precision values for the VEGF_PGI2 network.

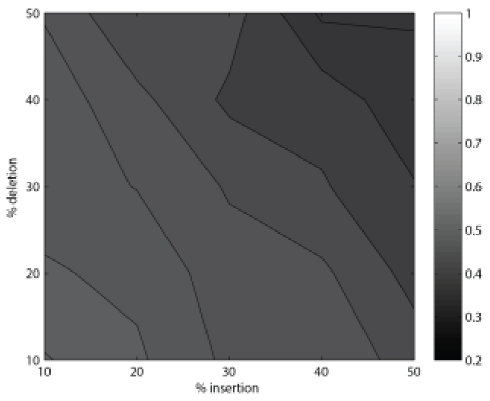




(a) SiNeC, (b) TopologyRNAiDivide

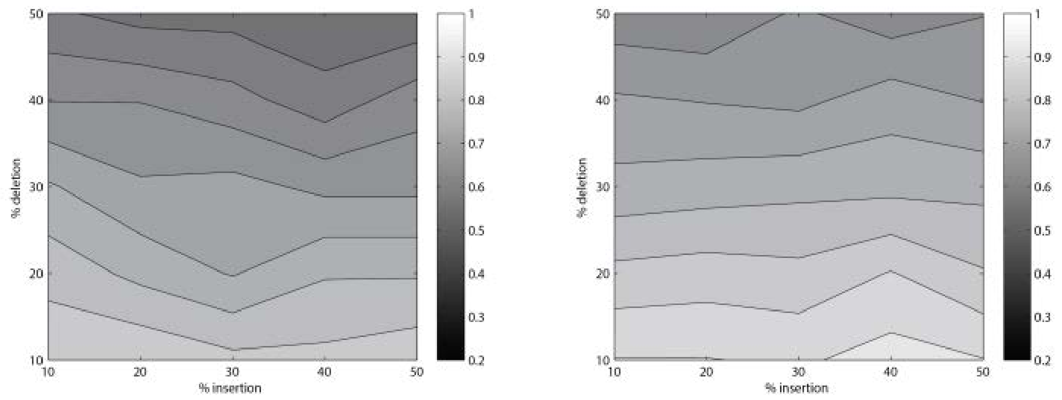
Figure 3.14 Contour plot for recall values for the VEGF_PGI2 network.





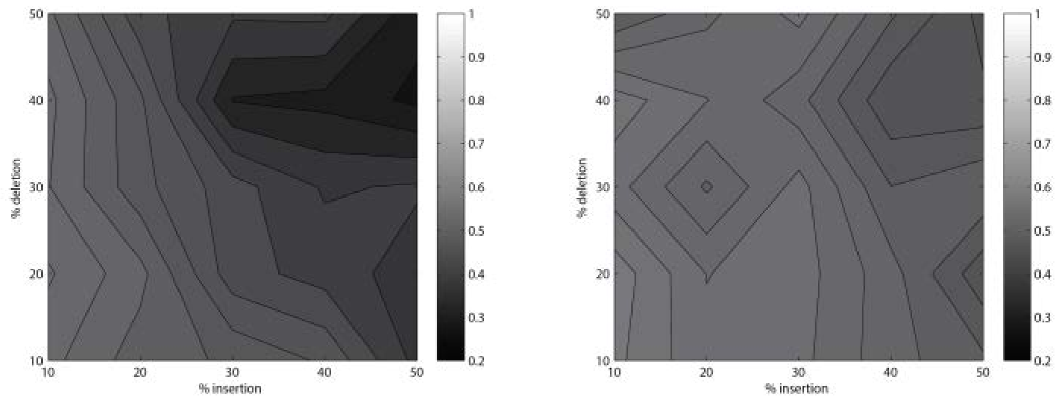
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.15 Contour plot for precision values for the Synthetic 72 network.



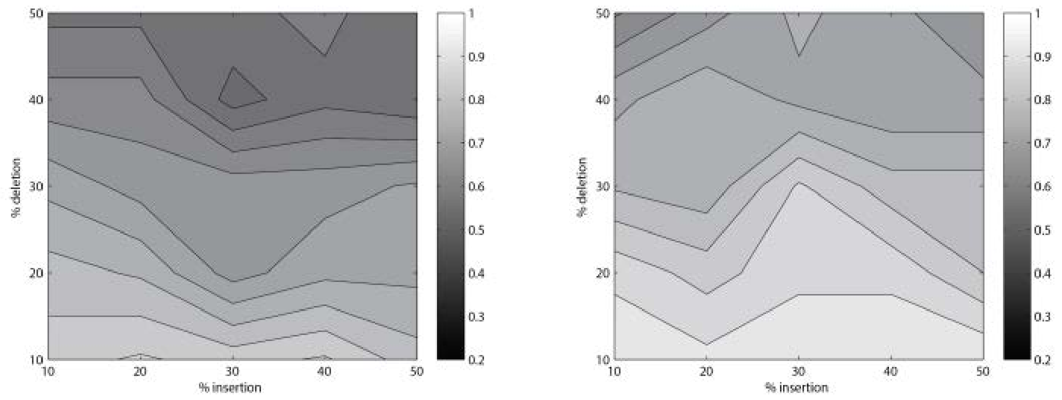
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.16 Contour plot for recall values for the Synthetic 72 network.



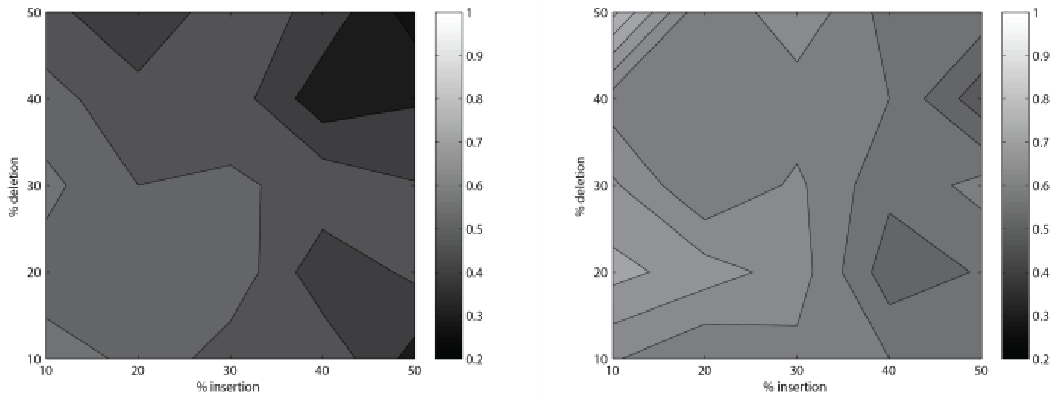
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.17 Contour plot for precision values for the Synthetic 24 network.



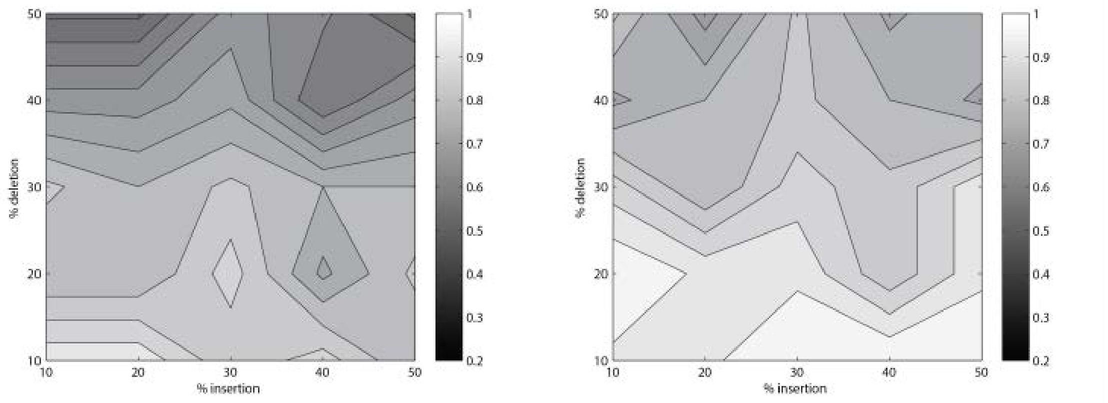
(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.18 Contour plot for recall values for the Synthetic 24 network.



(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.19 Contour plot for precision values for the FceRI network.



(a) SiNeC, (b) TopologyRNAiDivide

Figure 3.20 Contour plot for recall values for the FceRI network.

3.5 Running time performance comparison with SiNeC

Hashemikhabir et al. evaluated SiNeC running time performance in their study. While SiNeC can find solutions for large networks up to 200 genes in seconds, it fails to find a solution for dense networks in 1 hour. For example, SiNeC cannot find a solution for the HSA2 network which has 388 genes and 615 interactions in the true signaling network. To remedy this, they propose another method named S-SiNeC which can provide a solution to the HSA2 network in seconds. However, S-SiNeC fails to satisfy RNAi constraints for some networks; hence, cannot guarantee a correct solution for every input. On the other hand, the methods we propose can find a correct solution for any input. Therefore, in terms of robustness, we are more similar to SiNeC. The success rate of S-SiNeC is reported to be high when the distance between reference network and the actual network is small. When the distance gets larger, it fails to satisfy some of the constraints but the number of such constraints is found to be small. S-SiNeC solves approximately 60 percent of the reference networks at the highest noise rate (40 percent noise rate). We think that in real settings, reference networks can be noisy and S-SiNeC will not be able to produce a solution for a considerable percentage of the instances. Therefore, in this chapter we compare our results with SiNeC, instead of S-SiNeC.

We apply our method on all 200 mutated reference networks for the HSA2 network described in the data sets section and measure the running time. We divide each network into equal size sub-networks and calculate the total solution time. Each sub-network is solved on a quad core 2.66 GHz CPU (Intel XEON E5430) 16 GB memory HPC node, sequentially. The total solution time for a network is calculated by summing up solution times given by CPLEX for each sub-network. The results show that the average of the running times for the 200 networks is 4.6 minutes with a minimum of 0.47 minutes and a maximum of 18.9 minutes. While SiNeC cannot find a solution to these large networks in 1 hour, we can reconstruct the networks in minutes, which shows that our approach can scale for large networks very well. This is a significant benefit of our proposed divide and conquer approach over the state of the art methods. Note that, although we report sequential solution times, it is also possible to solve each divided network in parallel with parallel architectures independently and construct networks in even shorter times.

3.6 Conclusion

In the previous part of this study, we formulate the network reconstruction problem as a linear optimization problem in which we construct a network satisfying the given RNAi data with minimum edit operations applied on a given reference PPI network. This ILP formulation can construct networks with high accuracy, for small networks of size up to 10 genes with single receptors and single reporters. In order to be able to construct large networks, we develop a divide and conquer based approach in this part of the study. The main idea is dividing the whole problem into solvable small problems, then solve each problem separately, and finally bring all solutions are together to find the final topology. We develop two different approaches depend on the given network topology. Comparison with the state of the art methods shows that the proposed methods scale better and give more accurate results. While the state of the art methods simply do not include a non-critical gene in the final constructed network if the reference network does not show a connection for this gene; we always include all the genes in the final constructed network. Therefore, if the input list of genes is all known to be involved in the signaling network, our method is more likely to construct biologically correct networks. Also, we are able to find solutions to both sparse and dense networks in minutes, while state-of-the-art methods fail to construct networks in 1 hour if the networks are dense and large networks.

In conclusion, since studying and understanding biological network topologies are essential for interpreting biological systems for drug target identification, the methods proposed in this study are valuable and convenient to construct novel network topologies from screening experiments.

Table 3.2 Comparison of proposed methods with SiNeC

		Network #:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
MAPK fruit fly	Precision	oILP	0.59	**0.63**	0.47	0.44	0.50	0.45	0.53	0.39	0.39	0.53	0.42	0.43	0.45	0.39	0.53	0.45	0.44	0.45	0.38	0.35	0.53	0.39	0.38	0.35	0.47
		SINEC	**0.47**	0.39	0.41	0.44	0.45	0.38	0.35	0.39	0.39	0.44	0.36	0.32	0.38	0.28	0.43	0.42	0.40	0.27	0.29	0.21	0.39	0.24	0.26	0.23	0.25
	Recall	oILP	**1.00**	**1.00**	0.80	0.70	0.70	0.90	**1.00**	0.70	0.70	0.80	0.80	0.90	0.90	0.70	0.80	**1.00**	0.80	0.90	0.60	0.60	**1.00**	0.90	0.80	0.70	0.90
		SINEC	0.90	0.70	0.70	0.70	0.50	0.80	0.70	0.70	0.70	0.70	0.80	0.70	0.80	0.50	0.67	**1.00**	0.80	0.60	0.50	0.40	0.90	0.60	0.60	0.50	0.50
VEGF_PGI2	Precision	TopologyRNAiDivide	**0.50**	0.46	0.46	0.39	0.47	0.44	0.44	0.46	0.42	0.32	0.42	0.42	0.40	0.44	0.39	0.37	0.36	0.38	0.37	0.30	0.37	0.37	0.39	0.28	0.38
		SINEC	0.45	0.44	**0.46**	0.38	0.42	0.33	0.41	0.44	0.42	0.36	0.41	0.42	0.37	0.38	0.39	0.37	0.34	0.29	0.31	0.32	0.38	0.36	0.36	0.30	0.37
	Recall	TopologyRNAiDivide	**1.00**	0.87	0.80	0.60	0.60	0.93	0.80	0.87	0.67	0.53	0.93	0.93	0.80	0.73	0.60	0.93	0.87	0.87	0.73	0.60	**1.00**	0.87	0.80	0.60	0.73
		SINEC	0.93	0.80	0.80	0.53	0.53	0.73	0.80	0.80	0.67	0.53	0.93	0.93	0.73	0.67	0.60	0.93	0.80	0.67	0.60	0.60	**1.00**	0.87	0.80	0.60	0.73
FceRI	Precision	TopologyRNAiDivide	0.56	0.67	0.61	0.52	**0.76**	0.50	0.63	0.50	0.50	0.50	0.53	0.58	0.56	0.52	0.59	0.50	0.42	0.46	0.50	0.48	0.47	0.45	0.52	0.37	0.48
		SINEC	**0.47**	0.43	0.46	0.42	0.37	0.44	0.40	0.40	0.37	0.30	0.37	0.44	0.41	0.38	0.38	0.39	0.31	0.39	0.27	0.33	0.28	0.36	0.35	0.29	0.23
	Recall	TopologyRNAiDivide	0.94	**1.00**	0.88	0.69	0.81	0.94	0.94	0.75	0.75	0.63	**1.00**	0.94	0.88	0.81	0.81	**1.00**	0.81	0.81	0.75	0.69	**1.00**	0.94	0.94	0.69	0.75
		SINEC	**0.94**	0.75	0.81	0.63	0.44	**0.94**	0.75	0.75	0.63	0.44	0.81	0.88	0.81	0.69	0.63	0.88	0.69	0.75	0.50	0.56	0.75	0.81	0.75	0.63	0.44
neutrotrophin	Precision	TopologyRNAiDivide	**0.38**	**0.38**	0.35	0.33	0.31	0.37	0.35	0.32	0.35	0.28	0.34	0.32	0.32	0.28	0.28	0.32	0.30	0.28	0.29	0.28	0.32	0.28	0.26	0.27	0.22
		SINEC	**0.45**	**0.45**	0.44	0.44	0.41	0.42	0.40	0.39	0.39	0.36	0.38	0.37	0.35	0.35	0.33	0.35	0.34	0.34	0.34	0.29	0.34	0.32	0.30	0.28	0.29
	Recall	TopologyRNAiDivide	0.90	0.81	0.71	0.61	0.52	0.90	0.84	0.74	0.68	0.55	0.90	0.81	0.71	0.61	0.58	0.90	0.81	0.71	0.65	0.61	**0.94**	0.81	0.71	0.68	0.55
		SINEC	**0.90**	0.81	0.71	0.61	0.52	**0.90**	0.81	0.71	0.61	0.52	**0.90**	0.81	0.71	0.61	0.52	**0.90**	0.81	0.71	0.65	0.52	**0.90**	0.81	0.71	0.61	0.55
p53apoptosis	Precision	TopologyRNAiDivide	0.41	0.40	0.41	0.36	0.34	0.39	**0.42**	0.33	0.33	0.30	0.37	0.36	0.31	0.33	0.28	0.32	0.34	0.31	0.30	0.27	0.31	0.31	0.29	0.25	0.24
		SINEC	**0.46**	0.45	**0.46**	0.43	0.41	0.42	0.45	0.40	0.42	0.36	0.38	0.38	0.35	0.35	0.34	0.35	0.36	0.32	0.33	0.29	0.33	0.33	0.31	0.28	0.28
	Recall	TopologyRNAiDivide	**0.92**	0.83	0.79	0.63	0.54	**0.92**	0.88	0.71	0.63	0.54	**0.92**	0.83	0.71	0.67	0.54	**0.92**	0.88	0.71	0.67	0.54	**0.92**	0.83	0.75	0.63	0.54
		SINEC	**0.92**	0.83	0.71	0.63	0.50	**0.92**	0.83	0.71	0.63	0.50	**0.92**	0.83	0.71	0.63	0.54	**0.92**	0.83	0.71	0.63	0.50	**0.92**	0.83	0.75	0.63	0.54
Synthetic 24	Precision	TopologyRNAiDivide	**0.49**	**0.49**	0.45	0.47	0.38	0.44	0.44	0.40	0.44	0.42	0.45	0.45	0.45	0.41	0.45	0.41	0.41	0.38	0.34	0.38	0.38	0.35	0.40	0.34	0.33
		SINEC	0.43	**0.44**	0.42	0.43	0.40	0.39	0.41	0.37	0.36	0.33	0.38	0.33	0.33	0.26	0.30	0.36	0.31	0.30	0.25	0.31	0.29	0.29	0.30	0.24	0.25
	Recall	TopologyRNAiDivide	**0.94**	0.89	0.74	0.69	0.54	0.91	0.83	0.71	0.74	0.63	**0.94**	0.89	0.86	0.69	0.71	**0.94**	0.89	0.77	0.66	0.66	**0.94**	0.80	0.77	0.66	0.63
		SINEC	0.83	0.77	0.69	0.57	0.49	**0.86**	0.74	0.63	0.57	0.49	0.83	0.63	0.63	0.43	0.49	**0.86**	0.69	0.63	0.49	0.51	0.77	0.69	0.66	0.46	0.46
Synthetic 72	Precision	TopologyRNAiDivide	**0.40**	0.38	0.37	0.37	0.36	0.38	0.38	0.36	0.35	0.32	0.36	0.35	0.34	0.32	0.33	0.36	0.34	0.32	0.31	0.28	0.33	0.31	0.30	0.29	0.28
		SINEC	**0.45**	0.44	0.44	0.43	0.41	0.41	0.39	0.39	0.38	0.35	0.37	0.35	0.35	0.32	0.31	0.34	0.33	0.32	0.28	0.27	0.31	0.31	0.28	0.27	0.24
	Recall	TopologyRNAiDivide	0.90	0.81	0.72	0.66	0.57	0.90	0.82	0.73	0.65	0.56	0.89	0.81	0.74	0.64	0.61	**0.92**	0.85	0.74	0.68	0.57	0.90	0.80	0.74	0.65	0.60
		SINEC	0.83	0.78	0.71	0.60	0.51	**0.84**	0.74	0.66	0.60	0.48	0.81	0.70	0.67	0.57	0.48	0.81	0.75	0.64	0.52	0.46	0.83	0.75	0.64	0.58	0.46

The network numbers are given according to the distribution of noise rates as follows: The first 5 networks have constant 10 percent addition of edges but varying percentages of deletions (ascending from 10 percent to 50 percent as the network number grows). Similarly, next 5 networks have constant 20 percent addition but varying percentages of deletions as in the first 5 networks. And the rest of the networks in groups of five have similar distribution of additions and deletions. Boldface numbers are the maximum values at each row.

CHAPTER 4

CONSTRUCTION OF MULTI-SOURCE/MULTI-SINK NETWORKS

The approach that we have developed for reconstruction of biological networks using RNA interference data is convenient with networks which have a single receptor gene and a single reporter gene. However, in real biological networks there are more than one such genes/proteins. Our previous approaches which are given in Chapter 2 and Chapter 3 are not suitable for networks with multiple sources and sinks. In order to handle such networks, we use gene expression data and develop another ILP based graph theoretical method. In the method we developed, we use time-series data which consists of gene expression measurements at different time periods. Such data allows us to track the changes happening in the network, and thus to find the interactions between genes.

The simulations carried out on 10 and 20 gene synthetic networks show that the proposed methods perform very well even when significant amount of noise is present in the data. Additional biological information, such as perturbation experiments, results in more accurate networks. Using mice agouti signaling network, the method is shown to perform well on real biological networks and identify several interactions.

4.1 Introduction

For network inference, several models are developed which depend on the available data (Bower and Bolouri, 2001; de Jong, 2002). The type and amount of data, prior knowledge about the network, experimental and computational resources are important in the development of such models. In the last two decays, DNA microarray technology has emerged and widely used by researchers. The expression levels of the genes in a living organism can be measured in one array with this technology.

The functionality of a gene can be determined by measuring the amount of mRNA production during DNA transcription, in a cell. This can be done by using microarrays, which can detect the mRNA levels of thousands of genes in a single experiment. Microarrays, or gene-chips, consist of DNA spots attached to a surface consisting of specific DNA sequences. When a gene is activated, it produces mRNA which is complementary. Therefore, it then binds to the original portion of the DNA strand from which it was copied. To determine the activated or de-activated genes, the mRNA molecules are collected from a cell in consideration. Then, these mRNA molecules are labeled by using an enzyme which generates the so-called complementary DNA (cDNA) to the mRNA. Fluorescent agents are attached to the cDNA during this process. Then, these labeled cDNAs are placed onto the microarray, and they bind to their complementary DNAs on the microarray with fluorescence. Measuring the intensity of fluorescence at each spot on the microarray using a scanner, activity level of a gene can be obtained.

Microarray data is available in terms of a gene expression matrix, which is a table of rows representing gene names and columns representing the experiment number. The columns can be either time points or treatments depending on the experiment type. The expression profile for a gene is accessed from the corresponding row on the matrix.

Depending on the type of experiment, microarray data can be either time-series or perturbation data. Time series data gives information about the time-change of gene expression levels. Important information on time-varying cellular processes can be obtained. Similar to RNA interference experiments, perturbation experiments gives information about the effects of a change or treatment on the cellular activities. Following a disruption, the response of pathways can be determined. These experiments are very suitable to be used in determining causal relationships between genes (Wagner, 2001; Milo et al., 2002).

4.2 Models used for network inference

There are several regulatory network inference models available in the literature. In Boolean networks, the gene expression levels are discretized in binary form and the Boolean functions of the genes in the network are found to explain these discretized values of gene expression (Liang et al., 1998; Kwon and Cho, 2007; Bornholdt, 2008). These models are usually easy to implement but not suitable for noisy data such as microarray data. Bayesian networks are directed acyclic graphs that represent the data best making use of a scoring function (Lee and Yang, 2008). The expression levels of the genes in the network are represented as random variables determined by a probability distribution function. Dynamic Bayesian network model is simply an extended version of the Bayesian network model that takes into account dynamic effects in the network (Friedman et al., 2000; Zou and Conzen, 2005; Zhang et al. 2007; Li et al., 2011). The changes in time series gene expression data is assumed to occur in discrete intervals. In relevance networks, interactions are modeled by an undirected graph with weighted edges depending on a similarity or statistical dependence (Schafer and Strimmer, 2005; Margolin et al, 2006). Due to the relative simplicity of these models, they require less computational effort. Differential and difference equation models use system of differential equations to model the time change of gene regulation in the network (de Jong, 2002; Gebert et al., 2007; Gardner et al., 2003; Bansal et al., 2006). The computational cost of such quantitative models is usually very high.
In this study, we develop a graph theoretical model, which can describe causal

regulatory relations in the network, for construction of signaling and gene-regulatory networks. We use gene expression data and develop an ILP based graph theoretical method. In the proposed method, we use time-series data which consist of gene expression measurements at different time points. Such data allow us to track the changes happening in the network, and thus to find the interactions between genes.

4.3 Proposed approach

For reconstruction of signaling and regulatory networks, we first developed a model that utilizes only time series gene expression data. Then, we improved the model by using perturbation experiments as additional data. Since these data are noisy, we proposed two methods to handle the conflicts that may be imposed by such data.

4.3.1 Model Based on Gene-Expression Data Only

Consider a given directed graph *G(V,E)* where *V* represents the node set and *E* represents the edge set, with several source nodes s_i, and sink nodes t_j. This graph may be a literature curated gene regulatory network or taken from any of the gene regulatory network database, where each node represents a gene and assume that the gene expression levels at different time points are available from microarray experiments. The aim is to reconstruct a new network from the given network satisfying microarray data by making minimum changes on the given network. The approach would be to formulate this problem as a linear optimization problem, which will provide a network satisfying the microarray data with a minimum change applied on the given network.

Since the method we previously developed (Eren Ozsoy and Can, 2013) uses RNAi data and cannot handle networks with multiple sources and sinks, another approach is required to find real biological networks. Our goal is to develop an approach for reconstruction of gene regulatory networks using gene expression data. The approach has two main phases: pre-processing phase and solution of ILP formulation. Since the microarray data may contain several thousands of genes, only some of them must be taken into consideration to obtain valuable information about the signaling pathways.

Preprocessing phase: In the pre-processing phase, the gene expression data is processed to identify the active genes at each time point. The states of the genes in the network, either active or inactive, are determined by using a threshold that depends on the dataset for the expression levels or fold changes. The state variables for active genes are assigned as 1, while remaining are assigned as 0. Finally, we obtain a matrix of ones and zeros for each gene at all time points. An example matrix is given in Figure 4.1.

Solution phase: In this phase, the objective function for the ILP formulation and the corresponding constraints are derived.

Objective function:
We assume that the reference network is given (it can be taken from literature or any known pathway database) and it has to be modified to satisfy the constraints derived from the new biological data (microarray data). The network will be reconstructed by applying minimum edit operations, i.e. insertions/deletions on the network. The proposed method works even when there is no reference network available. For such case, minimum number of edge additions is sought. Let x_{ij} be the binary variable representing the presence of the edge between nodes *i* and *j* which is from node *i* to node *j* in the given network. If the edge is present, then the value of x_{ij} is 1, otherwise it is 0. Similarly, let w_{ij} represent the edges in the network that is to satisfy the given microarray data. The microarray data consists of the information whether a gene is active or inactive. We name these binary variables "the state variables". The goal is to reconstruct the given network with respect to the microarray data by minimizing the changes that have to be applied to the initial reference network. The objective function for this linear problem would be the sum of the absolute values of the differences between w_{ij} and x_{ij}, i.e. $|x_{ij} - w_{ij}|$. If the edge is present both before and after the optimization, then the difference becomes 0, which means no change is made on the corresponding edge in the network. However if the difference is 1, it means that the edge is either taken out from the network or it is inserted into the network. Therefore, minimizing the sum of these differences results in a network that is obtained by making minimum number of changes on it while satisfying the constraints obtained from the gene expression data.

	Time 1	Time 2	Time 3	.
Gene 1	1	0	0	.
Gene 2	1	0	0	.
Gene 3	0	1	0	.
.	.	.	.	.

Figure 4.1 Example matrix of time series data

The corresponding objective function is given in Eq. (4.1) for the graph *G(V,E)* with *n* nodes, *s* source nodes and *t* sink nodes. The first term in Eq. (1) takes into account that there are no directed edges into the source nodes while the second term takes into account that there are no directed edges into the sink nodes.

$$Minimize \quad \sum_{i=1}^{s} \sum_{j=s+1}^{n-t} \left| x_{ij} - w_{ij} \right| + \sum_{i=s+1}^{n-t} \sum_{\substack{j=s+1 \\ i \neq j}}^{n} \left| x_{ij} - w_{ij} \right| \qquad (4.1)$$

Constraints:
In the solution phase, the matrix of state variables is used in construction of the constraints to be used in our ILP approach. A gene is assumed to be activated when the corresponding state variable becomes 1 the first time. It does not matter what value the state variable is assigned thereafter.

For the construction of constraints, the kinematics of the system is taken into consideration. The gene regulation rate is variable, and therefore the gene is assumed to be activated by all genes which are already activated at all previous time points and also can activate all the nodes at the following time points.

The constraints are based on the following assumptions:

- Sources are the genes which are activated at the first time point and sinks are the genes which are activated at the last time point;
- Each source and sink has to be connected to the network;
- At each time point, the genes may only be activated by the upstream genes, which are active in ALL of the previous time points;
- No direct edges from sources to sinks are allowed;
- No edges between sources or sinks are allowed;
- No self-edges are allowed.

Note that these assumptions do not yield an upstream edge. However, there may be such edges in the reference network. Based on these assumptions, the following mathematical constraints are derived.

1. There should be at least one edge going out of each source to the genes activated at the second time point.
2. There should be at least one edge going into each sink from the genes activated at the last time point.
3. There should be at least one edge going into an intermediate node from the upstream nodes activated at all previous time points. Also, there should be at least one edge going out of an intermediate node to all of the downstream nodes, including the sink nodes.

Note that these constraints are derived only from the time series experiments. It is also possible to add additional constraints if any perturbation experiment is available for the network. The following section explains the integration of perturbation data to this model for a better accuracy.

As an example, consider the following microarray data (Table 4.1) with 4 time-points after the pre-processing phase and assume that there are 10 nodes. According to this table, Gene 1 and 2 are the source nodes, Gene 8, 9 and 10 are the sink nodes.

Table 4.1 Example microarray data

	Time 1	Time 2	Time 3	Time 4
Gene 1	1	0	0	0
Gene 2	1	0	0	0
Gene 3	0	1	0	0
Gene 4	0	1	0	0
Gene 5	0	0	1	0
Gene 6	0	0	1	0
Gene 7	0	0	1	0
Gene 8	0	0	0	1
Gene 9	0	0	0	1
Gene 10	0	0	0	1

1: activated genes 0: inactive genes

1) The constraints stated in the first article can be written mathematically as:

$x_{13} + x_{14} >= 1$ (source 1 = Gene 1)
$x_{23} + x_{24} >= 1$ (source 2 = Gene 2)

The number of constraints here is equal to the number of sinks (n_s).

2) Similarly, the constraints in the second article can be written as:

$x_{58} + x_{68} + x_{78} >= 1$ (sink 1 = Gene 8)
$x_{59} + x_{69} + x_{79} >= 1$ (sink 2 = Gene 9)
$x_{510} + x_{610} + x_{710} >= 1$ (sink 3 = Gene 10)

Here, the number of constraints is equal to the number of sinks (n_t).

3) The constraints in article 3a for genes activated at Time 2 can be written as:

$x_{13} + x_{23} >= 1$ (Gene 3 - in)
$x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{310} >= 1$ (Gene 3 – out)
$x_{14} + x_{24} >= 1$ (Gene 4 - in)
$x_{45} + x_{46} + x_{47} + x_{48} + x_{49} + x_{410} >= 1$ (Gene 4 – out)

Therefore, the number of constraints is 2 times the number of genes activated at Time 2.

Similarly, the constraints for genes activated at Time 3 can be written as:

$x_{15} + x_{25} + x_{35} + x_{45} >= 1$ (Gene 5 - in)
$x_{58} + x_{59} + x_{510} >= 1$ (Gene 5 – out)
$x_{16} + x_{26} + x_{36} + x_{46} >= 1$ (Gene 6 - in)
$x_{68} + x_{69} + x_{610} >= 1$ (Gene 6 – out)
$x_{17} + x_{27} + x_{37} + x_{47} >= 1$ (Gene 7 - in)
$x_{78} + x_{79} + x_{710} >= 1$ (Gene 7 – out)

The number of constraints is 2 times the number of genes activated at Time 3.

4.3.2 Integration of Time Series Data and Perturbation Data

Assume that in addition to the time series data, data from perturbation experiments are available. In perturbation experiments, each node (or only some of them) in the network are knocked-down and the effects on the remaining nodes are observed. A knock-down of one of the nodes in the network potentially changes the expression levels of the rest of the genes. If there is a significant decrease in the expression level, we assume that this gene is repressed. To make this assumption, we first pre-process the perturbation data as we do for the time series data. By determining a threshold for activation–repression, we mark the nodes which are activated or repressed by assigning their state variables as 1 or 0, respectively. An example perturbation data in matrix form is shown in Figure 4.2. As an example, when gene 2 is knocked down (KD), gene 3 is repressed while no change in gene 4 is observed.

	Gene 1	Gene 2	Gene 3	Gene 4	.
Gene 1	KD	1	1	0	.
Gene 2	1	KD	0	1	.
Gene 3	1	1	KD	1	.
Gene 4	1	1	1	KD	.
.	.	.	.	.	.

Figure 4.2 Example matrix of perturbation data

Perturbation data provides additional constraints and even absence or presence of some edges. These additional constraints are stated below:

4) If a sink node *i* is affected by the knockdown of a gene *j* at the time point just before the last one, then there must be a direct edge from node *i* to node *j*.
5) Assume that knockdown of node *i* results in silencing of node *j*. It means that the edges that lead to *j* from all nodes activated in the previous or current time (except node *i*, since it is knocked down) are not present. Therefore, the state variables of these edges are zero.
6) Assume that knockdown of node *i* does not have any effect on node *j* (i.e.

node *j* is still active). Node *j* can only be activated by the nodes which are activated in the current time step or previous time steps. Therefore, there should be at least one edge from these nodes to node *j*. Note that, the intermediate nodes are assumed that they can be affected by the nodes that are activated at the previous or current time step.

Consider now the previous network with 10 nodes and having microarray data for 4 time-points. We will simulate the knockdown data directly from the actual given network by knocking down each gene (or only some of them). The perturbation data will be a binary data after pre-processing phase, stating that if it is zero for a gene, the gene is repressed by the corresponding knockdown; if it is one, then the gene is not affected by this knockdown. Only downstream effects are taken into account. The corresponding perturbation data is given in Table 4.2. Note that there is no need to knockdown the genes that are active at the last time step since they are the sink nodes in the network.

Table 4.2 Perturbation data for the 10-node network.

	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6	Gene7	Gene8	Gene9	Gene10
Gene1	KD	1	0	0	1	0	1	0	0	1
Gene2	1	KD	1	1	1	1	1	1	1	1
Gene3	1	1	KD	1	1	1	1	1	1	1
Gene4	1	1	1	KD	1	0	1	0	0	1
Gene5	1	1	1	1	KD	1	0	1	1	0
Gene6	1	1	1	1	1	KD	1	0	0	1
Gene7	1	1	1	1	1	1	KD	1	1	0

For example; when gene 1 is knocked down, genes 3, 4, 6, 8 and 9 are repressed while no change is observed in the network by the knockdown of node 2.
With the perturbation data, additional constraints as well as absence or presence of some edges can be written considering the 3 statements given above accordingly:

Statement 1) Since knockdown of gene 6 results in repression of genes 8 and 9 (which are the sink nodes of the network), then there must be direct edges from gene 6 to gene 8 and 9. Similarly, there must be a direct edge from gene 7 to gene 10. Therefore, we have

$$x_{68} = x_{69} = x_{710} = 1$$

Statement 2) As an example, consider gene 4. Knockdown of gene 4 results in repression of genes 6, 8 and 9. Therefore, the edges from genes 1, 2, 3 and 5 into genes 6, 8 and 9 are not present. If they were present, then knockdown of gene 4 could not result in silencing of these genes, because they would be activated over these edges. Therefore, we have

$x_{16} = x_{26} = x_{36} = x_{56} = 0$
$x_{18} = x_{28} = x_{38} = x_{58} = 0$
$x_{19} = x_{29} = x_{39} = x_{59} = 0$

Similarly, for other genes, we can write;

for KD of gene 1: $x_{23} = x_{24} = x_{26} = x_{28} = x_{29} = 0$

for KD of gene 2: no results – KD of gene-2 does not have an effect on the other genes.

for KD of gene 3: no results – KD of gene-3 does not have an effect on the other genes.

for KD of gene 5: $x_{17} = x_{27} = x_{37} = x_{47} = x_{67} = 0$

for KD of gene 6: $x_{18} = x_{28} = x_{38} = x_{48} = x_{58} = x_{78} = 0$
$x_{19} = x_{29} = x_{39} = x_{49} = x_{59} = x_{79} = 0$

for KD of gene 7: $x_{110} = x_{210} = x_{310} = x_{410} = x_{510} = x_{610} = 0$

Statement 3) As an example, consider gene-4. Knockdown of gene-4 shows no effect on genes-7, and 10. Therefore these genes must be activated by the genes which are still active, i.e. the genes activated at the current time step or the previous time steps. The genes which are active when gene-4 is knocked down are gene-1, 2, 3 and 5. Therefore, there must be at least 1 edge from genes 1, 2, 3 and 5 into each of the genes 7 and 10. The corresponding constraints can be written as follows:

$x_{13} + x_{23} + x_{43} >= 1$
$x_{15} + x_{25} + x_{35} >= 1$
$x_{17} + x_{27} + x_{37} + x_{57} >= 1$
$x_{310} + x_{510} + x_{710} >= 1$

Note that x_{110} and x_{210} are direct connections between the sources and the sinks and therefore they are zero.

Similarly, for the rest of the genes, we can write;

for KD of gene 1: genes 2, 5, 7, 10 are active
$x_{25} >= 1$
$x_{27} + x_{57} >= 1$
$x_{510} + x_{710} >= 1$

for KD of gene 2: all other genes are active.

$x_{13} + x_{43} + x_{53} >= 1$
$x_{14} + x_{34} + x_{54} >= 1$
$x_{15} + x_{35} + x_{45} >= 1$
$x_{16} + x_{36} + x_{46} + x_{56} + x_{76} >= 1$
$x_{17} + x_{37} + x_{47} + x_{57} + x_{67} >= 1$
$x_{38} + x_{48} + x_{58} + x_{68} + x_{78} >= 1$
$x_{39} + x_{49} + x_{59} + x_{69} + x_{79} >= 1$
$x_{310} + x_{410} + x_{510} + x_{610} + x_{710} >= 1$

for KD of gene 3: all other genes are active.

$x_{14} + x_{24} + x_{54} >= 1$
$x_{15} + x_{25} + x_{45} >= 1$
$x_{16} + x_{26} + x_{46} + x_{56} + x_{76} >= 1$
$x_{17} + x_{27} + x_{47} + x_{57} + x_{67} >= 1$
$x_{48} + x_{58} + x_{68} + x_{78} >= 1$
$x_{49} + x_{59} + x_{69} + x_{79} >= 1$
$x_{410} + x_{510} + x_{610} + x_{710} >= 1$

for KD of gene 5: genes 3, 4, 6, 8 and 9 are active.

$x_{13} + x_{23} + x_{43} >= 1$
$x_{14} + x_{24} + x_{34} >= 1$
$x_{16} + x_{26} + x_{36} + x_{46} >= 1$
$x_{38} + x_{48} + x_{68} >= 1$
$x_{49} + x_{59} + x_{69} >= 1$

for KD of gene 6: genes 7 and 10 are active

$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} >= 1$
$x_{310} + x_{410} + x_{510} + x_{710} >= 1$

for KD of gene 7: genes 6, 8 and 9 are active

$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} >= 1$
$x_{38} + x_{48} + x_{58} + x_{68} >= 1$
$x_{39} + x_{49} + x_{59} + x_{69} >= 1$

The complete ILP formulation is given below for the problem with combined time series and perturbation data. Here; *n* is the total number of nodes, *s* and *t* are the number of source and sink nodes in the system, respectively. The node set *V* is defined as the union of all node sets defined at each time level according to the given time series data, i.e. V=V_s U V_2 U V_3 U … U V_x U V_{x+1} U … V_p U V_t. Therefore, V_s and V_t are the set of source and sink nodes; V_2, V_3, V_x, V_{x+1} are the nodes activated at the corresponding time levels; and V_p is the set of nodes activated just before the sink nodes. V_d is the set of downstream nodes that are activated after the knockdown of node *i*.

$$\textit{Minimize} \quad \sum_{i=1}^{s}\sum_{j=s+1}^{n-t}\left|x_{ij}-w_{ij}\right| + \sum_{i=s+1}^{n-t}\sum_{\substack{j=s+1 \\ i\neq j}}^{n}\left|x_{ij}-w_{ij}\right|$$

Subject to

1) $\sum_{j\in V_1} x_{ij} \geq 1$ for all $i \in V_s$

2) $\sum_{i\in V_p} x_{ij} \geq 1$ for all $j \in V_m$

3) $\sum_{j\in V_d} x_{ij} \geq 1$ for all $i \in V_x$

$\sum_{i\in (V-V_x)} x_{ij} \geq 1$ for all $j \in V_{x+1}$

4) KD of $i \in V_p$ results in silencing of $j \in V_t$:

$x_{ij} = 1$

5) KD of $i \in (V - V_s)$ results in silencing of $j \in (V - V_s)$:

$x_{kj} = 0$ for all $k \in (V - V_d - \{i\})$

6) KD of $i \in (V - V_s)$ has no effect on $j \in (V - V_s)$:

$$\sum_{k\in (V-V_d-\{i\})} x_{kj} \geq 1$$

If only time series data is available, the first three constraints should be used. If any perturbation experiment result is available, then the last three constraints should be added. Addition of these constrains to the constraints given in previous section results in more accurate network reconstruction.

Note that it is not required for each gene to be knocked down in this approach. However, each knockdown results in more strict constraints and therefore provides more accurate results.

In addition to time series data, we can use any novel biological information about the interactions of the genes. With this information we can construct more accurate networks which are similar to the actual network.

When only time series data is used, the total number of constraints is:

$$2(n - s - t) + s + t = 2n - s - t$$

which has an order of O(n). This is much lower than our previous approach which has an exponential increase in the number of constraints, while this increase is linear for the current approach. Therefore, a network with thousands of genes can be solved easily with this approach. When the perturbation data is used with time series data, the computational time is not affected significantly although there are additional constraints. Since knockdown results of n genes can be known, and for each knockdown at most n constraints can be written (n genes can be active), there can be at most n^2 additional constraints. Although the additional constraints increase the order of the problem to O(n^2), networks with several thousands of genes can be reconstructed in seconds on a desktop computer by this approach.

4.3 Handling of Noisy Perturbation Data

Due to the nature of the perturbation experiments, there is noise in the expression values. Noise in the perturbation data may result in infeasible solution if the constraints stated above are used. For example, consider a network with noise in its perturbation data and assume that genes 1 and 2 are activated at time t_x and gene 3 is activated at time t_{x+1} according to the time series data. On the other hand, the perturbation data states that with knockdown of gene 2, gene 3 is repressed. According to Statement 2 given for perturbation data, the edge from gene 1 to gene 3 should be zero. Then, we have $x_{13} = 0$. Moreover, assume that gene 3 is repressed with the knockdown of gene 1 also. Therefore, we also have $x_{23} = 0$. According to the first constraint due to the time series data (there should be at least 1 edge going into an intermediate node from the upstream nodes activated at all previous time points), we have the constraint $x_{13} + x_{23} \geq 1$, which contradicts the two statements given above ($x_{13} = 0$ and $x_{23} = 0$). This contradiction happens if a gene is repressed by at least two genes which are activated at the same time point. In the given example, genes 1 and 2 are activated in time t_x and they both repress gene 3, which causes a conflict in the constraints. This condition implies that there is an uncertainty in the data about these genes. In order to handle such conflicts, two alternative ways can be utilized:

1) Conflicting Constraint Relaxation (CCR): In CCR, we neglect all the equality constraints for the genes that are conflicting since there is an uncertainty in the information about these genes. None of the equality constraints should include the conflicting genes and no equality constraints should be written at the time level when the conflict occurs.

2) Conflicting Data Elimination by Expression Values (CDEE): Instead of neglecting all equality constraints due to conflicts, it may be possible to extract some useful information by assuming one of the conflicting states that belong to the

corresponding gene is correct. This can be done by choosing the gene which has the lowest expression level. For example, in the example above, we compare the expression levels of gene 3 when gene 1 and gene 2 are knocked down. If the expression level of gene 3 is lower when gene 1 is knocked down, we choose it as correct and therefore keep the state value of gene 3 as 0 while replacing its state value with 1 in knockdown of gene 2.

The CCR method is a more conservative method that ignores all biological data in case of a conflict. The network is constructed with minimum information available. On the other hand, the CDEE method aims to recover as much evidence as possible for the construction of the network. It uses a simple heuristic in assessing the confidence of an observation by using the level of repression of a gene due to a knock-down.

4.4 Data sets

Synthetic Dataset

Performing simulations with synthetic data is advantageous because it is possible to test the method at different conditions by applying noise on the actual synthetic data and observe the changes in the results. In order to evaluate the performance of our method, we use 2 synthetic datasets for networks with 10 and 20 genes. The datasets are explained below.

a) 10 node network example:

The schematic of an example 10 node actual network is given in Figure 4.3. Assume that the filtered gene expression data for this network is given as the following (consistent with the given network): Genes 1, 2 @ Time 1; Genes 3, 4, 5 @ Time 2; Genes 6, 7 @ Time 3; Genes 8, 9, 10 @ Time 4 are active. Since nodes 1 and 2 are activated at Time 1, they are the sources and since nodes 8, 9 and 10 are activated at the last time point, they are the sink nodes.

Using this data, 25 mutated reference networks are created and they are solved to satisfy the given gene expression data. Generation of these 25 synthetic data sets for each actual network is done by inserting edges to the actual networks at a certain rate and removing edges at the same or at a different rate, i.e., inserting or removing different number of edges from each of them, generating both dense and sparse networks. The insertion/deletion rates range from 10 to 50 percent, each. In this data set, we have 25 unique reference networks with different number of edges. The perturbation data is extracted from Figure 4.3 and is given as the following (only repressed genes are given for the corresponding knockdown): Genes 3, 4, 6, 8, 9 after KD of gene 1; genes 6, 8, 9 after KD of gene 4; genes 7, 10 after KD of gene 5; genes 8, 9 after KD of gene 6; gene 10 after KD of gene 7.

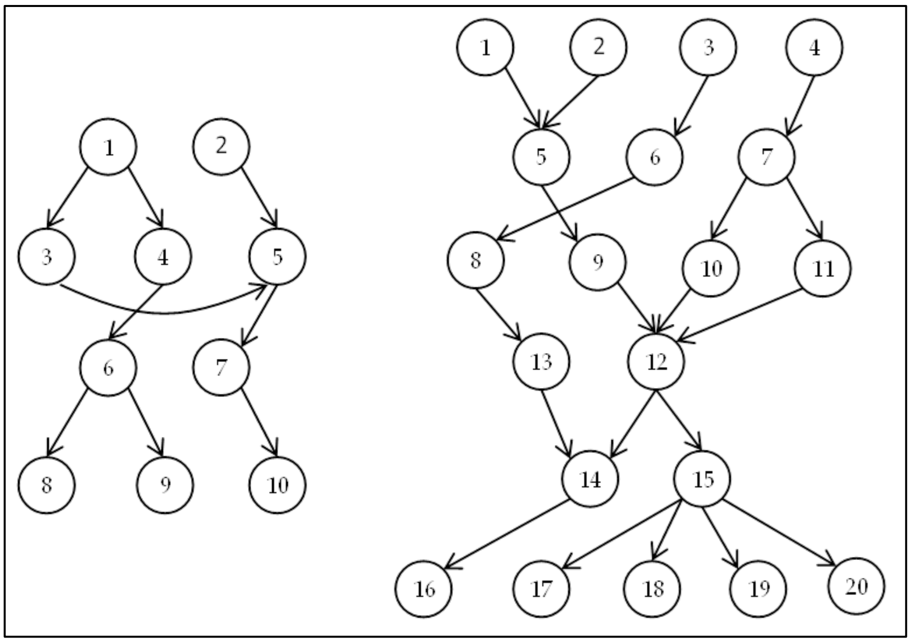


Figure 4.3 Actual synthetic 10- and 20- node networks used in the experiments.

Perturbation data: We have generated noisy perturbation data sets for each actual network with the noise rates of 0.02, 0.04, 0.1, 0.2. This is done by toggling r × (# of genes)^2 of perturbation data fifty-one times and rounding off the average to the nearest number (0 or 1) to generate noisy perturbation data.

b) 20 node network example:
The schematic of the example 20 node network is also given in Figure 4.3. The base gene expression data and the perturbation data for this network are derived from this figure. Similar to the 10-node network example, 25 mutated reference networks and noisy perturbation data are created from the actual network and they are solved to satisfy the given gene expression data.

Real Dataset
Mice: Agouti signaling pathway in mice is selected to test the performance with a real dataset. K14-Agouti transgenic mice are hemizygous for the transgene and are non-agouti (a/a) at the endogenous agouti locus (Bultman et al., 1992) Mice of the transgenic line TG2579K14iA (Kucera et al., 1996) are backcrossed to C57BL/6J mice for more than 10 generations in order to obtain K14-Agouti transgenic mice that are congenic on the C57BL/6J genetic background. C57BL/6J (a/a) littermates were used as controls for K14-Agouti transgenic mice.

RNA Isolation and cDNA Preparation: Dorsal skin sections from K14-Agouti transgenic mice and *a/a* control mice are dissected and snap-frozen in liquid nitrogen. Total RNA was isolated using TRIzol (Invitrogen), followed by on-column DNAse treatment and a purification step with the Qiagen RNeasy Mini Kit. SuperScript II Reverse Transcriptase (Invitrogen Life Technologies) was used to generate cDNA according to the manufacturer's instructions.

Real-Time qRT-PCR: The Cepheid SmartCycler real-time PCR machine is used for the qRT-PCR measurements. The TaqMan primers and probes were purchased from Applied Biosystems (TaqMan Assays-on-Demand Products Assay numbers: Mm00438337_m1, Mm00476174_m1, Mm00516876_m1, Mm00515219_s1, Mm00448100_m1, Mm00435540_m1, Mm00440911_m1, Mm00439518_m1, Mm00456961_m1, Mm00436931_m1). The 18S RNA (*Rn18s*) gene was used as an internal control to normalize experimental data. The threshold cycle (C_T) for all experimental genes was first normalized to the corresponding *Rn18s* C_T. Relative fold differences were then determined using the $2^{-\Delta\Delta CT}$ method (Livak and Schmittgen, 2001) by comparing each experimental sample to the control sample. At least two transgenic and two control mice were analyzed at each time point, and the expression level of each gene was measured in duplicate for each mouse. The results are as shown in Table 4.3 (Son, 2006).

Table 4.3 Differential expression levels of 10 selected genes in the skin of K14-Agouti mice compared to control mice at various ages. Fold differences for each gene between transgenic animals and their littermate controls are shown.

Mice ID	Age in Days	Hras1	Krt1-14	Ivl	Pdgfa	Plaur	Stat1	Stat3	Csnk2a2	Pcna	Tert
Y27	33	0.87	1.40	1.91	0.89	0.91	0.40	0.80	0.72	0.86	1.40
Y28	33	1.27	2.40	1.11	1.26	1.52	0.60	1.10	0.68	1.25	1.46
1225	46	1.74	1.13	2.93	1.20	1.33	0.90	0.90	1.05	2.78	3.23
1227	46	1.48	1.38	2.17	0.77	1.18	0.85	0.80	0.56	1.27	1.44
1311	47	0.42	0.39	1.06	0.53	0.21	0.63	1.09	0.53	0.35	0.46
1312	47	1.40	1.18	4.27	4.29	2.24	3.67	4.00	2.17	1.71	2.79
1314	47	1.95	2.23	5.05	2.17	5.58	2.39	2.71	3.17	1.76	2.84
1315	47	1.64	1.40	5.08	2.64	4.06	3.34	2.64	4.76	2.28	4.54
1264	48	0.81	0.80	2.58	1.88	2.56	1.20	1.00	1.50	3.15	1.31
1265	48	0.72	0.70	5.22	0.97	1.30	0.80	0.80	1.40	3.20	1.36
1306	54	2.09	1.59	5.41	2.87	4.11	2.08	2.49	3.42	2.50	1.68
1307	54	2.79	3.13	7.52	4.32	6.87	5.37	3.53	4.55	4.39	2.40
1254	74	16.62	4.97	11.47	9.82	10.97	7.70	5.70	0.90	2.29	2.08
1256	74	2.47	3.31	8.34	4.26	3.99	5.30	2.70	2.72	3.84	2.51
1257	74	7.70	7.43	8.14	10.74	7.21	9.70	6.00	1.95	2.15	2.05
1295	81	1.31	0.75	4.32	0.93	1.34	1.21	0.94	2.23	2.11	1.20
1296	81	0.66	0.74	2.06	1.34	0.48	0.52	0.58	1.31	0.97	1.17
Y100	89	3.02	5.64	12.17	11.79	3.76	4.04	4.27	3.77	4.63	1.83
Y200	89	1.22	2.15	8.37	4.10	2.04	1.13	1.92	1.82	1.79	1.46
1288	130	1.41	2.55	5.72	4.42	0.52	0.86	2.31	1.02	0.79	1.80
1290	130	0.53	0.85	0.81	0.99	0.91	1.29	1.26	0.70	0.66	0.45
1286	187	1.64	0.92	3.63	0.99	0.88	3.48	0.66	1.08	1.69	1.48

Genemania Network Analysis: GeneMANIA is a gene network analysis tool that presents known and predicted relationships between the query genes and gene sets through the large set of functional association data comprising protein and genetic interactions, pathways, co-expression, co-localization and protein domain similarity (by Jan'13, indexing 1,464 association networks containing 292,680,904 interactions mapped to 149,747 genes from 7 organisms) (Warde-Farley et al., 2010). GeneMANIA query is limited to the 10 genes with differential expression levels in K14-Agouti mouse with the NCBI gene symbol: Gene IDs listed as following: a: 50518, Krt14:16664, Ivl: 16447, Tert:21752, Plaur:18793, Pdgfa:18590, Hras1:15461, Pcna:18538, Csnk2a2:13000, Stat1:20846, and Stat3:20848. Default options are used and the "co-expression" analysis is extracted for the evaluation.

Further processing of data: The original fold-change data is further processed to make it suitable for use in our model. The data at each time level is first averaged and then normalized. At most three genes having the largest normalized value are assumed to be active at the corresponding time level. We set a value of 0.02 to be the maximum difference between two genes to be considered as active. After marking a gene as active at a certain time level, it is considered to stay active at all proceeding time levels and omitted from the further evaluation of active genes. If there are more than three genes within 0.02 at a certain time, we omit this time level from our pre-processed data. The genes which are determined to be active at each time level according to this criterion are given in Table 4.4 in bold face. Note that at Time 5 and Time 8, no genes are determined to be active since there are more than three genes within the 0.02 range of the highest value at these time levels.

4.5 Experimental Evaluation

The approach described above is applied to the 10 and 20-gene networks by using only time-series microarray data, combined time-series and perturbation data, and combined noisy data. We use recall (R) and precision (P) values as accuracy measures, which are described in Chapter 2, for evaluation of the method.

The results are given comparatively in Table 3 and Table 4 for the 10 and 20-gene networks, respectively. In these tables, the first column shows the amount of noise applied to the actual network in percentages; the second main column exhibits the results when only time series data is used. The rest of the columns show the results for combined time series data with pure or noisy perturbation data.

For the 10-gene network, although the accuracies of the results when only time series data is used are high, addition of new information by perturbation experiments result in even higher accuracy. All the recall values become 1.0, meaning that the constructed networks have all edges that the actual network has. The major gain is in precision values. For the network with 50% insertion and 40% deletion, the precision value increases from 0.62 to 1.0, therefore the actual network is found perfectly.

Similar results are obtained for the 20-gene network. Both precision and recall values are improved with the additional information about the network. These improvements can be as high as 0.30 in recall and 0.44 in precision.

Table 4.4 Activated genes at each time point in agouti signaling network.

	Hras1	Krt1-14	Ivl	Pdgfa	Plaur	Stat1	Stat3	Csnk2a2	Pcna	Tert
Time 1	0,094	**0,167**	0,132	0,094	0,106	0,044	0,083	0,062	0,092	0,125
Time 2	0,111	0,086	**0,175**	0,068	0,086	0,060	0,058	0,055	0,139	**0,161**
Time 3	0,057	0,054	0,162	0,101	**0,126**	0,105	0,109	0,111	0,064	0,111
Time 4	0,046	0,045	0,235	0,086	0,116	0,060	0,054	0,087	**0,191**	0,080
Time 5	0,067	0,065	0,177	0,098	0,150	0,102	0,082	0,109	0,094	0,056
Time 6	**0,153**	0,090	0,160	**0,142**	0,127	**0,130**	0,082	0,032	0,047	0,038
Time 7	0,075	0,057	0,244	0,087	0,069	0,066	0,058	**0,135**	0,118	0,091
Time 8	0,052	0,096	0,254	0,196	0,072	0,064	0,077	0,069	0,079	0,041
Time 9	0,065	0,114	0,218	0,181	0,048	0,072	**0,119**	0,058	0,049	0,075

These results show that time series data gives the underlying structure of the constructed network. In addition to these data, integration of any biological information (such as perturbation data, RNAi data, novel information about some interactions, and etc.) to our approach makes it accurate.

Note that, the perturbation data used in these calculations are perfect, i.e. they are produced from the actual network suitable with the graph. Therefore the results may not reflect the reality. In reality, the perturbation data is a noisy data and thus, noise should be added to the perturbation data.

We performed all the previous simulations with noisy perturbation data to inspect the effect of noise and accuracy of our approach. 2, 4, 10 and 20 percent noise is added to the perturbation data by randomly changing the state variables either from 0 to 1 or from 1 to 0. Let q be the number of expression levels obtained from perturbation experiments and p_{rate} be the percentage of noise to be applied to this data. $q \times p_{\text{rate}}$ expression values are randomly toggled to generate new perturbation data.

The results with noisy data for 10 and 20 genes are given in Table 4.5 and Table 4.6, respectively. In these calculations, the conflicts are handled using the first approach explained above.

For the 10-gene network, addition of 2 percent noise does not make any difference. The reason is that the network is small and therefore the corresponding noise addition is small. When there is 4 percent noise in the perturbation data, very little change in only precision values is obtained. A significant difference in precision is obtained at 10 and 20 percent noise levels while the difference is not much for recall values. In general, as the noise level increases, decrease in recall and precision values is observed. However, it is found that for the same experiment (same insertion/deletion), recall values decrease less than the precision values. It means that despite high noise levels, our model is accurate in predicting the real network. Besides, it finds extra interactions with an increase in noise. It is possible to search for the presence of these interactions in wet-lab studies. In addition, in the next example (20-gene network), we observe that the decrease in recall and precision values are not only related with the noise itself, but also related with the amount of conflicts that the given noise created.

Table 4.5 Comparison of results for the 10-gene network

	Only time series data		Time series and perturbation data									
			No noise		2% noise		4% noise		10% noise		20% noise	
insertion/ deletion	R	P	R	P	R	P	R	P	R	P	R	P
10/10	1.00	0.82	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.69	0.89	0.67
10/20	1.00	0.82	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.69	0.89	0.62
10/30	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.89	0.62
10/40	0.89	0.73	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.78	0.64
10/50	0.89	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.89	0.67
20/10	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.89	0.62
20/20	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.89	0.67
20/30	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.89	0.62
20/40	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.89	0.62
20/50	0.89	0.80	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.69	0.78	0.58
30/10	1.00	0.75	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.69	0.89	0.62
30/20	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.89	0.67
30/30	1.00	0.75	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.64	0.78	0.58
30/40	0.89	0.67	1.00	1.00	1.00	1.00	1.00	1.00	0.89	0.62	0.78	0.64
30/50	0.78	0.64	1.00	1.00	1.00	1.00	1.00	0.90	1.00	0.69	0.89	0.62
40/10	1.00	0.69	1.00	1.00	1.00	1.00	1.00	0.90	1.00	0.69	0.89	0.62
40/20	0.89	0.67	1.00	0.90	1.00	0.90	1.00	0.90	0.89	0.62	0.78	0.54
40/30	0.89	0.67	1.00	0.82	1.00	0.82	1.00	0.82	1.00	0.64	0.89	0.57
40/40	1.00	0.69	1.00	1.00	1.00	1.00	1.00	1.00	0.89	0.67	0.78	0.58
40/50	0.89	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.89	0.67
50/10	1.00	0.64	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.60	0.89	0.57
50/20	1.00	0.64	1.00	0.90	1.00	0.90	1.00	0.82	1.00	0.69	0.89	0.67
50/30	0.89	0.67	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.60	0.89	0.57
50/40	0.89	0.62	1.00	1.00	1.00	1.00	1.00	0.90	0.89	0.62	0.67	0.50
50/50	0.89	0.62	1.00	0.90	1.00	0.90	1.00	0.90	1.00	0.64	0.89	0.57

R: recall, P: precision

Table 4.6 Comparison of results for the 20-gene network

	Only time series data		Time series and perturbation data									
			No noise		2% noise		4% noise		10% noise		20% noise	
insertion/ deletion	R	P	R	P	R	P	R	P	R	P	R	P
10/10	0.95	0.90	1.00	1.00	1.00	1.00	0.95	0.79	0.85	0.61	0.90	0.60
10/20	0.90	0.82	1.00	1.00	1.00	1.00	1.00	0.91	0.75	0.54	0.85	0.59
10/30	0.85	0.77	0.95	0.86	0.95	0.86	0.90	0.82	0.70	0.54	0.80	0.57
10/40	0.85	0.77	0.90	0.95	0.90	0.95	0.90	0.90	0.75	0.56	0.70	0.54
10/50	0.70	0.67	0.90	0.90	0.90	0.90	0.85	0.77	0.70	0.54	0.65	0.50
20/10	0.95	0.79	0.95	0.95	0.95	0.95	0.95	0.79	0.80	0.57	0.90	0.60
20/20	0.90	0.75	0.90	0.95	0.90	0.95	0.85	0.81	0.70	0.48	0.75	0.54
20/30	0.85	0.71	1.00	1.00	1.00	1.00	0.95	0.79	0.85	0.61	0.75	0.48
20/40	0.80	0.70	0.95	0.90	0.95	0.90	0.85	0.68	0.55	0.39	0.65	0.42
20/50	0.80	0.70	0.95	0.95	0.95	0.95	0.90	0.82	0.60	0.43	0.65	0.46
30/10	0.95	0.73	1.00	0.95	1.00	0.95	0.95	0.76	0.80	0.53	0.90	0.56
30/20	1.00	0.77	1.00	0.91	1.00	0.91	1.00	0.80	0.85	0.55	0.85	0.52
30/30	0.85	0.65	0.80	0.80	0.80	0.80	0.85	0.71	0.75	0.52	0.70	0.50
30/40	0.75	0.60	0.85	0.77	0.85	0.77	0.75	0.65	0.70	0.52	0.75	0.50
30/50	0.70	0.56	0.95	0.95	0.95	0.95	0.85	0.77	0.65	0.43	0.65	0.43
40/10	0.95	0.68	0.95	0.90	0.95	0.86	0.95	0.76	0.80	0.53	0.90	0.53
40/20	1.00	0.71	0.95	0.83	0.95	0.83	0.90	0.72	0.75	0.50	0.90	0.55
40/30	0.90	0.64	0.95	0.95	0.95	0.95	0.95	0.79	0.80	0.50	0.80	0.47
40/40	0.75	0.54	0.90	0.86	0.90	0.86	0.85	0.71	0.65	0.43	0.65	0.42
40/50	0.65	0.46	0.95	0.90	0.95	0.90	0.90	0.72	0.65	0.41	0.70	0.42
50/10	1.00	0.67	1.00	0.91	1.00	0.91	0.95	0.66	0.80	0.47	0.85	0.46
50/20	0.95	0.68	0.95	0.86	0.95	0.86	0.90	0.62	0.75	0.52	0.80	0.46
50/30	0.75	0.54	0.95	0.83	0.95	0.83	0.85	0.65	0.70	0.45	0.75	0.48
50/40	0.75	0.50	0.85	0.77	0.85	0.74	0.80	0.62	0.75	0.47	0.75	0.44
50/50	0.70	0.50	0.90	0.86	0.90	0.82	0.80	0.57	0.65	0.42	0.65	0.38

R: recall, P: precision

For the 20-gene network, there is a slight difference in the results at 2 percent noise level compared to the results without noise. The difference is only in precision which is at most 0.04 and can be considered as negligible. When the noise level is 4 percent, the decrease in both recall and precision values, especially in precision values, are more significant compared to the 10-node network. This is because the network is larger and therefore the noise is more. As the noise level increases to 10 percent, the difference becomes even more significant. This is again due to the large number of noisy state variables in the perturbation data. However, at 20 percent noise level, we see an improvement in most of the recall and precision values compared to 10 percent noise level. Since there is more noise, it is reasonable to expect a less accurate result, however this is not the case for this example. The reason lies behind the number of conflicting data that are created during the noise addition process, rather than the amount of the noise. As the number of conflicts increases, less equality constraints (i.e. x_{ij}=0) are generated, which in turn results in an increase in solution space. This is what happens in this example: by chance, the perturbation data with 10 percent noise contains more conflicting genes than the data with 20 percent noise. Therefore, less number of equality constraints result in less accurate results at 10 percent noise level.

4.6 Evaluation of the analysis results on the real dataset in comparison to GeneMANIA networks

The method is applied to mice agouti signaling network using relative change of expression levels by time, i.e. without any perturbation experiment data (Figure 4.4). As no literature curated network or any network taken from other databases are taken as the reference network in the analysis, the reference network for the method was empty. As the K14-Agouti mice caries the overexpressed agouti transgene, agouti is assumed to start the stimulation in our computational experiments. The constructed network with our method and the co-expression network built by GeneMANIA are given in Figure 4.4. It can be seen that the method finds several common interactions and sub-networks.

When we have searched for the edge Tert-Pcna through KEGG database, HTLV-I infection Pathway (map05166) was the only KEGG pathway result returned, where Tert and Pcna represented together within the same pathway. As also presented on the KEGG pathway, both Tert and Pcna are signaling molecules transcribed due to induction of intercellular molecules and do not directly induce each other’s transcription. As the proposed model is based on time series expression data, the Tert-Pcna interaction proposed in the model might be suggesting the boundary between the initial signaling cascade and the following loop of proliferation signals in the second phase (Figure 4.5), which keeps the cells under pressure of agouti overexpression at the proliferative stage, thus susceptible to oncogenic stimuli (Kuklin et al., 2004).

4.7 Discussion

In this chapter, two ILP based methods for reconstruction of gene regulatory networks are proposed. While one of the methods uses only the time series data, the other one uses combined time-series and perturbation data that employ gene expression levels. The additional perturbation data result in reconstructing more accurate networks especially when the data include no or very little noise. Moreover, integrating genetic perturbations with gene expression data enables learning of causal regulatory relations between genes.

Additionally, we show that our method works well on a real dataset. The method is applied on the mouse agouti signaling network and it is found that it identifies several interactions and sub-networks very well. Also, our method is able to scale for large networks easily.

In conclusion, since studying and understanding signaling networks is of utmost importance for the identification and treatment of several diseases, the method proposed in this chapter is valuable and convenient for construction of novel signaling networks.

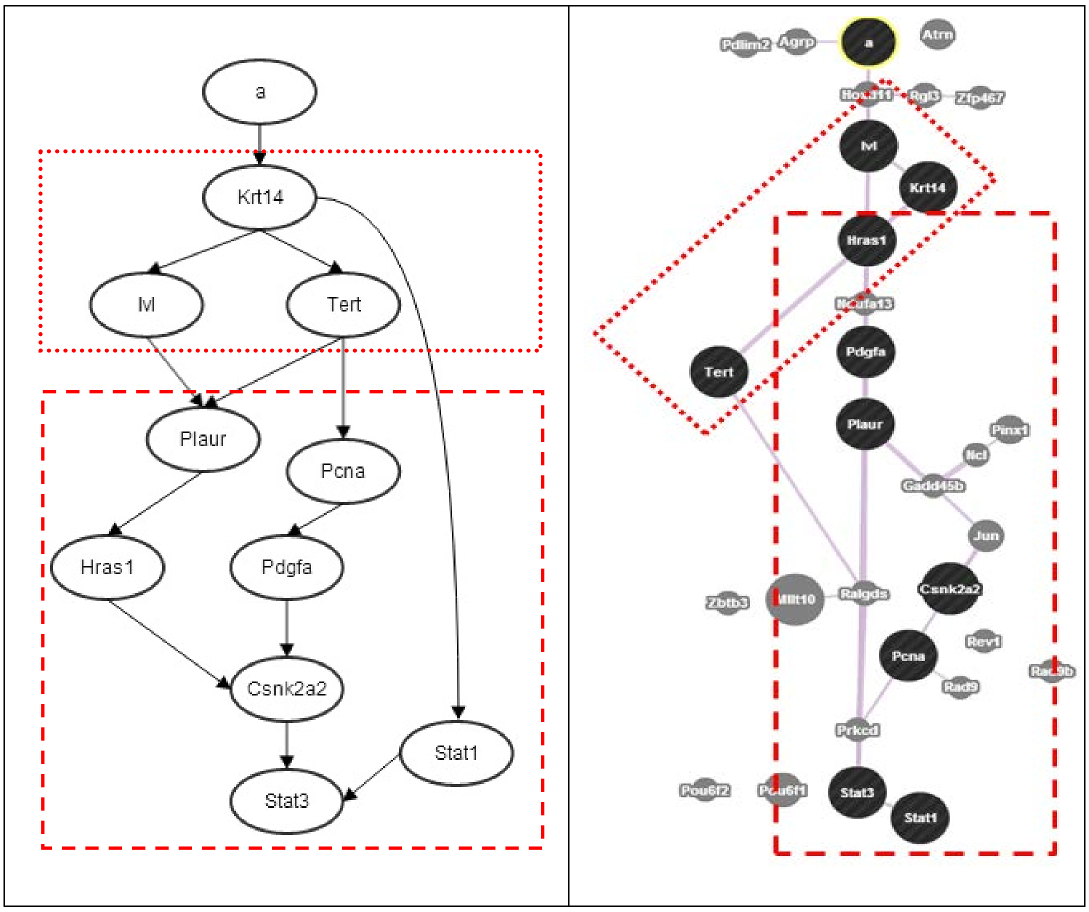


Figure 4.4 Shared interaction between proposed vs. co-expression networks: The agouti signaling network model build based on differential expressions level changes over time with no pre-assumptions is presented on the left. The signaling network compared to the co-expresion network built by GeneMANIA analysis is on the right. The two main shared sub-networks are labeled with the corresponding boxes.

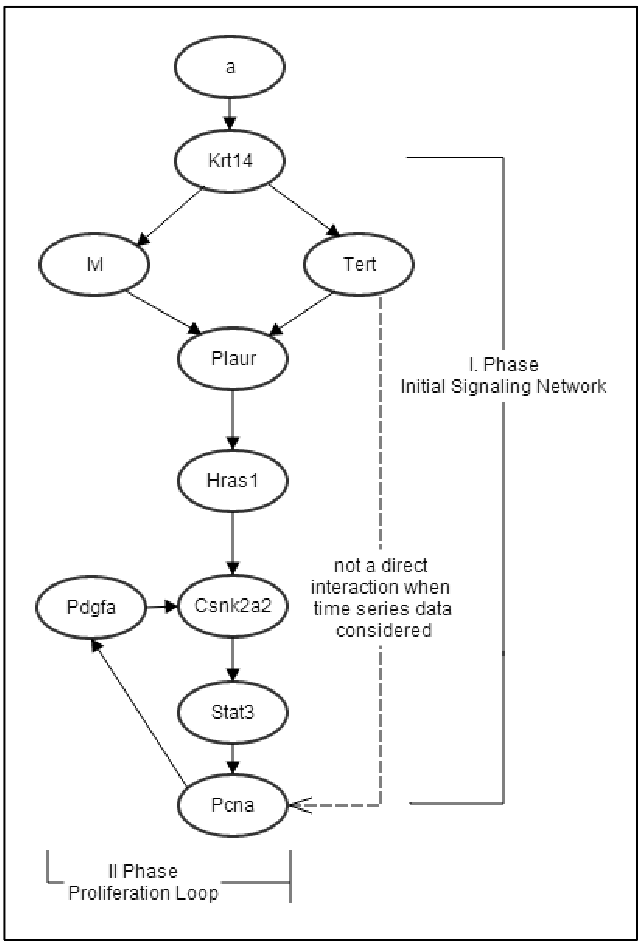

a
Krt14
Ivl
Tert
Plaur
Hras1
Pdgfa
Csnk2a2
Stat3
Pcna
I. Phase
Initial Signaling Network
not a direct interaction when time series data considered
II Phase
Proliferation Loop

Figure 4.5 I. Phase : Initial signaling network, II. Phase: Proliferation loop.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this dissertation, we have presented integer linear programming based methods for construction of biological networks that help biologists to reconstruct signaling networks from screening experiments. Our contributions can be grouped under three main categories; reconstruction of small size, single receptor single reporter networks optimally, large scale network reconstruction with single receptor and single reporter networks, and large scale network reconstruction with multiple receptor multiple and reporter networks.

For reconstruction of small size, single receptor single reporter networks optimally, we presented an integer linear programming approach which can reconstruct more accurate networks compared with state of the art methods and guarantees optimal solution. Constructed network topologies contain all genes that are known to be members of the network, which makes the final network topology connected. Moreover the constructed network is consistent with the RNAi data which shows that our approach can figure out all critical genes correctly. However, it has a drawback in finding large scale networks because of the exponential growth of the number of constraints with increasing number of genes in the network. With the available computer resources, it scales up to 10 nodes in reasonable time.

For large scale network reconstruction with single receptor and single reporter networks, we have presented divide and conquer based approaches applied on several networks with single receptor and single reporter, which can be sparse or dense, have small or large number of nodes and contain noisy data. The results show that our methods can construct networks better than the state of the art methods in terms of recall and precision measures, scalability, robustness, and consistency with the RNAi data. For dense and large networks, while the state of the art methods fail to construct networks in 1 hour, we are able to find solutions in minutes. While these proposed methods can reconstruct large scale networks easily, they are not able to reconstruct networks with multiple receptor and multiple reporters.

For large scale network reconstruction with multiple receptor and multiple reporter networks, we have presented an integer linear programming based graph theoretical model using time series gene expression data. When the time series data is used alone, the method produces a basic structure of the network. The additional biological information in any form such as perturbation data, RNAi data, or novel information about some interactions, improve the results significantly and more accurate networks are constructed with the proposed method To assess the performance of our methods on noisy data, we applied these methods on synthetic data set which have different noise levels. After obtaining promising results on the synthetic data set, we implemented the proposed approach on mouse agouti signaling network as a real data set, and found several interactions and sub-networks correctly. It was shown that the developed methods were able to solve large scale networks easily.

We have represented the activation interactions by our models. To simulate real networks both activation and inhibition interactions are essential and biologically meaningful. As a further study, inhibition interactions can be modeled along with the activations which make the problem more complex because of the additional information that brings additional constraints. Moreover, the methods can be improved by making use of combinatorial knockdowns, i.e. simultaneous knockdowns of multiple genes, which will greatly enhance the reconstructed network.

REFERENCES

Bansal, M., Della Gatta, G., & di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics (Oxford, England), 22*(7), 815-822.

Bornholdt, S. (2008). Boolean network models of cellular regulation: Prospects and limitations. *Journal of the Royal Society, Interface / the Royal Society, 5 Suppl 1*, S85-94.

Bower, J. M., & Bolouri, H. (2001). *Computational modeling of genetic and biochemical networks*. Cambridge, Mass.: MIT Press.

Brass A.L., Dykxhoorn D.M., Benita Y., Yan N., Engelman A., Xavier R.J., Lieberman J., & Elledge S.J. (2008) Identification of host proteins required for HIV infection though a functional genomic screen. *Science*, 319(5865):921-926.

Bultman, S. J., Michaud, E. J., & Woychik, R. P. (1992). Molecular characterization of the mouse agouti locus. *Cell, 71*(7), 1195-1204.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1), 67–103.

Eren Ozsoy, O., & Can, T. (2013). A divide and conquer approach for construction of large-scale signaling networks from PPI and RNAi data using linear programming. *IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM,* 10(4), 869-883.

Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. E., & Mello, C. C. (1998). Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. *Nature, 391*(6669), 806-811.

Friedman, A., & Perrimon, N. (2006). A functional RNAi screen for regulators of receptor tyrosine kinase and ERK signalling. *Nature, 444*(7116), 230-234.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology, 7*(3-4), 601-620.

Froehlich, H., Fellmann, M., Sueltmann, H., Poustka, A., & Beissbarth, T. (2007). Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinformatics, 8*, 386.

Gao, K., & Huang, L. (2013). Achieving efficient \RNAi\ therapy: Progress and challenges. *Acta Pharmaceutica Sinica B, 3*(4), 213.

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science (New York, N.Y.), 301*(5629), 102-105.

Gebert,J., Radde N. &Weber G.-W. (2007) Modeling gene regulatory networks with piecewise linear differential equations, *European Journal of Operational Research*, 181 (3), 1148–1165.

Gitter, A., Klein-Seetharaman, J., Gupta, A., & Bar-Joseph, Z. (2011). Discovering pathways by orienting edges in protein interaction networks. *Nucleic Acids Research, 39*(4), e22.

Hashemikhabir, S., Ayaz, E. S., Kavurucu, Y., Can, T., & Kahveci, T. (2012). Large-scale signaling network reconstruction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 9*(6), 1696-1708.

Hillier, F. S., & Lieberman, G. J. (2001). *Introduction to operations research* (7th ed.) McGraw-Hill.

Kaderali, L., Dazert, E., Zeuge, U., Frese, M., & Bartenschlager, R. (2009). Reconstructing signaling pathways from RNAi data using probabilistic boolean threshold networks. *Bioinformatics (Oxford, England), 25*(17), 2229-2235.

Kaderali, L., & Radde, N. (2008). Inferring gene regulatory networks from expression data.*94*, 33-74.

Kanehisa M. & Goto S. (2000). Kegg: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1) 27-30.

Karlebach,G. & Shamir,R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9, 770–780.

Kerrien, S., Aranda, B., Breuza, L., Bridge, A., Broackes-Carter, F., Chen, C., et al. (2012). The IntAct molecular interaction database in 2012. *Nucleic Acids Research, 40*(Database issue), D841-6.

Knapp, B. & Kaderali, L. (2013). Reconstruction of cellular signal transduction networks using perturbation assays and linear programming. *Plos One, 8*(7), e69220.

Kucera, G. T., Bortner, D. M., & Rosenberg, M. P. (1996). Overexpression of an agouti cDNA in the skin of transgenic mice recapitulates dominant coat color phenotypes of spontaneous mutants. *Developmental Biology, 173*(1), 162-173.

Kuklin, A. I., Mynatt, R. L., Klebig, M. L., Kiefer, L. L., Wilkison, W. O., Woychik, R. P., et al. (2004). Liver-specific expression of the agouti gene in transgenic mice promotes liver carcinogenesis in the absence of obesity and diabetes. *Molecular Cancer, 3*, 17.

Kwon, Y. K., & Cho, K. H. (2007). Analysis of feedback loops and robustness in network evolution based on boolean models. *BMC Bioinformatics, 8,* 430.

Lee,W.P. & Yang,K.C. (2008) A clustering-based approach for inferring recurrent neural networks as gene regulatory networks, *Neurocomputing,* 71 ,600–610.

Li, Z., Li, P., Krishnan, A., & Liu, J. (2011). Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic bayesian network analysis. *Bioinformatics (Oxford, England), 27*(19), 2686-2691.

Liang, S., Fuhrman, S., & Somogyi, R. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing.Pacific Symposium on Biocomputing,* , 18-29.

Livak,K.J. & Schmittgen,T.D. (2001) Analysis of relative gene expression data using real-time quantitative PCR and the 2(-Delta Delta C(T)) Method. *Methods,* 25 (4), 402-8.

Margolin,A.A., Nemenman I., Basso K., Wiggins C., Stolovitzky G., Favera R.D. & Califano A. (2006) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, BMC Bioinformatics, 7(Suppl 1): S7

Markowetz, F., Kostka, D., Troyanskaya, O. G., & Spang, R. (2007). Nested effects models for high-dimensional phenotyping screens. *Bioinformatics (Oxford, England), 23*(13), i305-12.

Milo R., Kashtan N., Itzkovitz S., Newman M. E. J., & Alon U. (2003), On the uniform generation of random graphs with prescribed degree sequences, eprint arXiv:cond-mat/0312028

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science (New York, N.Y.), 298*(5594), 824-827.

Moffat, J., & Sabatini, D. M. (2006). Building mammalian signalling pathways with RNAi screens. *Nature Reviews Molecular Cell Biology, 7*(3), 177-187.

Platanias, L. C. (2005). Mechanisms of type-I- and type-II-interferon-mediated signalling. *Nature Reviews. Immunology, 5*(5), 375-386.

Prados, J., Melguizo, C., Roldan, H., Alvarez, P.J., Ortiz, R., Arias, J.L., & Aranega A. (2013). RNA Interference in the Treatment of Colon Cancer. *BioDrugs*, 27 (4), 317-327.

Rao, D.D., Wang, Z., Senzer, N., Nemunaitis, J. (2013). RNA interference and personalized cancer therapy. *Discovery Medicine*, 15(81),101-110.

Rossi J.J. (2006). RNAi as a treatment for HIV-1 infection, *BioTechniques*, 40(4), 25-29.

Ruths, D., Tseng, J., Nakhleh, L., & Ram, P. (2007). De novo signaling pathway predictions based on protein-protein interaction, targeted therapy and protein microarray analysis. *Systems Biology and Computational Proteomics, 4532*, 108-118.

Sacher, R., Stergiou, L., & Pelkmans, L. (2008). Lessons from genetics: Interpreting complex phenotypes in RNAi screens. *Current Opinion in Cell Biology, 20*(4), 483-489.

Sahin, O., Frohlich, H., Lobke, C., Korf, U., Burmester, S., Majety, M., et al. (2009). Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Systems Biology, 3*, 1-0509-3-1.

Sarkies, P. & Miska, E.A. (2013), RNAi pathways in the recognition of foreign RNA: antiviral responses and host-parasite interactions in nematodes. *Biochemical Society Transactions*, 41(4), 876-880.

Schafer, J., & Strimmer, K. (2005). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics (Oxford, England), 21*(6), 754-764.

Scott, J., Ideker, T., Karp, R. M., & Sharan, R. (2006). Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology, 13*(2), 133-144.

Seo, M. Y., Abrignani, S., Houghton, M., & Han, J. H. (2003). Small interfering RNA-mediated inhibition of hepatitis C virus replication in the human hepatoma cell line huh-7. *Journal of Virology, 77*(1), 810-812.

Singh, R., & Massachusetts Institute of Technology. Department of Electrical Engineering and Computer Science. (2012). *Algorithms for the analysis of protein interaction networks*

Steffen, M., Petti, A., Aach, J., D'haeseleer, P., & Church, G. (2002). Automated modelling of signal transduction networks. *BMC Bioinformatics, 3*, 34.

Son, Y.A. (2006) Differential Expression of Skin Cancer and Hair-Follicle Cycle Regulated Genes in Tumor Susceptible K14-Agouti Mice. PhD Thesis, University of Tennessee, Knoxville, USA.

Wagner,A. (2001) How to reconstruct a large genetic network from n gene perturbation in n2 easy steps. *Bioinformatics*, 17, 1183–97.

Warde-Farley, D., Donaldson, S. L., Comes, O., Zuberi, K., Badrawi, R., Chao, P., et al. (2010). The GeneMANIA prediction server: Biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research, 38*(Web Server issue), W214-20.

Zhang, Y., Deng, Z., Jiang, H., & Jia, P. (2007). Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural EM. *Proceedings of the 4th International Conference on Data Integration in the Life Sciences*, Philadelphia, PA, USA. pp. 204-214.

Zou, M., & Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England), 21*(1), 71-79.

CURRICULUM VITAE

Oyku EREN OZSOY

Education

June 2014	Doctor of Philosophy in Health Informatics, Middle East Technical University, Ankara, Turkey.
June 2008	Master of Science in Computer Engineering, Baskent University, Ankara, Turkey.
January 2006	Bachelor of Science in Computer Engineering, Baskent University, Ankara, Turkey.

Field of Study

Bioinformatics, Computational Biology, Mathematical Mmodeling and Algorithm Development, Optimization Algorithms, Machine Learning Techniques, Data Mining, Biological Networks.

Professional Experience

Ph.D. scholar, TUBITAK – Short Term R&D Funding Program, Prediction of Signal Transduction Networks from Gene Expression and RNAi Data, Middle East Technical University, Ankara, Turkey, 2013- 2014

Ph.D. researcher, SYSPATHO:EU-FP7 Research Project, New Algorithms for Host Pathogen Systems Biology, Middle East Technical University, Ankara, Turkey, 2011- 2013

Teaching assistant, Informatics Institute, Hacettepe University, Ankara, Turkey, 2010-2011

Visiting Scientist, BioQuant center of Heidelberg University, Heidelberg, Germany, August 2011- November 2011

Teaching and research assistant, Department of Computer Engineering, Baskent University, Ankara, Turkey, 2006-2010

Publications

O.Eren Ozsoy, Y. Aydin Son, T. Can
Reconstruction of Signaling and Regulatory Networks using Time-Series and Perturbation Experiments. *submitted*.

O. Eren Ozsoy, T. Can
A Divide and Conquer Approach for Construction of Large-Scale Signaling Networks from PPI and RNAi Data Using Linear Programming.
IEEE/ACM Transaction on Computational Biology and Bioinformatics, 2013, 10 (4), 869-883.

O. Eren Ozsoy, T. Can
Construction of signaling pathways from PPI and RNAi data using Linear Programming.
25th Conference of European Chapter on Combinatorial Optimization ECCO 2012, 2012, Antalya, Turkey.

H.Ogul, C.Beyan, O.Eren, K.Yildiz, T.Ercelebi, B.Sonmez
MicroRNA target recognition from compositional features of aligned microRNA-mRNA duplexes.
Proceedings of International Symposium on Innovations in Intelligent Systems and Applications, 2010, Kayseri, Turkey.

O.Eren, H.Ogul
Automated classification of allergen proteins.
Proceedings of 14th Annual Biomedical Engineering Conference (BIYOMUT'08), 2008, Ankara, Turkey.

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü	☐
Sosyal Bilimler Enstitüsü	☐
Uygulamalı Matematik Enstitüsü	☐
Enformatik Enstitüsü	X
Deniz Bilimleri Enstitüsü	☐

YAZARIN

Soyadı :	EREN ÖZSOY
Adı :	ÖYKÜ
Bölümü :	Tıp Bilişimi

TEZİN ADI (İngilizce) : INTEGER LINEAR PROGRAMMING BASED SOLUTIONS FOR CONSTRUCTION OF BIOLOGICAL NETWORKS

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora X

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. X

2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası .... Tarih 06/06/2014