Identification of topological and dynamic properties of biological networks through diverse types of data

A Thesis
Presented to
The Academic Faculty

By

Ugur Guner

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
School of Chemical and Biomolecular
Engineering

Georgia Institute of Technology

August, 2011

Identification of topological and dynamic properties of biological networks through diverse types of data

Approved by:

Dr. Matthew J. Realff, Advisor
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology

Dr. Mark Borodovsky
School of Biology
Georgia Institute of Technology

Dr. Jay H. Lee, Co-advisor
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology

Dr. Daniel Ziemek
Center of Emphasis
Pfizer

Dr. Andreas Bommarius
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology

Date Approved, May 2011

==========

To the unconditional love of my parents and siblings

ACKNOWLEDGEMENTS

First I would like to acknowledge my thesis advisors, Jay H. Lee and Dr. Matthew J. Realff. Their collaboration on Systems Biology gave me the unique opportunity to explore this challenging interdisciplinary area. Their determination, guidance and persistence has inspired me to become who I am today as a professional. Therefore, I thank them for trusting me and guiding me in this long journey of PhD.

I also need to acknowledge for my group members. Wee Chin Wong has helped me a lot with my research ranging from understanding important concepts to guiding my career goals. He has also been a great friend and mentor. I also would like to thank Farminder Anand for his support. With his unique working style and intelligence he motivated me in many aspects of professional life. Prabuddha Bansal has always been there to help me whenever I need most. He has been first person to consult in every problem I faced. It was a great pleasure to work with these three people. They made these long years full of fun and great memories. I also would like to thank other group members Kevin Yeh, Rakshita Agrawal and Nikolaos Pratikakis, Anshul Dubey, Tina Tosukhowong and Jihoon Lee.

Moving on I would like to thank my supervisor Dmitriy Leyfer in my internship during summer of 2009. He has been a great supervisor and friend. He has helped me a lot with Biology and motivated me in my research. He set a great example to me as a hard-working and enthusiastic scientist.

I need to acknowledge my committee Member Daniel Ziemek who also was my supervisor in summer 2010. I learned a lot from him on programming and computational pharmaceutical research. He is a conscientious leader mindful of his team. It was a pleasure to work in his team. I would like to thank him specially as he helped me with my foolish problems and never ignored any request of mine.

I would like to thank my other committee members Dr. Andreas Bommarius and Dr. Mark Borodovsky for their constructive inputs during my research.

Finally, I would like to acknowledge the most important people in my life; my parents. They have been always supportive and they were with me when writing this thesis. My mother sacrificed a lot for me and her great cooking give me the strength to write this thesis at the cost of gaining 20 lbs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AA	Adamic-Adar
AERS	Adverse Event Reporting System
BP	Bipartite
CTLS	Constrained Total Least Squares
CN	Common Neighbors
DNA	Deoxyribonucleic Acid
JAC	Jaccard
KEGG	Kyoto Encyclopedia of Genes and Genomes
MGI	Mouse Genome Informatics
mRNA	Messenger ribonucleic acid
ROC	Receiver Operating Characteristics
RNA	Ribonucleic Acid
SIDER	Side Effect Resource
SR	Spontaneous Reports
TLS	Total Least Squares

SUMMARY

It is becoming increasingly important to understand biological networks in order to understand complex diseases, identify novel, safer protein targets for therapies and design efficient drugs. 'Systems biology' has emerged as a discipline to uncover biological networks through genomic data. Computational methods for identifying these networks become immensely important and have been growing in number in parallel to increasing amount of genomic data under the discipline of 'Systems Biology'.

In this thesis we introduced novel computational methods for identifying topological and dynamic properties of biological networks. Biological data is available in various forms. Experimental data on the interactions between biological components provides a connectivity map of the system as a network of interactions and time series or steady state experiments on concentrations or activity levels of biological constituents will give a dynamic picture of the web of these interactions. Biological data is scarce usually relative to the number of components in the networks and subject to high levels of noise. The data is available from various resources however it can have missing information and inconsistencies. Hence it is critical to design intelligent computational methods that can incorporate data from different resources while considering noise component.

This thesis is organized as follows; Chapter 1 and 2 will introduce the basic concepts for biological network types. Chapter 2 will give a background on biochemical network identification data types and computational approaches for reverse engineering of these networks. Chapter 3 will introduce our novel constrained total least squares

approach for recovering network topology and dynamics through noisy measurements. We proved our method to be superior over existing reverse engineering methods. Chapter 4 is an extension of chapter 3 where a Bayesian parameter estimation algorithm is presented that is capable of incorporating noisy time series and prior information for the connectivity of network. The quality of prior information is critical to be able to infer dynamics of the networks. The major drawback of prior connectivity data is the presence of false negatives, missing links. Hence, powerful link prediction methods are necessary to be able to identify missing links. At this junction a novel link prediction method is introduced in Chapter 5. This method is capable of predicting missing links in a connectivity data. An application of this method on *protein-protein* association data from a literature mining database will be demonstrated. In chapter 6 a further extension into link prediction applications will be given. An interesting application of these methods is the drug adverse effect prediction. Adverse effects are the major reason for the failure of drugs in pharmaceutical industry, therefore it is very important to identify potential toxicity risks in the early drug development process. Motivated by this chapter 6 introduces our computational framework that integrates *drug-target*, *drug-side effect*, *pathway-target* and *mouse phenotype-mouse genes* data to predict side effects. Chapter 7 will give the significant findings and overall achievements of the thesis. Subsequent steps will be suggested that can follow the work presented here to improve network prediction methods.

CHAPTER 1

INTRODUCTION

Most biological functions arise from complex interactions between cell's numerous components, such as, DNA, RNAs. metabolites and proteins. These interactions form complex networks involving thousands of genes, proteins and metabolites. Understanding these networks helps scientists shed light on the complex diseases such as cancer and diabetes as well as control and manipulate biological functions in living organisms. Two important components of a network are the topology and dynamics. Topology refers to wiring diagram of the network, in other words it is the connectivity in a network. Dynamics of the network is the quantification of the connections and time course response of the networks. Many diseases are due to interaction of complex networks from different tissue and organ levels. It is important to understand both topology and dynamics of these networks to be able to identify novel targets for interventions that may help prevent or cure the diseases. A major challenge in biology is to map out and model the connectivity and dynamical properties of these networks.

Motivated by this in this research we developed computational approaches for identifying biological networks. The goals in this thesis can be stated in two ways; Predicting network topology and dynamics to understand complex machinery of biology and finding missing and significant links that have various applications in getting a better picture of network wiring. Following sections will give an introduction to biological

networks, biological data types, important computational aspects and applications for reverse engineering.

1.1 Types of Biological Networks

At a highly abstract level the components of a living organism can be reduced to a series of nodes that are connected to each other by links, with each link representing the associations between two components. These associations can be in the form of binding of one component to the other thereby affecting its function. In gene regulatory networks specialized proteins bind to genes to modulate their expression level. Drug target networks can be another example for this kind of association where a link represents binding between a drug and a target. In metabolic networks a component can catalyze reactions where each reaction and catalysis action of an enzyme on this reaction can be represented as a link in the network. The notion of a biological network can sometimes be extended for defining more abstract associations between biologically relevant components where the exact mechanisms are not yet known. Drug side effect networks can be given as an example for this type of networks. There are different kinds of biological networks that take part in different functions of the living organisms at different levels. In the next few paragraphs we will give brief information on each type of networks.

Metabolic pathway networks are the series of reactions that share reactants and products. Enzymes catalyze these reactions and often require dietary minerals and vitamins and other co-factors in order to function properly. Because of the large number of metabolites involved these networks can be quite complex and numerous pathways

may exist within these networks. A substrate enters a metabolic network leading to a series of reactions and the production of intermediates and final molecules. This final product molecule may be used as a substrate for another network.

Protein-Protein interaction occurs when two proteins bind together often to carry out a biological function. These interactions are at the core of entire interactomics system of many living organisms. Signals from exterior of a cell are conducted to the cell trough through protein-protein interactions. This is also called signal transduction. Signal transduction networks refer to the interactions where an extracellular molecule activates a membrane protein that in turn alters a cascade of intracellular proteins creating a response. These molecular cascades detect, amplify and integrate diverse external signals to generate responses such as enzyme activity, gene expression, or ion-channel activity. Diseases may be due to malfunction of one or more signal transduction networks.

Transcriptional regulation is the most common way in which cells use these interaction webs to perform its functions. This is achieved by cell's specialized proteins called transcription factors. A transcription factor or a combination of transcription factor can bind to an upstream of a gene and modify that gene's output. A gene's product is the mRNA (Messenger RNA) and mRNAs are translated into proteins. Therefore, it is a higher level control mechanism in cell that can account for functional diversity of cells and it can control metabolic, transduction and protein interaction networks.

In addition to these networks, more abstract level of biological associations is studied as networks. For example, *drug – side effect* networks are one of them. A side effect is an effect, whether therapeutic or adverse, that is secondary to the one intended. Side effects are also called adverse effects. An adverse effect can also apply to

unintended but beneficial consequences of a drug though it commonly refers to its toxic results. Drug-target networks are another interaction webs where potential target proteins of drugs are collected and analyzed. In the pharmaceutical industry it is often important to analyze combination of these networks in addition to the molecular networks to identify novel targets and predict possible toxicity of developed drugs.

1.2 Computational Methods for Biological Network Identification

System biology has emerged as a discipline to understand and reverse engineer biological networks through biological data at a systems thinking level. It uses methods from mathematics, statistics, biology, information technology and the technology of high performance computing and the manipulation of large datasets. This discipline is developing in parallel with the amount and quality of biological data becoming available. The approaches that are employed changes with respect to the amount of information available and the resolution and accuracy of prediction needed. A detailed modeling effort can include the differential equation representation of the network in continuous time domain. These methods usually target small portions of the network and they require topology information. Other methods include causal networks such as Boolean and Bayesian networks. Sometimes biological networks are simplified to bipartite network representations in the case of *drug-target* and *drug-side effect* networks.

Recently, large amounts of diverse types of genomic data are obtained to shed light on these networks. For transcription regulation, e.g., DNA sequence data, micro-array gene expression data, protein-DNA binding data are the major data sources. For metabolic networks metabolomics data is predominantly used. Signal transduction

network inference methods utilize protein-protein interaction data as well as protein phosphorylation data. There are several information resources available for *drug-target, drug-side effect,* as well as *pathway-target* networks. These resources are crucial for understanding diseases and designing novel drugs. Next Chapter will give a background on network inference efforts and data types.

CHAPTER 2

BACKGROUND

This section starts with the definition of transcriptional regulation and elaborates on the data types and methods for inference of regulatory networks. Microarray technology is an experimental method where thousands of gene expression levels are measured simultaneously. This technology has created means to generate abundant data to shed light on gene networks [4]. Therefore, many efforts of biological network applications particularly focused on inferring gene regulatory networks from microarray data. The noise levels in microarray data is high [5] and only small portion of these studies concentrated on addressing this problem [62]. Hence, in chapter 3 we will develop identification methods of regulatory networks under noisy measurements.

The second part of this chapter introduces the link prediction methods in biological networks. These methods are essential to uncover hidden connections and can be potentially used in improving the topology of networks. As biological networks are complex, connectivity information is crucial for reverse engineering methods. However this information can have noise in the form of inconsistencies, false links and missing links. Link prediction methods are usually based on scoring a pattern or a link in the topology of the network and they can potentially be a remedy to improve the topology of the network. Common link prediction methods are based on the notion that nodes they share common features are more likely to be linked. These features are usually based on common number of nodes that are shared by pair of nodes, shortest path between the pair of nodes or topological features of the nodes in the neighborhood of the node pairs.

However there is no probabilistic approach that considers degree distribution of the nodes in a systematic and comprehensive way in the network when scoring the local structures in the context of biological networks. In chapter 5 we introduced a score function based on a probabilistic framework to quantify links in biological networks.

The last part of this chapter provides a background on *drug-side effect* network prediction methods as well as *drug-target* networks. These networks are commonly used by pharmaceutical industry to bring novel protein targets for drugs or to predict potential toxicity risks involved with them.

2.1 Transcriptional Regulation

Two foundational concepts in molecular biology are: (i) genes, the fundamental units of heredity, are encoded as sequences of chemical bases in DNA and (ii) a gene is expressed when its DNA sequence is transcribed into an RNA intermediate and then translated into proteins. Proteins, in turn, perform regulatory, catalytic, mechanical, and electrical functions [1].

Gene expression is the process by which cells produce proteins from the information encoded in DNA [1]. The information flow from DNA to proteins occur in following steps: Specialized proteins transcribe a region of DNA (gene) into a Messenger RNA molecule, and RNA molecule is translated into a polypeptide chain, and polypeptides fold into three dimensional structures and modified with additional proteins to become biochemically active [1].

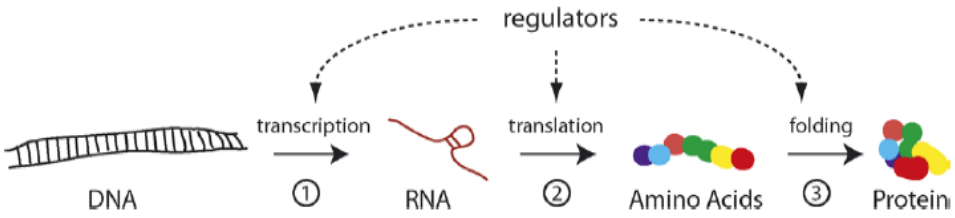


Figure 2.1: Gene expression is shown as a three step process [2].

Initiation of transcription in eukaryotes is a complicated process that depends on the binding of transcription factors (TF) to the promoter region as well as the action of RNA polymerase complex to the transcription start site. Transcription occurs as a result of combinatorial and cooperative binding of multiple factors on the same promoter region. TFs regulate the transcription process either positively of negatively. Occupancy of a transcription region by a TF is a necessary but not sufficient condition for that gene to be activated or inhibited.

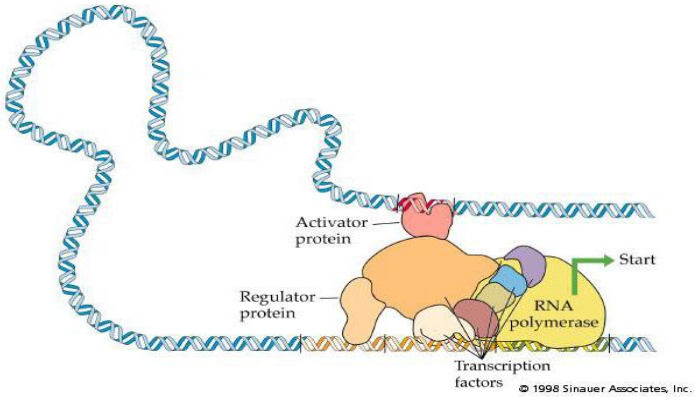


Figure 2.2: Transcription process by transcription factor and RNA polymerase action [67].

The process of gene expression allows for control at many levels. Changing the rate of transcription into a RNA molecule, stalling the process of its translation into protein, and cleaving the final product protein into pieces can serve as mechanism for

controlling the gene expression. Cells have evolved to use all of these mechanisms, but regulation of the transcription process is the most common. Transcription factors are the proteins that can bind to the DNA in order to regulate this process [1].

Genes, proteins, and metabolites can regulate one another in various ways. Regulatory proteins bind to a DNA molecule to affect the transcription of genes. Proteins can also combine to form multi-protein complexes that can take part in various functions in regulation, for example unzipping a DNA molecule or cleaving an RNA molecule. Metabolites can also attach proteins to alter their activity level [1].

2.2 Data types on gene networks

Previous section gave a description of gene regulatory networks. This section will introduce types of experimental data that are used for understanding gene networks. Microarray and protein-DNA binding experiments are two major data sources for regulatory networks.

2.2.1 Micro-array data

Micro-array is a chemical assay that uses fluorescent labeling to measure the RNA concentrations of all the genes in a cell in single experiment. A micro-array includes thousands of distinct chemical probes, each specific for a gene's RNA, arranged on silicon or glass substrate in the size of a coin [1]. Total RNA is fluorescently-labeled and washed over the chip and the chip is illuminated. Each probe will fluorescence according to how much labeled-RNA is bound. Therefore, fluorescence pattern on the chip provides a global picture of gene expressions for a given experiment. Unlike physical binding interactions between molecules, micro-array experiments only provide indirect evidence for gene interactions [1].

DNA micro-arrays have been used to measure mRNA abundance for essentially all protein-coding genes in the genome under a large number of conditions. These data provide measurement of expression levels of thousands of genes simultaneously, and promise to be a very important tool in revealing the gene networks.

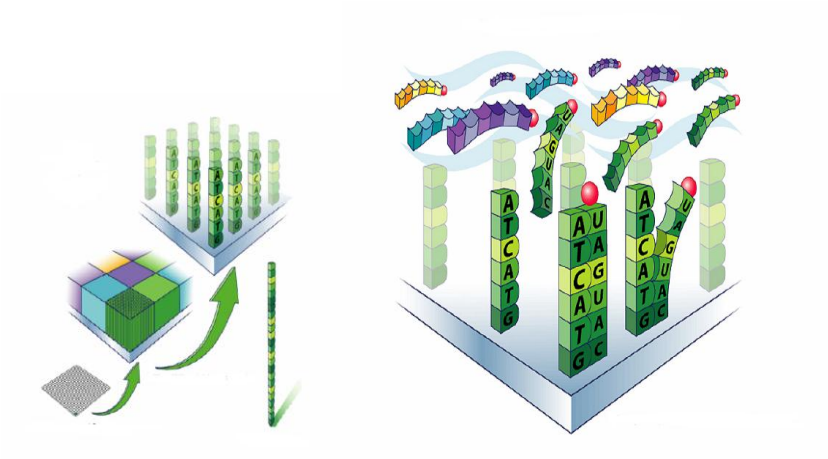


Figure 2.3: A DNA micro-array is a collection of DNA fragments attached to a solid surface which serves probes for specific genes[68]

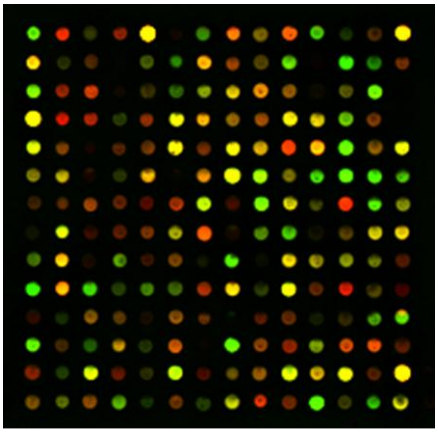


Figure 2.4: A DNA micro-array data is shown. Each dot corresponds to the mRNA level for a specific gene with respect to a reference level. Red color shows an increase in expression level, green color indicates a decrease, and black color corresponds to an undetectable change [69]**.**

There are two kinds of micro-array data obtained to infer gene regulatory networks: Steady-state measurements and time-course data.In steady-state data, upon a single gene knockout, over expression or simultaneous perturbation of a group of genes, the system reaches to a new steady state. The gene expression levels are measured against the reference steady state values. In algorithms using steady state data, a major drawback is the necessity of keeping the perturbation levels sufficiently low to render final results reliable.

In time-series data, the system is perturbed and measurements of gene expression levels with respect to unperturbed level are obtained in successive time points. Compared to steady-state data, time series offer rich opportunities for understanding the dynamics of biological processes [6].

Limitations of micro-array data

Micro-array data is scarce for the gene regulatory networks inference methods which require large data set. These data are expensive to obtain and include experimental noise up to 15% . In other words, micro-array data are typically noisy, high dimensional, and significantly undersampled [2].

2.2.2 Binding Data

Chromatin immuno-precipitation (ChIP) is an experimental method in molecular biology to quantify the occupancy of upstream non-coding regions by transcription factors. In budding Yeast Saccharomyces Cerevisae, ChIP has been used to globally map the binding sites over a hundred transcription factors [7].

Limitations of Binding data

Occupancy of the promoter region of a gene by a transcription factor protein is necessary but not a sufficient condition for a gene to be regulated by it [7]. As a result, quantification of genome wide transcription binding patterns by ChIP experiments alone can only indicate the potential for a gene to be regulated by a given TF [7]. Independent information will be required to establish that the gene is indeed a functional target. Binding of a TF on a promoter region of a gene is leads to observation of a link between for that TF-gene pair from the perspective of a network. However as binding of a TF protein to a gene doesn't necessarily mean regulation of this gene by TF protein this link is considered as a false link. Binding data has been obtained for a limited number of organisms. So far, protein-DNA interaction quantification has been studied for K12 E.Coli [8], and for yeast.

The microarray and protein-DNA binding data may not be sufficient for a detailed reverse engineering of the gene networks due to the high number of TF proteins, genes and mechanisms involved in these networks. Therefore researchers resort to certain abstractions in inference algorithms. Next section will give the details of such simplifications and assumptions.

2.3 Main assumptions in gene network inference algorithms

Gene networks contain the complex and nonlinear interactions of proteins, metabolites and genes. However, micro-array data provides only mRNA concentration information. Protein and metabolite concentration (proteomics and metabolomics) data are still difficult to obtain. As a result, network inference methods based on micro-array data can only capture the regulation dynamics in an indirect manner [10, 1]. In other words, all these techniques make the implicit assumption that the expression of the transcription

factor genes can be used as a proxy for the true transcription factor activity-the concentration of the protein in the form of that is able to bind and induce/repress transcription [10]. Nevertheless, algorithms based on revealing gene to gene relations, provides a global view of gene regulation [2].

Another challenge is the scarce nature of micro-array data. The reverse engineering approach requires large amounts data and extensive computational resources [11]. Typically, there are a huge number of network topologies that fit a given set of expression data [11]. To circumvent this problem, many research efforts have focused on clustering,.i.e grouping genes into hierarchical functional units based on correlations in expressional patterns [12, 13, 14]. A fundamental shortcoming of the clustering approach is that they are based on the assumptions that: gene regulatory networks are hierarchical in the structure and genes performing related biological functions exhibit similar expression patterns. These assumptions may not always be valid [15].Due to data scarcity, several studies have targeted small networks using many different frameworks.

The most popular way to get around the scarce data problem is the assumption of sparse connectivity in gene networks. This assumption greatly decreases the number of parameters to be inferred thus let researches tackle otherwise underdetermined problem [15]. However, this comes at the cost of computational complexity; a heuristic or Monte Carlo search for the best combination of regulators of each transcript is required.

One of the main challenges among gene regulation inference is the validation of the method. Upon application of methods on experimental data, researchers try to validate their method by delving into biological literature. While this approach can give some confidence in the algorithm, it does not solve the problem of the lack of knowledge

about the phenomena under study. For this reason, researchers usually apply their algorithms on well studied, usually small size networks.

As a second approach, the algorithms are applied to synthetic data sets. In order to obtain an objective validation for the methods proposed, Mendes et al introduced a nonlinear continuous differential equation model that mimics characteristics of known gene networks as much as possible [16] (See Section C.1). Researchers measure the performance of their algorithms against synthetic data by various ways. The two common ways of measuring performance is the coverage of connections (true positives) and false positives in recovered network. True positive ratio is the proportion of the number of correct connections identified to the total number of connections in the true network. False positive ratio is the proportion of incorrect connections in recovered model to the total number of recovered connections. [17].

2.4 Reverse engineering strategies for gene networks

Engineers and scientists have previously developed reverse engineering techniques in the fields of computer science, engineering, and statistics, which are respectively called machine learning, system identification and statistical learning [75]. With the emergence of DNA micro-array data, researches proposed many approaches to reverse-engineer the mechanism of transcriptional regulation [2].

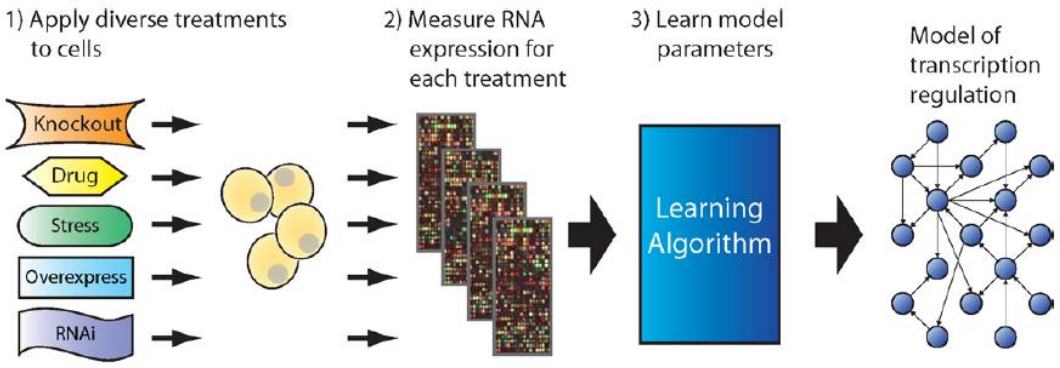


Figure 2.5: General Strategy for reverse engineering transcription control systems [2].

Several approaches have been proposed to reconstruct the gene regulatory network from the data. One can broadly group these methods in two categories: Stochastic and deterministic approaches.

2.4.1.1 Stochastic Approaches

Many stochastic approaches have been proposed to reverse-engineering of gene networks. Among these, Bayesian Networks and Dynamic Bayesian networks are the most popular ones.

2.4.1.2 *Bayesian networks*

Bayesian network is one of the popular frameworks that have been applied successfully for gene networks. Bayesian network methods were first proposed by Friedman et al. [18], and further developed by Hartemink et al [19].

A Bayesian network is a graphical model that represents the causal relationship in random variables [70]. In the context of gene networks, each gene represents a node in the graph and expression level of each gene is represented as a continuous random variable. The probability density function for that random variable is assumed to be

conditionally dependent on the expression levels of other genes. In a Bayesian framework, the task to infer the network is to identify the weight of these dependencies [1] (Section A.1).

The probabilistic structure of a Bayesian network enables straight-forward incorporation of prior knowledge via Bayes rule, thus one can complement the micro-array data with prior information.

The network structure is usually determined using heuristic search, such as a greedy-hill climbing approach or Markov-Chain Monte Carlo method [76]. For each network structure searched, algorithms find the maximum likelihood parameters and compute a score for each structure using Bayes rule. The Bayesian network approach typically requires a vast data set and it can-not handle cycles in the network [1].

2.4.1.3 Dynamic Bayesian networks

Dynamic Bayesian networks represent the dependency in gene expression levels based on time-course data. The directed graph of the causal relationship among N random variables is then constructed by estimating the bipartite graph of transcription factor proteins and genes. (See Section A.2). A Hidden Markov model can be considered as the simplest dynamic Bayesian network [65].

Bayesian network is a quite powerful method to model networks however they require a vast amount of data. Data is scarce in biological networks [2] and subject to high levels of noise [66]. Therefore, Bayesian networks can only target small networks whereas linear and deterministic models are capable of identifying larger networks. The next section will introduce deterministic methods for the network inference problem.

2.4.2 Deterministic Approaches

Most studied modeling schemes for gene networks are deterministic approaches. The methods of ordinary differential equations and Boolean methods are two of the most popular deterministic methods employed for the inference of gene networks.

2.4.2.1 *System of differential equations*

The most common approach to the modeling of dynamics of gene regulation is to view a gene regulatory network as a biochemical network of gene products, typically mRNA and proteins, and to describe their rate of changes through a system of ordinary differential equations[6, 15, 21, 22, 24, 28]. Therefore, the modeling framework is that of continuous time and, deterministic dynamical system often cast as ordinary differential equations (ODEs). (See Section B.1).

Linear Methods

In many studies using ODEs, the main underlying assumption is that the system is operating near a steady state, so that dynamics can be approximated by linear differential equations (See Section B.1).Several groups have applied linear ODE models to infer gene networks [15, 21, 22]. These methods usually require a certain degree of prior information. One of popular approach is to infer networks using steady state data on linear models [22, 23, 6]. Gardner et al. [22] proposed NIR (Network Identification by Multiple Regression) algorithm to reverse engineering a SOS network (DNA damage response pathway) using linear ODE model structure and steady state measurements. In the study of Gardner et al. [22], experimental data are collected by artificially increasing the level of RNA for individual genes in the network. In each perturbation the system re-

settles to a new steady-state. The response of the system is calculated by the shift of the state variable from the initially observed steady-state. Though steady state data models can shed light on the structure of the network, they cannot give details on the dynamics of the networks.

In Bansal et al [21], an algorithm to infer gene networks is proposed. They used time series data instead of steady state perturbations. They perturb a gene of interest and subsequently measured the gene expression profiles at multiple time points. However their model didn't consider any noise element in their study. In their method they identified the parameters (regulatory strengths in the network) in the discrete domain and transform estimated parameters into the continuous domain. Noise in the data will lead to noisy parameter estimation in discrete domain and this error is further amplified when the system is transformed into the continuous domain. Therefore it is crucial to consider noise and any possible correlation structure in it.

S-System based models

Genetic networks are complex nonlinear systems. The S-System is one of the best formalisms to estimate mechanism of interactions in gene regulation. It is one of the most well studied methods [24-28]. The structure of S-System is rich enough to capture many relevant biological dynamics.

The S-system belongs to the type of power-law formalism because it is based on a particular type of ordinary differential equation in which the component processes are characterized by power-law functions [28, 29]. (See Section B.2).

The major disadvantage of S-System formalism is that it requires a large number of parameters to be estimated. Thus, this formalism is demanding in terms of data. In

gene regulatory network applications, where the data is highly limited, only small size networks can be the target of estimation by S-Systems.

2.4.2.2 Boolean Networks

First proposed by Kaufman (1969), Boolean networks represents gene networks as logical switching networks; it is a coarse grain approximation of the real network. In this model, time is taken discrete and gene expression is discretized into two qualitative states, present or absent. Several algorithms have been proposed for inferring Boolean networks [30, 31]. The goal is to construct an algorithm to find an optimal Boolean function for the given state data. A sparseness assumption is made to make the problem tractable under scarce data. The number of inputs to a function is limited to a certain degree. The main disadvantage of the Boolean algorithm is that it loses large amounts of information as the expression levels are reduced to only ON/OFF.

True behavior of biological networks are highly nonlinear, therefore nonlinear ODE approaches such as S-Systems can model networks accurately. The major issues with nonlinear modeling efforts are the number of parameters to be estimated ,data scarcity compared to network complexity, and the difficult of constraining model behavior outside the range of measured data so that reasonable generalization error results. On the other end of spectrum sits Boolean methods. They are very simplistic representation of network interactions. Though they can give an idea about the initial picture of the networks they are far from providing dynamic details of the system. Linear continuous and discrete time models can give enough details with reasonable data requirements. However the majority of these models do not consider the noise component in the biological measurements. Measurement noise can lead to large errors in the

topology and parameter estimation. Furthermore, these methods assumed known topologies or didn't use topological information available from different resources.

2.4.3 Methods integrating diverse types of genome data

Some research has been directed to reveal the transcription factor activities using micro-array data alone or with binding data [7, 10, 32- 40]. Liao et al. [32] proposed Network Component Analysis (NCA) to infer transcription factor activities, which can incorporate prior knowledge. However, the NCA method imposes strong restrictions on the network topologies. Alter and Golub [40] introduced an approach for integrating binding and micro-array data using pseudo-inverse projection.

Boulesteix and Strimmer [35] proposed a statistical approach based on partial least squares regression to infer the true transcription factor activities from a combination of mRNA expression and DNA-protein binding data.

Gao et al [7] presented MA-Networker algorithm that combines micro-array and binding data and infer the activity of transcription factors using multivariate regression. Brynildsen et al. [34] proposed a Gibbs sampling algorithm combining two types of data which concentrates on the instances of agreement of both data. By doing this, they aimed at minimizing the effects of experimental noise in the data, and lack of correlation between binding and regulation.

Sabatti and James [37] introduced an algorithm using sequence and expression data to infer transcription factor activities. They used sequence data to define a prior distribution on the topology of the network and expression array data allows them to identify which of the potential binding sites are actually used by regulatory proteins and their activation profile. To carry out their reconstruction algorithm, they proposed using a

Bayesian framework to identify unknowns in a linear model [37]. In matrix notation, their model is represented as follows;

$$E = A \times P + \Gamma \quad \textbf{(2.1)}$$

where E represents the micro-array data. In $E = \{ e_{ij} \}^{N \times M}$ row indices correspond to the gene numbers and each column represents an experiment at a different time point. N is the number of genes and M is the number of experiments. $A = \{ a_{ij} \}^{N \times L}$ is the regulatory strength matrix denoting the effect of TF proteins on gene expression. L is the number of transcriptional proteins. Each element, a_{ij} shows the regulatory effect of the j^{th} TF protein on the expression of the i^{th} gene (mRNA level). A is usually a tall matrix as the number of TF proteins is smaller than number of genes. ($L < N$). It is unknown along with the $P = \{ p_{jt} \}^{P \times M}$ matrix representing TF levels at different time points. $\Gamma = \{ \gamma_{it} \}^{N \times M}$ captures the measurement error in each gene expression level. Each γ_{it} s assumed to be i.i.d according to the Gaussian distribution, $N = (0, \sigma_i^2)$.

$Z = \{ z_{ij} \}^{N \times L}$ is a binary matrix with element z_{ij} is 1 if j^{th} TF factor is regulating i^{th} gene and zero otherwise.

The Bayesian reconstruction framework becomes as follows;

$$\Pr(Z, A, P, \sigma^2 \mid E) \propto \Pr(E \mid Z, A, P, \sigma^2) \Pr(Z, A, P, \sigma^2)$$

$$\propto \prod_{i=1}^{N}\left(\frac{1}{\sigma_i}\right)^{M} \exp\left(-\sum_{i=1}^{N}\frac{1}{2\sigma_i^2}\sum_{t=1}^{M}\left(e_{it}-\sum_{j=1}^{N}a_{ij}p_{jt}\right)^2\right)$$
$$\times\left[\prod_{i=1}^{N}\left(\frac{1}{\sigma_i^2}\right)^{\alpha_i-1}\exp\left(\frac{\beta_i}{2\sigma_i^2}\right)\right]\left[\prod \pi^{i(z^i)}(1-\pi_i)^{1-z^i}\right] \qquad \textbf{(2.2)}$$
$$\times\exp\left\{-\frac{1}{2}\left(\sum_{i=1}^{N}\frac{a^i\left[z^i\right]'a^i\left[z^i\right]}{\sigma_a^2}+\sum_{t=1}^{M}\frac{p_t' p_t}{\sigma_p^2}\right)\right\}$$

In this reconstruction framework, the unknowns are A the regulatory strength matrix, Z the binary version of A and P is the TF protein levels in different time points and σ^2 is the variance vector storing the variances of measurements of each gene's expression level. a^i and z^i represent the i^{th} column of A and Z matrices respectively. p_t is the vector of elements in P matrix in row number, t. The vector form is adopted in order to have a compact representation for the expressions. The posterior distributions of Z, A , P and σ^2 are obtained according to the Bayesian rule shown in equation (2.2). a_{ij}, p_{jt} and σ_i^2 assumed to be mutually independent with following distributions;

$$\Pr\left(a_{ij} \mid z_{ij}=1\right)\approx N\left(0,\sigma_a^2\right) \qquad p_{ij}\approx N\left(0,\sigma_p^2\right)$$
$$\Pr\left(z_{ij}=1\right)=\pi_{ij} \qquad \frac{1}{\sigma_i^2}\approx Gamma\left(\alpha_i,\beta_i\right) \qquad \textbf{(2.3)}$$

In this equation, α_i, β_i, σ_a^2 , σ_p^2 are assumed to be hyper parameters. In the presence of a regulatory relation ($z_{ij}=1$), the regulatory strength term, a_{ij} has Gaussian distribution with zero mean and a variance of σ_a^2 . Zero mean indicates that there is no information on the regulatory strength a priori. Similarly, p_{ij} has Gaussian distribution

with zero mean with a variance of σ_p^2 . The distribution on each element of binary matrix, Z is considered to be binomial with parameter, π_{ij}. This parameter is obtained through sequence information. The difference between model and measurement is also assumed to have Gaussian distribution with zero mean shown in the first term in the right hand side of equation (2.3).

Sabbatti and James used a collapsed Gibbs Sampling algorithm to solve the problem sequentially and applied their methodology to the E.Coli expression data. Sun et al. [38] introduced a Bayesian error analysis model to integrate binding and gene expression data to reconstruct transcriptional regulatory network. In their algorithm, they accounted for measurement errors in both types of data by considering these within a Bayesian model framework. Transcriptional factor activities and their effect on genes defined as parameters and along with unknowns defined for error models are merged in this framework [38].

Sun et al's method is a slightly different version of the approach employed by Sabatti and James [37]. The major contribution in the paper is modeling transcription process as a set of biochemical reactions and ending up with the identical linear model between expression levels and TF protein that was adopted in [37] (Equation (2.1)). They also assumed measurement noise has independent structure and Gaussian distribution where the variances are among the unknowns in the inference algorithm. Unlike in [37], instead of Gamma distribution, they assumed an inverse gamma distribution on the variance of measurement noise.

Their model is identical to equation (2.1). Instead of sequence data, they used protein-DNA binding data. They considered transcriptional regulation networks

consisting of two components; a binary connectivity matrix, $R = \{r_{ij}\}$ and regulatory strength matrix, $A = \{a_{ij}\}$ (equation (2.1)). R is binary version of A and essentially shows the presence and absence of the edges in the network where nodes representing gene expression and TF levels. Through protein-DNA binding data, the observed binary binding information $Z = \{z_{ij}\}$ is obtained. This Z matrix is analogous to the Z matrix in [37], but note that Sabatti and James obtained this observation through sequence analysis instead of protein-DNA binding data.

As Z has observational errors (false positives and false negatives), they introduced an misclassification model between Z and R (observed and real binary binding matrices , respectively). Their misclassification model is as follows;

$$\Pr(w_{ij}=1 \mid r_{ij}=1) = p \qquad \Pr(w_{ij}=0 \mid r_{ij}=1) = 1-p$$

$$\Pr(w_{ij}=0 \mid r_{ij}=0) = q \qquad \Pr(w_{ij}=0 \mid r_{ij}=1) = 1-q \qquad \textbf{(2.4)}$$

where p and q are true positive and true negative rates respectively. They obtained relative binding intensity data through protein-DNA binding experiments and represented it as, $B = \{b_{ij}\}$. The regulatory strength, a_{ij} is approximated by using binary binding information, r_{ij} and binding intensity, b_{ij} according to the following formula;

$$a_{ij} = b_{ij} r_{ij} \qquad \textbf{(2.5)}$$

In [37], there was no binding intensity information, and regulatory strengths a_{ij} have Gaussian distributions with zero mean (Equation (2.3)), however in this study , binding intensity is utilized to obtain an approximation for a_{ij} through equation (2.5).

MCMC is employed as a solution strategy similar to [37]. The method is applied to the yeast cell data to illustrate its application.

In both of these studies [37, 38], the nodes of the gene network are the gene expression levels and TF protein levels. TF protein levels are treated as unknown hidden variables and expression levels are the variables observed through time series micro-array data.

As genomic data is limited, corrupted by high levels of noise and systems are complex, prediction TF protein levels for all time points in addition to regulatory strengths may result in inefficient reconstruction. TF protein levels as hidden variables will provide additional information on networks, but comes at the cost of higher quality and quantity of data.

In another study Bernard et al [39] presented a dynamic Bayesian method for jointly learning models of transcriptional regulatory network from expression and binding data. They incorporated expression data in likelihood term and binding data is modeled in a probabilistic manner to serve as a prior. Dynamic Bayesian networks is a class of Bayesian network model that permit cyclic structures like regulatory feedback loops and have been used to analyze the time series data in the context of transcriptional regulation [20]. In the process of learning dynamic Bayesian networks, most probable topology is determined using Bayes rule given the time-series data. (For detailed discussion of Dynamic Bayesian Networks , see section A.2 in appendix). The prior on structures is usually assumed to be non-informative. In [39], binding location data is used and converted to a probabilistic model to serve as an informative prior. Location data provides evidence as to whether a regulatory relationship exists. This evidence is

signified through p-test. In their probabilistic informative prior model, the more significant the location data (the lower the p-value), the more likely the edge is to be included). They employed a function that maps p-values to corresponding probabilities of edges being present in the topology, G of the gene network. In this network, the vertices (nodes) are gene expression levels and edges denote the regulatory relationships between the genes. The p-value is defined as a random variable $\Pr_i$ in the interval, $[0,1]$. It is assumed to be exponentially distributed if $E_i \in G$, and uniformly distributed if E_i absent from G.

$$\Pr_\lambda\left(\Pr_i = p \mid E_i \in G\right) = \frac{\lambda e^{-\lambda p}}{1-e^{-\lambda}} \qquad \textbf{(2.6)}$$

The probability of $E_i \in G$ is taken as β, $\Pr\left(E_i \in G\right) = \beta$. Using Bayes rule, the probability of edge E_i being present in G after observing the corresponding p-value is shown as follows:

$$\Pr_\lambda\left(E_i \in G \mid \Pr_i = p\right) = \frac{\lambda e^{-\lambda p}\beta}{\lambda e^{-\lambda p}\beta + \left(1-e^{-\lambda}\right)(1-\beta)} \qquad \textbf{(2.7)}$$

They applied their framework on both simulated and experimental data and demonstrate that regulatory networks recovered through joint learning algorithms from multiple types of data are more accurate than those reconstructed from each type of data alone. However Bayesian networks are probabilistic frameworks and they require vast amounts of data.

2.5 Link Prediction Methods in Biological Networks

With the advent of new technologies in biology, there are many resources for data types for biological networks. Some of this data represents the connectivity structure of the biological networks. These networks can be listed as protein-protein interaction, regulatory and metabolic networks. For example, Text mining data identifies biological relationships based on co-occurrence of *gene/protein/drug/disease* terms in the abstracts of scientific publications [48]. Protein-DNA binding data is another resource for connectivity of regulatory networks. PPI (*protein – protein* interaction) databases are also quite commonly used for inference of networks. Recently link prediction has attracted increasing attention from network from computer scientists and physicists [57-61]. The link prediction problem can be categorized into two groups. The first category concerns predicting links that exist yet are unknown. Biological network connectivity prediction can be classified under this category. The other link prediction category is predicting connections that will become available in the future for networks evolving in time such as social networks [49]. For biological networks the discovery of a link is costly and time consuming therefore it is logical to make prediction using existing links and focus on verifying these predicted links.

For link prediction first step is to define node similarity in the network depending on the commonly shared features. Lieben-Nowell and Kleinberg [50] compared topology based node similarity indices for social networks. They showed that Common Neighbors (CN) and Adamic Adar (AA) methods are the best in terms of predicting future links in social network examples. Common Neighbors method assumes that two nodes are more likely to form or have a link if they have many common neighbors.

$$Score\,(CN) = |\Gamma(x) \cap \Gamma(y)| \quad \textbf{(2.14)}$$

Here, $\Gamma(x)$ is the set of nodes connected to node-x and similarly $\Gamma(y)$ is the set of neighbors for node-y.

Adamic-Adar (AA) similarity index is a modification of common neighbors where the in-degrees of common neighbors are taken into account. It is represented as follows;

$$Score\,(AA) = \sum_{k \in (\Gamma(x) \cap \Gamma(y))} \frac{1}{\log(k(z))} \quad \textbf{(2.15)}$$

Jaccard is another quantification of similarity index between two nodes in a network. It is simply the number of common neighbors divided by the union set of neighbors of the pair of the nodes. Therefore, it is a function of out-degrees of the node pair as well as number of common neighbors they have.

$$Score\,(J) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \quad \textbf{(2.16)}$$

Common Neighbors method only relies only on the number of commonly shared nodes therefore it doesn't consider degree distribution of the nodes. Jaccard index was originally introduced for comparing the similarity between two sets. It is also commonly used in graph theory. For the Jaccard score out-degree of the nodes are considered in addition to the number of common neighbors. On the other hand, Adamic Adar method is based on both number of commonly shared nodes and natural logarithm of their degree distribution. However this method doesn't take into account the out-degrees of the nodes. To the best of our knowledge there has been no study on p-value based probabilistic approaches for scoring node similarity based on the all three components; out-degree, in-degree and number of commonly shared nodes . We will present such a method in Chapter 5 of this thesis.

2.6 Biological Networks as Bipartite Networks

There are several other biological networks that can be represented in the form of bipartite networks. Among them is Drug-target networks. Identification of drug-target interactions (interactions between drugs and target proteins) is a key area in drug discovery. Both the number of new drugs and targets has remained rather relatively unchanged in the last 20-25 years [51]. Yamanishi et al [52] integrated know drug-target information with target protein sequence data and drug chemical structure. In another study, Campillos et al [53] used side effect similarity between drugs to predict novel targets for drugs. They based their method on the assumption that the drugs that share common side effects are more likely to share targets. They combined drug-target, drug-side effect information with drug chemical similarity. They also validated some of their predictions with experimental results. Drug-target information is available through different databases. KEGG brite [54], BRENDA [55] and Drugbank [56] are among these databases.

Increasing scientific, regulatory and public scrutiny is focused on the obligation of the medical community, pharmaceutical industry and health authorities to ensure that marketed drugs have acceptable benefit-risk profiles. In that regard adverse event prediction methods for drugs become increasingly important. Drug side effects relationships can also be visualized as a bipartite graph. Campillos et al [53] defined a similarity measure for drug-drug pairs for their commonly shared side effects. They also created a drug-side effect database using drug package inserts (SIDER,[53]). Drug package inserts are simply the results of clinical trials where the number of subjects is on

the order of thousands. Another database for drug-adverse event is AERS that is maintained by FDA. The name AERS stands for Adverse Event Reporting System. It is a collection of Spontaneous reports that are product of surveillance of drugs in the post marketing phase. Hence, this database includes exposure of much larger population of patients compared to clinical trial data [74].

It is essential to identify adverse events in the early phases of drug development. Two of these early phases include target discovery and animal models. An important parameter in target discovery is the side effect [71]. Pathways are crucial components in target validation. The knowledge of a pathway allows separate targeting of upstream or downstream targets. Inhibition or modulation of selected targets in the same pathway could lead to the same therapeutic with fewer side effects or better druggability. Furthermore, knowledge of pathways and their relation to each other helps researchers understand side effect profiles [71]. A valuable resource for biological pathways is the KEGG pathway database [54]. This is a collection of manually drawn pathway maps representing the collected knowledge on the molecular interaction and reaction networks [54]. This database can be viewed as a bipartite network of the Pathway-Target associations.

In chapter 6 by integrating *Pathway–Target* relations (ex. KEGG pathway [54]), *Target-Drug* (ex. Drugbank [56] and Brenda [55]) and *Drug –Side Effect* (ex. SIDER,[53] and AERS) we formed a multipartite network of *Pathway-Target-Drug-Side Effect* relations. Integrating these databases and finding significant structures in these resulting networks can serve as a framework to associate side effects with targets and molecular pathways. This framework can be a useful resource for side effect prediction in

the early phases of drug development. To the best of our knowledge there has been no systematic study of integrating these databases on a network framework to find significant *Target-Side Effect* or *Pathway-Side Effect* relations.

Animal models have specific characteristics that mimic human diseases. The technologies for the creation of transgenic animals, where certain genes are either deleted, modulated, or added, have progressed tremendously in the last decade. As a result, the predictive power of animal models for human disease and pharmacology is improving. It is crucial to note that some experts in the pharmaceutical industry and the U.S. Food and Drug Administration (FDA) believe that inadequate animal models, or the lack of animal models altogether, are a major obstacle in drug discovery and development. Pharmaceutical companies have long used model organisms in preclinical efficacy [71]. The laboratory mouse is the premier animal model for understanding the genetic and molecular basis of human biology and disease [72]. MGI database is a comprehensive information source that primarily provides genetic and genomic data to support laboratory mouse a model organism. [73] To achieve this goal, MGI maintains a comprehensive catalog of mouse genes and other genome features and associates these features with orthologous genes in other mammals, human diseases, functional annotation, mouse phenotype descriptions, DNA and protein sequence data and developmental gene expression information.

A valuable information resource that can be obtained from MGI database is the mouse phenotype-mouse gene associations. These relations can be represented as a bipartite network. Combination of '*Mouse Phenotype'*-'*Mouse Gene'* relations and orthologous of mouse genes in humans with target-drug network as well as drug-side

effect network can give a multipartite network of *Mouse Phenotype -Mouse Gene-Human Target-Drug-Side Effect*. In chapter 6 we will aim at finding significant motifs in such networks that can be used as a methodology to associate mouse phenotypes with human side effects. There is no network based method for associating mouse phenotypes with human side effects.

2.7 Discussion

In this chapter a background is given for reverse engineering biological networks. There are two broad groups of models for inferring biological networks; probabilistic and deterministic. Probabilistic models can give important details however in general these methods require a lot of data. This can result in insufficient inference accuracy as biological data is scarce, noisy and systems are complex. In deterministic models nonlinear modeling approaches such as S-Systems can explain biological data well enough however these models have many parameters and they can only target relatively smaller size networks. Linear ODE models can involve larger networks with reasonable accuracy. Much of the research on these models however didn't consider high noise component in biological networks. We introduced a novel constrained total least squares formulation based on a linear discrete ODE model in Chapter 3 to specifically address this problem.

As biological data become abundant more data has been collected to shed light on connectivity of biological networks. There have been several studies on combining topology data with dynamic measurements and details of some of these studies are given [37-39]. For example, in studies [37, 38] binding or sequence data can provide initial connectivity (positions of the linkages) in the networks. As genomic data is scarce and

noisy and systems are complex, estimating TF protein levels for all time points in addition to regulatory strengths may lead to poor reconstruction accuracy. Estimating the levels of Transcription Proteins may give a better picture for understanding regulatory networks, but it comes at the cost of higher quality and quantity of data. In [39] Bayesian networks are employed to integrate the data but since they are probabilistic frameworks the data requirement can be high and for limited data it can lead to poor performance. In chapter 4 we introduced a Bayesian parameter estimation framework that integrates connectivity and dynamic data. In our model gene to gene connectivity is considered. By doing this, we mainly focus on distributions of regulatory strengths between TF genes and target genes. These gene levels can be obtained through the microarray data.

Due to complexity of biological networks accurate prior connectivity information is essential to be able to reverse engineer these networks. Missing links in the connectivity information can be a major hurdle for prediction of these networks dynamics. Therefore predicting missing connections in the networks becomes crucial. Link prediction methods are essentially based on estimating links between a pair of nodes depending on the commonality of their topological features. Common Neighbors is a standard method of link prediction. It only relies only on the number of commonly shared nodes between the pair of nodes therefore it doesn't consider degree distribution of the nodes. Jaccard index is a variation of CN method in which out-degrees of the nodes are considered in the score function. Adamic Adar method is based on both number of commonly shared nodes and natural logarithm of their degree distribution but this method doesn't take into account the out-degrees of the nodes. There has been no study on p-value based probabilistic approaches for scoring node to node associations based on the

all three components; out-degree, in-degree and number of commonly shared nodes. In chapter 5, we introduced a novel link prediction method based on a probabilistic approach.

Many biological networks can be represented as a multipartite network that is a combination of several bipartite networks. Predicting significant links on these networks can have many potential applications. One important application can be drug side effect prediction. Side effect prediction has not been studied from the perspective of networks. In chapter 6, we introduce a framework for side effect prediction from targets, pathways as well as mouse phenotypes.

CHAPTER 3

A NOVEL CONSTRAINED TOTAL LEAST SQUARES METHOD FOR THE IDENTIFICATION OF BIOLOGICAL NETWORKS FROM NOISY MEASUREMENTS

3.1 Summary

A detailed overview of biological networks, data types, and computational methods was given in the previous chapter. There are two main issues with network identification problem; inference of topology and dynamics. Biological data is quite noisy and can be scarce. Furthermore, topological information can be incomplete, inconsistent or unknown for many biological systems. Therefore, in this chapter we address the problem of biological network identification without prior knowledge on connectivity only using noisy time series data. Least Squares is a commonly used method as parameter estimation framework for this kind of problem. However a discrete time model for this identification problem will lead to noise in both dependent and independent variables. Moreover, this error is serially correlated. To address this problem we propose a novel constrained total least squares algorithm. We demonstrate its superior performance over commonly used regression techniques such as least squares (LS), total least squares and existing Constrained Total Least squares approaches [64] on artificial network examples.

3.2 Introduction

The functions of living organisms are achieved through interactions of cell's components. These interactions create large networks. It has become essential to understand these networks to have a better picture of diseases and design better drugs. Gene regulatory

networks sit at the core of these diverse networks and they have been studied intensively. Since the advent of diverse genomic data techniques from mathematics, statistics, engineering and computer science methods have been proposed to understand the topology and dynamics of regulatory networks. These methods are collected under the umbrella of "Systems Biology" that has emerged as an interdisciplinary science. An outstanding addition to the ability to generate genomic information is the microarray technology, and the majority of network inference efforts are focused on reverse engineering regulatory networks from time series measurements [6,15,21, 22, 24, 28].

These studies model the dynamics of gene regulation as a biochemical network of gene products, typically mRNA and proteins, and describe their rate of changes through system of ordinary differential equations. Since the time series experiments are available in discrete time points inference methods are developed as discrete time equations. In this type of model the expression level of a gene is assumed to be the concentration of its transcript. The concentration of a particular transcript at time point $k+1$, $\hat{x}_i^{k+1}$ is given by the linear function of the concentrations of other RNA species at time point, k;

$$\hat{x}_i^{k+1} = \sum_{j=1}^{N} a_{ij}\hat{x}_j^k + \varepsilon_i^k + u_i^k, \qquad i = 1,..,N \qquad \varepsilon_i^k \approx N\left(0, \sigma_e^2\right) \tag{3.1}$$

where N is the number of transcripts in the network and a_{ij} is the regulatory strength between gene pairs i and j. ε_i^k is the error term for the difference between observation and the model. The errors are assumed to have Gaussian distribution with zero mean and standard deviation of σ_e^2. The input term for this model is represented as u_i^k. The aim is to estimate parameter values, a_{ij}'s, from micro-array observations, $\hat{x}_i^k$, thereby

reconstructing the gene network. A negative a_{ij} indicates an inhibition, and a positive value for a_{ij} stands for activation between the gene pair. In general, only a small subset of all RNA species regulates a particular transcript, which means most of the a_{ij}'s are zero. In other words, the gene networks are sparse. [4].

Microarray data is usually subject to high levels of additive and multiplicative errors [5]. Therefore, one can write concentration levels for genes as follows;

$$\hat{x}_i^k = x_i^k + e_i^k \qquad e_i^k = x_i^k u_i^k + v_i^k \qquad u_i^k \approx N\left(0, \sigma_u^2\right) \qquad v_i^k \approx N\left(0, \sigma_v^2\right) \tag{3.2}$$

In this equation, x_i^k is the unknown true value for concentration of i^{th} gene at k^{th} time point and e_i^k is the measurement error. The terms $x_i^k u_i^k$ and v_i^k correspond to multiplicative and additive parts of the measurement error.

Using equation (3.1) and (3.2), one can write the model for all genes,

$$\vec{x}^{k+1} + \vec{e}^{k+1} = A\left(\vec{x}^k + \vec{e}^k\right) + \vec{u}^k \tag{3.3}$$

where , $\vec{x}^k = \left[x_1^k, ..., x_N^k\right]^T$, $e^k = \left[\varepsilon_1^k, ..., \varepsilon_N^k\right]^T$, $u^k = \left[u_1^k, ..., u_N^k\right]^T$ and $A = \left\{a_{ij}\right\} \in IR^{N \times N}$

Equation (3.3) can be written for all time points, $k = 1, ..., M$, as follows;

$$X_2 + E_2 = A\left(X + E_1\right) + U \tag{3.4}$$

where $X_2 = \left[\vec{x}^2, ..., \vec{x}^M\right]$, $X_1 = \left[\vec{x}^1, ..., \vec{x}^{(M-1)}\right]$, $E_2 = \left[e^2, ..., e^M\right]$,and $E_1 = \left[e^1, ..., e^{M-1}\right]$.

One can see that both dependent and independent variables have error terms (Eq.3.4). Furthermore E_1 and E_2 are serially correlated as they have same columns except for the first and last columns.

Majority of inference algorithms for discrete time models focused on least squares regression. Least squares assume error terms are limited to only dependent variables. A significant problem from the regression standpoint is that both independent and dependent variables have high level of noise. Moreover, these noise terms are serially correlated. Noise has significant impact on parameter estimation of the networks. It is obvious that more advanced inference algorithms are needed that can take into account critical noise component. Kim et al [64]. proposed an application of constrained total least squares algorithm (CTLS) that is ideally suited for this model formulation. As seen in Equation (3.4) the models is corrupted by noise in both sides of the equation and noise is serially correlated. In their CTLS approach for a multi-variable network model, parameter estimation for each dependent variable is calculated separately. However since error propagates in time with parameter matrix (Eq. (3.3)) one should estimate parameters for dependent variables simultaneously. We introduced a novel CTLS algorithm that takes care of correlated noise and estimates parameters for dependent variables simultaneously. We compared our methods with Kim et al [61]'s examples as well as other common regression methods. We observed significant improvement over traditional regression models and their CTLS framework.

3.3 Methods

We adopt a linear discrete time model for gene regulatory network. All equations are written for all M data points and N nodes in a state space representation. States refer to the values of expression levels and errors representation experimental error.

$$X_2 + E_2 = A(X + E_1) + U \qquad \textbf{(3.5)}$$

In this representation the observation for dependent and independent variables are decomposed into true value and error terms.

In this equation state matrices are written as $X_2 = \left[\vec{x}^2,...,\vec{x}^M\right]$, $X_1 = \left[\vec{x}^1,...,\vec{x}^{(M-1)}\right]$. Similarly error matrices are; $E_2 = \left[\vec{\varepsilon}^2,...,\vec{\varepsilon}^M\right]$, $E_1 = \left[\vec{\varepsilon}^1,...,\vec{\varepsilon}^{M-1}\right]$. Each column of error and state matrices corresponds to the vectors at k^{th} time step; $\vec{x}^k = \left[x_1^k,...,x_N^k\right]^T$, $\vec{\varepsilon}^k = \left[\varepsilon_1^k,...,\varepsilon_N^k\right]^T$. Input matrix is represented as; $U = \left[u^1,...,u^M\right]$. Parameters are also represented in a matrix form; $A = \left\{a_{ij}\right\} \in IR^{N\times N}$

Inputs are assumed to remain same for each time step. Equation (3.5) can be rewritten as follows;

$$X_2^{N\times(M-1)} + E_2^{N\times(M-1)} = \left[A^{N\times N} \quad u^{N\times 1}\right]\left(\begin{bmatrix} X_1^{N\times(M-1)} \\ 1^{1\times(M-1)} \end{bmatrix} + \begin{bmatrix} E_1^{N\times(M-1)} \\ 0^{N\times(M-1)} \end{bmatrix}\right) \tag{3.6}$$

This model can be further extended for a P parallel experiment case as follows;

$$\begin{aligned} &\left[{}^{(1)}X_2^{N\times(M-1)} \;,..,\; {}^{(P)}X_2^{N\times(M-1)}\right] + \left[{}^{(1)}E_2^{N\times(M-1)} \;,...,\; {}^{(P)}E_2^{N\times(M-1)}\right] \\ &= \left[A^{N\times N} \quad {}^{(1)}u^{N\times 1} \;,..,\; {}^{(P)}u^{N\times 1}\right]\left(\begin{bmatrix} {}^{(1)}X_1^{N\times(M-1)} & {}^{(2)}X_1^{N\times(M-1)} & ,.., & {}^{(P)}X_1^{N\times(M-1)} \\ \vec{1}^{1\times(M-1)} & \vec{0}^{1\times(M-1)} & ,.., & \vec{0}^{1\times(M-1)} \\ \vec{0}^{1\times(M-1)} & \vec{1}^{1\times(M-1)} & ,.., & \vec{0}^{1\times(M-1)} \\ . & . & . & . \\ . & . & . & . \\ \vec{0}^{1\times(M-1)} & . & \vec{0}^{1\times(M-1)} & \vec{1}^{1\times(M-1)} \end{bmatrix} + \begin{bmatrix} {}^{(1)}E_1^{N\times(M-1)} & ,.., & {}^{(P)}E_1^{N\times(M-1)} \\ 0^{P\times(M-1)} & & 0^{P\times(M-1)} \end{bmatrix}\right) \end{aligned} \tag{3.7}$$

One can write this equation in a compact form.

$$\underline{X}_2 + \underline{E}_2 = \underline{A}\left(\underline{X}_1 + \underline{E}_1\right) \tag{3.8}$$

If the error terms are ignored the least square estimation for this equation can be written as;

$$\hat{\underline{A}} = \left(\underline{X}_1\left(\underline{X}_1\right)^T\right)^{-1}\left(\underline{X}_1\right)\left(\underline{X}_2\right)^T \tag{3.9}$$

Total Least Squares

Total least square estimation for this system is shown for i^{th} dependent variable for all data points and experiments as follows;

$$\left[{}^{(1)}_{i}X_2^{\,1\times(M-1)} \quad ,..., \quad {}^{(P)}X_2^{\,1\times(M-1)} \right] + \left[{}^{(1)}_{i}e_2^{\,1\times(M-1)} \quad ,..., \quad {}^{(P)}_{i}e_2^{\,1\times(M-1)} \right]$$

$$= \left[{}_{i}a^{1\times N} \quad {}^{(1)}_{i}u^{1\times 1} \quad ,.., \quad {}^{(P)}_{i}u^{1\times 1} \right] \left(\begin{bmatrix} {}^{(1)}X_1^{\,N\times(M-1)} & {}^{(2)}X_1^{\,N\times(M-1)} & ,.., & {}^{(P)}X_1^{\,N\times(M-1)} \\ \vec{1}^{1\times(M-1)} & \vec{0}^{\,1\times(M-1)} & ,.., & \vec{0}^{\,1\times(M-1)} \\ \vec{0}^{\,1\times(M-1)} & \vec{1}^{1\times(M-1)} & ,.., & \vec{0}^{\,1\times(M-1)} \\ . & . & . & . \\ . & . & . & . \\ \vec{0}^{\,1\times(M-1)} & . & \vec{0}^{\,1\times(M-1)} & \vec{1}^{1\times(M-1)} \end{bmatrix} + \begin{bmatrix} {}^{(1)}E_1^{\,N\times(M-1)} & ,.., & {}^{(P)}E_1^{\,N\times(M-1)} \\ 0^{P\times(M-1)} & ,.., & 0^{P\times(M-1)} \end{bmatrix} \right) \tag{3.10}$$

In this equation first term refers to the i^{th} row of the $\underline{X}_2$ and second term is the i^{th} row of the $\underline{E}_2$. The first term at the right side of the equation is the i^{th} row of the parameter matrix $\underline{A}$.

This equation is rearranged as follows;

$$\left[a^{1\times N} \quad {}^{(1)}_{i}u^{1\times 1} \quad ,.., \quad {}^{(P)}_{i}u^{1\times 1} \quad -1 \right] \left\{ \begin{bmatrix} {}^{(1)}X_1^{\,N\times(M-1)} & {}^{(2)}X_1^{\,N\times(M-1)} & ,.., & {}^{(P)}X_1^{\,N\times(M-1)} \\ \vec{1}^{1\times(M-1)} & \vec{0}^{\,1\times(M-1)} & ,.., & \vec{0}^{\,1\times(M-1)} \\ \vec{0}^{\,1\times(M-1)} & \vec{1}^{1\times(M-1)} & ,.., & \vec{0}^{\,1\times(M-1)} \\ . & . & . & . \\ . & . & . & . \\ \vec{0}^{\,1\times(M-1)} & . & \vec{0}^{\,1\times(M-1)} & \vec{1}^{1\times(M-1)} \\ {}^{(1)}X_2^{\,1\times(M-1)} & {}^{(2)}X_2^{\,1\times(M-1)} & ,.., & {}^{(P)}X_2^{\,1\times(M-1)} \end{bmatrix} + \begin{bmatrix} {}^{(1)}E_1^{\,N\times(M-1)} & ,.., & {}^{(P)}E_1^{\,N\times(M-1)} \\ {}^{(1)}0^{P\times(M-1)} & ,.., & {}^{(P)}0^{P\times(M-1)} \\ {}^{(1)}e_2^{\,1\times(M-1)} & ,.., & {}^{(P)}e_2^{\,1\times(M-1)} \end{bmatrix} \right\} = 0 \tag{3.11}$$

One can write this equation in a compact form as;

$$\left[{}_{i}a^{1\times N} \quad {}_{i}b^{1\times N} \quad -1 \right] \left(C + \Delta C \right) = 0 \qquad for \quad i = 1,.., N \tag{3.12}$$

Total least square formulation is written as follows;

$\min_{a,b} \left\| \Delta C \right\|_F^2$ Subject to Equation (3.12)

The solution to this becomes;

$$\hat{\underline{A}}_{TLS} = \left(\underline{X}_1 \left(\underline{X}_1\right)^T - \lambda^2 I\right)^{-1} \left(\underline{X}_1\right)\left(\underline{X}_2\right)^T \quad \textbf{(3.13)}$$

Where λ is the smallest singular value of C. Compared to least squares, the TLS solution has a correction term, λ^2, in the inverse term. This reduces the bias in the solution that is caused by noise in the independent variables.

Constrained Total Least Squares method

Least squares solution is not optimal when there is noise in independent variables. Total least squares takes into account the noise term in the independent variables. However it is not the best approach when the noise term is correlated, which is the case in this formulation. Kim et al [64] proposed a CTLS framework that considers correlation in the noise term. Their method based on estimating parameters for each dependent variable one at a time similar to the TLS methodology. However noise terms in the independent variables are correlated as a function of all the rows of parameter matrix rather than each row. In our formulation we address this problem and reformulated CTLS framework for this model.

To do that we started with rewriting equation (3.5) as follows;

$${}^{(k)}e_2^i = {}^{(k)}x_2^{(i)} + A \times \left({}^{(k)}e_1^{(i)} - {}^{(k)}x_1^{(i)}\right) - {}^{(k)}u^{(i)} \quad for \quad i = 1,.., (M-1) \quad \textbf{(3.14)}$$

Here ${}^{(k)}e_2^i$ is the i^{th} column of matrix, ${}^{(k)}E_2$ and it stands for error term for all dependent variables for all time points at the k^{th} experiment. Similarly ${}^{(k)}e_1^{(i)}$ and ${}^{(k)}x_1^{(i)}$ are the i^{th} columns of matrices, ${}^{(k)}E_1$ and ${}^{(k)}X_2$ respectively.

One can write equation (3.14) for $i = 1$ and $i = 2$ as follows;

$${}^{(k)}e_2^{(1)} = {}^{(k)}x_2^{(1)} + A \times \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right) - {}^{(k)}u^{(1)} \quad \textbf{(3.15)}$$

$${}^{(k)}e_2^{(2)} = {}^{(k)}x_2^{(2)} + A \times \left({}^{(k)}e_1^{(2)} - {}^{(k)}x_1^{(2)}\right) - {}^{(k)}u^{(2)} \quad \textbf{(3.16)}$$

The i^{th} column of ${}^{(k)}E_2$ is identical to the $(i+1)^{th}$ column of the matrix, ${}^{(k)}E_1$. Therefore, one can write;

$${}^{(k)}e_2^{(1)} = {}^{(k)}e_1^{(2)} \quad \textbf{(3.17)}$$

If this is plugged in equation (3.16), it becomes;

$${}^{(k)}e_2^{(2)} = {}^{(k)}x_2^{(2)} + A \times \left({}^{(k)}e_2^{(1)} - {}^{(k)}x_1^{(2)} \right) - {}^{(k)}u \quad \textbf{(3.18)}$$

Equation (3.15) and (3.18) will yield the following expression;

$${}^{(k)}e_2^{(2)} = {}^{(k)}x_2^{(2)} + A \times \left({}^{(k)}x_2^{(1)} + A \times \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)} \right) - {}^{(k)}u^{(1)} - {}^{(k)}x_1^{(2)} \right) - {}^{(k)}u^{(2)} \quad \textbf{(3.19)}$$

Similar to equation (3.17) i^{th} column of ${}^{(k)}X_2$ is identical to the $(i+1)^{th}$ column of the matrix, ${}^{(k)}X_1$.

$${}^{(k)}x_2^{(1)} = {}^{(k)}x_1^{(2)} \quad \textbf{(3.20)}$$

Plugging equation (3.20) in equation (3.19) and doing necessary cancellations will lead to the following;

$${}^{(k)}e_2^{(2)} = {}^{(k)}x_2^{(2)} + A \times \left(A \times \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)} \right) - {}^{(k)}u^{(1)} \right) - {}^{(k)}u^{(2)} \quad \textbf{(3.21)}$$

We assume that input stays constant throughout time for each experiment;

$${}^{(k)}u^{(1)} = {}^{(k)}u^{(2)} = \ldots = {}^{(k)}u^{(M-1)} = {}^{(k)}u \quad \textbf{(3.22)}$$

Equation (3.21) can be rearranged as;

$${}^{(k)}e_2^{(2)} = {}^{(k)}x_2^{(2)} + A^2 \times \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)} \right) - \left(A + I \right) {}^{(k)}u \quad \textbf{(3.23)}$$

One can write this equation for all $M-1$ columns;

$${}^{(k)}e_2^{(i)} = {}^{(k)}x_2^{(i)} + A^i \times \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)} \right) - M^{(i)} \times {}^{(k)}u \qquad M^{(i)} = \left(\sum_{n=0}^{n=i-1} A^n \right) \quad \textbf{(3.24)}$$

Sum of squared error terms for all time points and experiments are represented as follows;

$$E^{Total} = \sum_{k=1}^{p} \left({}^{(k)}e_1^{(1)} \right)^T \left({}^{(k)}e_1^{(1)} \right) + \sum_{k=1}^{p} \sum_{i=1}^{m-1} \left({}^{(k)}e_2^{(i)} \right)^T \left({}^{(k)}e_2^{(i)} \right) \quad \textbf{(3.25)}$$

Using equation (3.24) and equation (3.25), we obtain;

$$
\begin{aligned}
E^{Total} = & \sum_{k=1}^{p} \left({}^{(k)}e_1^{(1)}\right)^T \left({}^{(k)}e_1^{(1)}\right) + \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}x_2^{(i)}\right)^T \left({}^{(k)}x_2^{(i)}\right) \\
& + \sum_{k=1}^{p} \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right)^T \sum_{i=1}^{m-1} \left(A^i\right)^T {}^{(k)}x_2^{(i)} \\
& + \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}x_2^{(i)}\right)^T \left(A^i\right) \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right) \\
& + \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right)^T \left(A^i\right)^T \left(A^i\right) \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right) \\
& - \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}x_2^{(i)}\right)^T \left(M^{(i)}\right) \left({}^{(k)}u\right) - \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}u\right)^T \left(M^{(i)}\right)^T \left({}^{(k)}x_2^{(i)}\right) \\
& - \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right)^T \left(A^i\right)^T \left(M^{(i)}\right) \left({}^{(k)}u\right) \\
& - \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}u\right)^T \left(M^{(i)}\right)^T \left(A^i\right) \left({}^{(k)}e_1^{(1)} - {}^{(k)}x_1^{(1)}\right) \\
& + \sum_{k=1}^{p}\sum_{i=1}^{m-1} \left({}^{(k)}u\right)^T \left(M^{(i)}\right)^T \left(M^{(i)}\right) \left({}^{(k)}u\right)
\end{aligned}
\qquad \textbf{(3.26)}
$$

E^{Total} is a function of error and state at initial time step for all experiments, as well as input and measurements. Constrained total least squares is simply unconstrained minimization of E^{Total}. In this optimization problem decision variables are A, ${}^{(k)}u$, and ${}^{(k)}e_1^{(1)}$; input variables are ${}^{(k)}x_1^{(1)}$ and ${}^{(k)}X_2$. In other words, optimization should search for parameter space (A, ${}^{(k)}u$) that will minimize the equation (3.26).

$$E^{Total} = f\left(A, \left({}^{(k)}e_1^{(1)}\right), \left({}^{(k)}u\right), \left({}^{(k)}x_1^{(1)}\right), {}^{(k)}X_2\right) \qquad \textbf{(3.27)}$$

We approached this problem in a step wise manner. The first step searches for matrix A that minimizes E^{Total} with respect to error at initial time step, ${}^{(k)}e_1^{(1)}$. This is represented as;

$$\frac{\partial\left(E^{Total}\right)}{\partial\left({}^{(k)}e_1^{(1)}\right)} = \vec{0} \qquad \textbf{(3.28)}$$

Least squares solution from equation (3.9) is given as initial condition to this minimization problem. Calculating the above equation will give the following set of equations (see Appendix for details) ;

$$\left({}^{(k)}e_1^{(1)}\right)^T = \left(-2\left({}^{(k)}R\right)^T + \left({}^{(k)}x_1^{(1)}\right)^T S' + 2\left({}^{(k)}u\right)^T Z^T \right)\left(2I + S'\right)^{-1}$$

$${}^{(k)}R = \sum_{i=1}^{m-1}\left(A^i\right)^T \, {}^{(k)}x_2^{(i)} \qquad S = \sum_{i=1}^{m-1}\left(A^i\right)^T\left(A^i\right) \qquad M^{(i)} = \left(\sum_{n=0}^{n=i-1} A^n\right)$$

$$Z = \sum_{i=1}^{m-1}\left(A^i\right)^T \times \left(M^{(i)}\right) \qquad S' = S + S^T \qquad \textbf{(3.29)}$$

In the second step of minimization problem, the resulting value for the term $\left({}^{(k)}e_1^{(1)}\right)^T$ from the first step will be used. This step searches for parameters that will minimize E^{Total} with respect to ${}^{(k)}u$. This is simply the solution for the following equation;

$$\left.\frac{\partial\left(E^{Total}\right)}{\partial\left({}^{(k)}u\right)}\right|_{{}^{(k)}e_1^{(1)} = \left({}^{(k)}e_1^{(1)}\right)^*} = 0 \qquad \textbf{(3.30)}$$

After several matrix calculus steps this equation will lead to following set of expressions (see Appendix for details) ;

$$\left({}^{(k)}U\right)^T = 2\times\left\{\left({}^{(k)}D\right)^T + \left({}^{(k)}E_1^{(1)} - {}^{(k)}X_1^{(1)}\right)^T Z\right\}\left(H'\right)^{-1}$$

$$H = \sum_{i=1}^{m-1}\left(M^{(i)}\right)^T\left(M^{(i)}\right) \qquad {}^{(k)}D = \sum_{i=1}^{m-1}\left(M^{(i)}\right)^T\left({}^{(k)}X_2^{(i)}\right) \qquad Z = \sum_{i=1}^{m-1}\left(A^i\right)^T \times \left(M^{(i)}\right) \qquad \textbf{(3.31)}$$

In the case of multiplicative and additive noise in measurement are represented as in equation (3.2). CTLS framework can be modified to take into account of the nature of the noise terms.

$${}^{(k)}e_i^{(j)} = b\left({}^{(k)}x_i^{(j)}\right)\omega + v \qquad u_i^k \approx N\left(0\,,1\right) \qquad v_i^k \approx N\left(0, \sigma_v^2\right) \qquad \textbf{(3.32)}$$

In the above equation, ${}^{(k)}e_i^{(j)}$ refers to the error for i^{th} state measurement at j^{th} time point of the k^{th} experiment. Similarly, ${}^{(k)}x_i^{(j)}$ is the value of i^{th} state at j^{th} time points for the k^{th} experiment. The term ω is assumed to have a normal distribution. b is a constant that accounts for the ratio of variance of the error to the signal. One can redefine the noise term in equation (3.32) as follows;

$$ {}^{(k)}e_i^{(j)} \approx N\left(0,\ \left\{b\left({}^{(k)}x_i^{(j)}\right)+\sigma_v^2\right\}\right) \qquad \textbf{(3.33)} $$

Equation (3.25) can be modified to integrate the noise model in equation (3.33). This is achieved by calculating sum of squared of weighted errors. With the new noise model, equation (3.25) becomes;

$$ E^{Total} = \sum_{k=1}^{p}\left({}^{(k)}e_1^{(1)}\right)^T\left({}^{(k)}W^{(1)}\right)^T\left({}^{(k)}W^{(1)}\right)\left({}^{(k)}e_1^{(1)}\right)+\sum_{k=1}^{p}\sum_{i=1}^{m-1}\left({}^{(k)}e_2^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left({}^{(k)}e_2^{(i)}\right) \qquad \textbf{(3.34)} $$

In this expression, weight matrix ${}^{(k)}W^{(i)}$ is defined as follows;

$$ {}^{(k)}W^{(i)} = bI^{NxN}\left({}^{(k)}\vec{x}^{(j)}\right)^{Nx1}+I\left(\vec{\sigma}_v^2\right)^{Nx1} \qquad \textbf{(3.35)} $$

where, ${}^{(k)}\vec{x}^{(j)}$ is the state vector of size N at the j^{th} time point for the k^{th} experiment.

Minimization for the modified term can be calculated similar to the equations (3.28-3.31). The weighted version of equation (3.29) becomes;

$$ \left({}^{(k)}e_1^{(1)}\right)^T = \left(-2\left({}^{(k)}R_W\right)^T+\left({}^{(k)}x_1^{(1)}\right)^T S'_W+2\left({}^{(k)}u\right)^T Z_W{}^T\right)\left(2I+S'_W\right)^{-1} $$

$$ {}^{(k)}R_W = \sum_{i=1}^{m-1}\left(A^i\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left({}^{(k)}x_2^{(i)}\right) \qquad S_W = \sum_{i=1}^{m-1}\left(A^i\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left(A^i\right) $$

$$ Z_W = \sum_{i=1}^{m-1}\left(A^i\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left(M^{(i)}\right) \qquad M^{(i)} = \left(\sum_{n=0}^{n=i-1}A^n\right) \qquad S'_W = S_W + S_W{}^T \qquad \textbf{(3.36)} $$

Similarly equation (3.30) can be modified to include weights;

$$\left({}^{(k)}U\right)^T = 2\times\left\{\left({}^{(k)}D\right)^T + \left({}^{(k)}E_1^{(1)} - {}^{(k)}X_1^{(1)}\right)^T Z\right\}(H')^{-1}$$

$$H = \sum_{i=1}^{m-1}\left(M^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left(M^{(i)}\right) \qquad {}^{(k)}D = \sum_{i=1}^{m-1}\left(M^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left({}^{(k)}X_2^{(i)}\right)$$

$$Z = \sum_{i=1}^{m-1}\left(A^i\right)^T\left({}^{(k)}W^{(i)}\right)^T\left({}^{(k)}W^{(i)}\right)\left(M^{(i)}\right) \qquad \textbf{(3.37)}$$

3.4 Results

We applied our algorithm on network models that are presented by Kim et al. Their first example network is a four gene network modeled by nonlinear differential equations. The model is shown figure (3.1)

$$\dot{x}_1(t) = V_1^S\frac{1+A_{14}\left(x_4(t)/K_{14a}\right)^{n_{14}}}{\left(1+\left(x_4(t)/K_{14a}\right)^{n_{14}}\right)\left(1+\left(x_2(t)/K_{12i}\right)^{n_{12}}\right)} - V_{1d}\frac{x_1(t)}{k_{1d}+x_1(t)}$$

$$\dot{x}_2(t) = V_2^S\frac{1+A_{24}\left(x_4(t)/K_{24a}\right)^{n_{24}}}{\left(1+\left(x_4(t)/K_{14a}\right)^{n_{24}}\right)} - V_{2d}\frac{x_2(t)}{k_{2d}+x_2(t)}$$

$$\dot{x}_3(t) = V_3^S\frac{1+A_{14}\left(x_4(t)/K_{14a}\right)^{n_{14}}}{\left(1+\left(x_2(t)/K_{32a}\right)^{n_{32}}\right)\left(1+\left(x_1(t)/K_{31i}\right)^{n_{31}}\right)} - V_{3d}\frac{x_3(t)}{k_{3d}+x_3(t)}$$

$$\dot{x}_4(t) = V_4^S\frac{1+A_{43}\left(x_3(t)/K_{43a}\right)^{n_{43}}}{\left(1+\left(x_2(t)/K_{43a}\right)^{n_{43}}\right)} - V_{4d}\frac{x_4(t)}{k_{4d}+x_4(t)} \qquad \textbf{(3.38)}$$

In this set of equations $\dot{x}_3(t)$ is the rate of change for the expression of i^{th} gene at time t. V^S and K and values correspond to maximum enzyme rates and Michaelis constants respectively. Parameter values for this model are given as follows; $V_1^S = 5$, $V_2^S = 3.5, V_3^S = 3, V_4^S = 4, V_{1d} = 200, V_{2d} = 500, V_{3d} = 150, V_{4d} = 500, K_{14a} = 1.6,$ $K_{24a} = 1.6, K_{32a} = 1.5, K_{43a} = 0.15, K_{12i} = 0.5, K_{31i} = 0.7, K_{1d} = 30, K_{2d} = 60, K_{3d} = 10,$ $K_{4d} = 50, A_{14} = 4, A_{24} = 4, A_{32} = 5, A_{43} = 2, n_{12} = 1, n_{24} = 2$, $n_{31} = 1, n_{32} = 2, n_{43} = 2$. In

this model the levels of perturbation for V_i^S are 100% from their nominal values. The measurement noise is assumed to be zero-mean white Gaussian with variance equal to the square of the equilibrium 0.02. Equilibrium state values are given as follows; $x_1^{eq} = 0.4920, x_2^{eq} = 0.6052$, $x_3^{eq} = 0.1866$, $x_4^{eq} = 0.6514$ [64]. Drift noise case is also considered for this model. The details of this noise model can be found in [64]. There are four parallel experiments and in each experiment one of the four V_i^S values are perturbed in negative direction. Within each experiment, measurements are taken at a rate of (36s) for a number of time points varying from 3 to 60.

The second network example is the feedback interactions between the tumor suppressor $p53$ and the oncogene $mdm2$. Feedback mechanism causes oscillations in the systems that vary from cell to cell. This model has been received attention in the recent literature. The underlying nonlinear ordinary differential equation for this system is given as [64];

$$\frac{d[p53]}{dt} = s_{p53} - \delta_{p53}[p53]$$

$$\frac{d[mdm2]}{dt} = s_{mdm2} - \delta_{mdm2}[mdm2] + \varepsilon_{mdm2}\frac{[p53(t-\tau_1)]^n}{[p53(t-\tau_1)]^n + K^n} \quad \textbf{(3.39)}$$

The level of perturbation on $p53$ is negative 10% and the measurement sampling time is 2 hours and white noise is added to the measurements.

Jacobian for 4-gene and 2-gene network examples are given as follows [64];

$$F = \begin{bmatrix} -6.45 & -2.92 & 0 & 2.54 \\ 0 & -8.17 & 0 & 3.93 \\ -2.31 & 2.80 & -14.46 & 0 \\ 0 & 0 & 10.22 & -9.74 \end{bmatrix}, \qquad F = \begin{bmatrix} -0.02 & 0 \\ 0 & -0.02 \end{bmatrix} \quad \textbf{(3.40)}$$

The nonlinear models are perturbed at the equilibrium values to obtain a linear ordinary differential equation.

$$\Delta\dot{x}(t) = F\ \Delta x(t) + u(t) \tag{3.41}$$

where, F is the Jacobian, $u(t)$ is the input term at time t and $\Delta\dot{x}(t)$ is the vector for the rate of changes in the deviation of the expression levels of all four genes from their equilibrium values.

The aim of regression methods is to estimate Jacobian matrix correctly under the uncertainty and noise in the data. Since the measurement data are taken at discrete time steps, the discrete time representation of the equation is employed. This is identical to the equation (3.8).

$$\underline{X}_2 + \underline{E}_2 = \underline{A}\left(\underline{X}_1 + \underline{E}_1\right) \tag{3.42}$$

,where $\underline{X}_1, \underline{X}_2$ is the measurement of the deviation for all states, time steps and experiments. The open forms of these matrices are given in equation (3.11).

$\underline{A}$ is estimated through different methods in discrete domain. Estimation for Jacobian F can be calculated through relation [64];

$$\hat{F} = \frac{1}{\Delta T}\log\left(\hat{A}\right) \tag{3.43}$$

,where ΔT is the sampling time. Another transformation from discrete to continuous domain is the bilinear transformation.

$$\hat{F} = \frac{2}{\Delta T}\left(\hat{A} - I\right)\left(\hat{A} + I\right)^{-1} \tag{3.44}$$

Jacobian gives a quantitative picture of the local structure of the networks, therefore it is important to estimate each element correctly. The estimation error between

true Jacobian F and estimated Jacobian $\hat{F}$ can be calculated in different ways. We adopted the error definitions from [64]. The first error criteria is defined as;

$$\varepsilon_M = \frac{1}{N_1}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|\alpha_{ij}\right| + \frac{1}{N_2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|\beta_{ij}\right| \qquad \alpha_{ij} = \begin{cases} \dfrac{\hat{f}_{ij} - f_{ij}}{f_{ij}} & for\ f_{ij} \neq 0 \\ 0 & otherwise \end{cases} \qquad \beta_{ij} = \begin{cases} 0 & f_{ij} \neq 0 \\ f_{ij} & otherwise \end{cases} \tag{3.45}$$

where $\hat{f}_{ij}$ and f_{ij} are the i^{th} row and j^{th} column elements for matrices $\hat{F}$ and F respectively. N_1 and N_2 terms are the number of non-zero and zeros in true Jacobian matrix.

The second error criterion is defined as follows;

$$\varepsilon_S = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|\, sign\left(\alpha_{ij}\right) - sign\left(\beta_{ij}\right)\right| \tag{3.46}$$

Where $sign\left(a\right)$ is a function that has the sign of a as its value,i.e.,-1,0,1 for $a < 0$ $a = 0$, and $a > 0$, respectively.

The third error definition is based on the Frobenius norm of the difference in true and estimated Jacobian matrices.

$$\varepsilon_F = \left\|\hat{F} - F\right\| \tag{3.47}$$

For first four-gene network example all methods are compared according to this three error criteria for different number of samples ranging from 6-30. The results are generated from 1000 Monte-Carlo simulations and they are tabulated in table 3.1.As it can be seen for small numbers of data TLS method has larger error compared to LS. This is due to the minimum requirement of data for TLS. Here CTLS-1 refers to the constrained total least squares method proposed by Kim et al. Our reformulation for

CTLS is represented as CTLS-2. We outperformed all regression methods as well as CTLS-1 consistently. Our method reduced ε_M error by an average of 21% and it improved ε_F by an average of 12% compared to CTLS-1. Compared to CTLS-1 we observed an average reduction of 21% and 27% for the standard deviation of ε_M and ε_F errors respectively. For ε_M all methods give a similar level of performance. It can be observed that the accuracy of the estimation increases with increasing data points.

Table 3.1. Four-gene network example with white noise only

Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
6	LS	16.44	5.25	0.59	0.14	70.92	18.48
	TLS	79.93	333.79	0.74	0.19	580.46	2453.05
	CTLS-1	15.98	6.94	0.64	0.16	69.89	28.22
	CTLS-2	**13.58**	**4.16**	0.69	0.17	**53.56**	**14.14**
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
9	LS	7.75	2.53	0.46	0.10	36.00	9.46
	TLS	11.56	9.59	0.54	0.14	64.76	80.98
	CTLS-1	6.58	2.88	0.47	0.11	32.16	13.16
	CTLS-2	**5.73**	**1.78**	0.49	0.11	**23.70**	**7.09**
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
12	LS	5.15	1.65	0.40	0.06	24.91	6.62
	TLS	6.20	2.43	0.45	0.09	31.98	15.03
	CTLS-1	3.75	1.47	0.40	0.05	19.62	7.04
	CTLS-2	**3.32**	**1.01**	0.40	0.06	**14.73**	**4.45**
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
21	LS	3.64	1.04	0.38	0.02	17.87	4.27
	TLS	3.61	1.34	0.38	0.02	20.29	8.64
	CTLS-1	2.16	0.68	0.38	0.02	11.31	2.98
	CTLS-2	**1.89**	**0.55**	0.38	0.02	**9.05**	**2.30**
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
30	LS	3.70	0.90	0.42	0.06	17.31	3.74
	TLS	3.47	1.27	0.44	0.07	18.82	7.46
	CTLS-1	2.27	0.57	0.49	0.03	10.01	2.00
	CTLS-2	**2.22**	**0.54**	0.48	0.04	**9.01**	**1.61**

To evaluate to drift noise effect we used the drift noise model in [64]. For different strength of drift noise the results for 1000 Monte Carlo simulations are shown in table 3.2. Our method consistently outperforms all methods for all levels of drift noise.

Table 3.2. Four-gene network example with both drift and white noise

Strength of drift noise	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
2.0	LS	8.91	3.50	0.48	0.11	40.61	14.60
	TLS	32.97	190.54	0.59	0.16	244.44	2341.86
	CTLS-1	8.20	4.18	0.52	0.13	40.40	22.61
	CTLS-2	**7.31**	**2.62**	0.54	0.13	**30.34**	**10.52**
Strength of drift noise	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
1.0	LS	6.18	2.07	0.43	0.08	29.17	8.58
	TLS	8.06	4.30	0.49	0.12	43.23	29.42
	CTLS-1	4.97	2.09	0.43	0.09	24.94	10.81
	CTLS-2	**4.53**	**1.48**	0.44	0.09	**19.24**	**5.93**
Strength of drift noise	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
0.1	LS	5.20	1.62	0.40	0.06	25.18	6.69
	TLS	6.30	2.36	0.45	0.09	33.18	16.51
	CTLS-1	3.80	1.52	0.40	0.06	19.54	7.11
	CTLS-2	**3.34**	**1.10**	0.40	0.06	**14.62**	**4.47**

In the second example with 2-gene network, our reformulation generally shows better performance compared to other regression methods. In this example, there are 4 experiments and the number of samples for each example is varied 8-16. The results are tabulated for all error criteria in table 3.3.

Table 3.3. Two-gene network example with both drift and white noise

Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
8	LS	0.82	0.16	0.50	0.00	0.03	0.01
	TLS	15.49	51.01	1.03	0.16	0.66	2.26
	CTLS-1	0.40	0.07	0.50	0.00	0.02	0.00
	CTLS-2	**0.31**	**0.06**	0.50	0.00	**0.01**	**0.00**
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
12	LS	0.50	0.05	0.50	0.00	0.02	0.00
	TLS	15.01	23.02	1.01	0.24	0.94	1.47
	CTLS-1	0.56	0.19	0.54	0.13	0.02	0.01
	CTLS-2	**0.35**	0.22	0.54	0.13	0.02	0.01
Number of samples	METHODS	ε_M		ε_S		ε_F	
		Mean	STD	Mean	STD	Mean	STD
16	LS	0.45	0.04	0.50	0.00	0.02	0.00
	TLS	13.22	30.86	1.02	0.21	1.11	2.55
	CTLS-1	0.52	0.17	0.53	0.11	0.02	0.00
	CTLS-2	**0.45**	**0.04**	0.53	0.11	0.02	0.00

3.5 Conclusions

In this chapter we addressed the problem of network identification from noisy measurements. It is known that biological data has significant levels of noise. In regression from dynamic data the resulting estimation model has noise term in both dependent and independent variable. TLS is capable of taking error in independent variables into account. CTLS is a further improvement on TLS that can incorporate the correlation in the noise.

We demonstrated superior performance of our novel CTLS framework on both existing one and other estimation methods on examples under the wide range of data points and noise levels.

Though CTLS methods seem to improve parameter estimation significantly over the existing methods, the error levels are still high despite reasonable noise levels. Therefore, it is necessary to use network connectivity data with a combination of optimal

experimental design to obtain high accuracy parameter estimation. In next chapter we will demonstrate our approach for incorporating prior connectivity data with time series data.

CHAPTER 4

A BAYESIAN APPROACH TO THE REVERSE ENGINEERING OF GENE NETWORKS

4.1 Summary

In the previous chapter we demonstrated the superior prediction power of our CTLS over various regression methods when there is no connectivity information is available. Connectivity data can be available from various sources such as protein-DNA binding data, protein interaction network databases and literature mining data. However this data can have high noise levels, inconsistencies or missing information. Time series microarray data is also corrupted by noise. It is essential to incorporate connectivity and time-series data to reverse engineer biological regulatory networks. The Bayesian framework is a powerful technique when there is connectivity and time series information with different levels of noise. In this chapter we introduced a novel Bayesian parameter estimation framework that is capable of incorporating noisy topology and time series data. We demonstrated our framework on artificial network examples with a varying level of noise and number of data points.

4.2 Introduction

With the advent of various types of genomics data, there is an increasing necessity for computational regulatory network inference models that can serve to integrate diverse data. In this regard, Sabatti and James [37] introduced a Bayesian estimation framework for reconstructing gene networks. They aimed at estimating regulatory strength of TFs on

genes along with TF protein levels at different time points using sequence and time-course micro-array data. In [38], similarly, the authors sought to estimate protein levels and regulatory strengths; however, they reached the dynamical model by considering the transcription process as a set of biochemical reactions. In both studies, the nodes of the gene network are the gene expression levels and TF protein levels. TF protein levels are unknown (hidden variables) and expression levels are the variables observed through micro-array data. Binding or sequence data provide topology (positions of linkages) in the network. As genomic data is scarce and noisy, estimating TF protein levels in addition to regulatory strengths may lead to poor reconstruction accuracy. TF protein levels as hidden variables will provide additional information on networks, but comes at the cost of higher quality and quantity of data. To this end, we concentrated on gene-to-gene connectivity and regulatory strength estimation by combined use of binding and time-series micro-array data. We introduce an algorithm that uses connectivity and time-course gene expression data from micro-arrays. Binding data offers an initial topology for the networks, but they do come with significant errors giving false positives and false negatives in the network connectivity. Expression data also includes significant levels of experimental noise. The key to tackle these problems is that the inaccuracies are less likely when the algorithms focus on the agreement of both types of data. Bernard et al. [39] used both types of data in the context of dynamics Bayesian networks where binding data serves as the prior for the network topology. DBN is a directed graph model for representing probabilistic independence relation between multiple interaction entities using dynamic data on Bayes rule. In their DBN approach, gene expression levels are random variables and each represents a node in the network. Using the Bayes rule, *a*

posteriori distribution on the graphical structure (topology) is sought by utilizing time-course gene expression data. Unlike DBN method, we employ a Bayesian estimation model on which *a posteriori* distributions of model parameters are obtained. By doing this, we mainly focus on distributions of regulatory strengths between TF genes and target genes. We believe Bayesian estimation is a suitable framework to account for incorporating diverse types of noisy data and any prior knowledge. A deterministic linear model should be capable of revealing useful insights into larger-size networks.

4.3 Problem Formulation

In this section, incorporation connectivity data, micro-array time series data to the Bayesian framework and related probabilistic models will be explained in detail. Figure (4.1) depicts the pictorial representation of our approach.

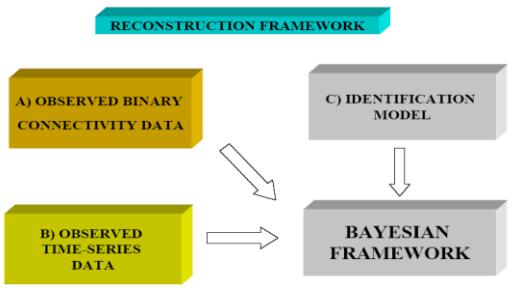


Figure 4.1: Pictorial representation of our reconstruction framework

DNA-protein binding data is the direct information available to understand the regulators involved in transcription. It provides evidence as to whether a regulatory relationship exists through quantification binding of TF protein on DNA. This quantification is achieved through the use of p-test. The binding between a TF and a gene that gives p-test value lower than a certain threshold is assumed to be a regulatory relationship. In a

network with the nodes TF proteins and genes, this corresponds to presence of an edge between the TF and the gene. In [37] and [38], binding data is utilized as the binary connectivity matrix in TF Protein-gene network graph. Therefore, the p-value that is below the threshold is associated with the TF-gene pair having an entry 1 in the binary connectivity matrix. Otherwise, it is equal to zero. In our approach, we employ the similar concept of binary connectivity matrix to incorporate any connectivity data. However, instead of TF-gene connectivity, we assumed gene-to-gene connectivity. Therefore, the binding between a TF and a gene is interpreted as the connectivity between the gene that is producing the TF of interest and the gene being bound by TF (target gene). In [39], protein-DNA binding data is adopted as gene-to-gene connectivity and formulated as an informative prior in Dynamic Bayesian network setting. However, instead of binary connectivity data, they map the p-values to a probabilistic model defining the probability of connectivity between the corresponding gene pairs.

Protein-DNA binding data involves high level of false positives. False positives can be due to two kinds of error. A significant binding observation does not necessarily indicate a regulation relationship. Secondly, noise in observation and threshold selection impose false positives and false negatives. In our model, we adopted a corresponding probabilistic model to account for the error between true binary connectivity and observed binary connectivity which is obtained through binding data. In figure 7, we illustrate true binary connectivity and observed binary connectivity matrices for 4-gene network.

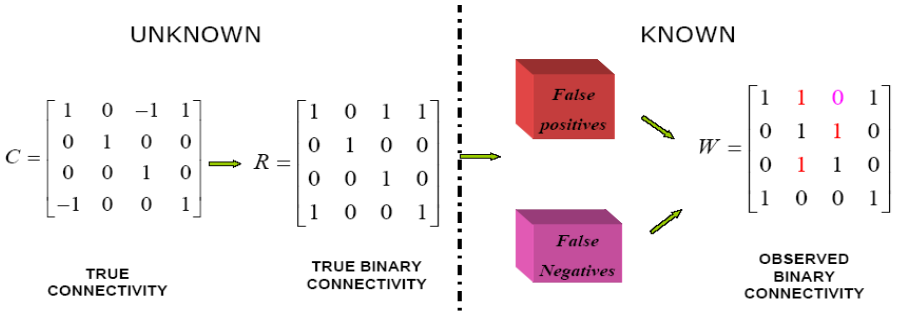


Figure 4.2: Illustration of true binary connectivity and observed binary connectivity

Note that true binary connectivity is not known and one of the aims of reconstruction algorithms is to reveal true connectivity. The discrepancy between true binary, $R = \{r_{ij}\}$ and observed binary connectivity, $W = \{w_{ij}\}$, is modeled according to the following equations;

$$\begin{aligned} &p(w_{ij}=1 \mid r_{ij}=1) = p && p(w_{ij}=0 \mid r_{ij}=1) = 1-p \\ &p(w_{ij}=0 \mid r_{ij}=0) = q && p(w_{ij}=1 \mid r_{ij}=0) = 1-q, \end{aligned} \qquad \textbf{(4.1)}$$

where p and q are true positive and true negative rates respectively.

Time-series micro-array expression experiments are a complementary source of data, which provides dynamic information about the expressions of thousands of genes that are activated or repressed in response to external stimuli [41]. Compared to steady-state data, time series is more appropriate for understanding the dynamics of biological processes.

Let $x^t \in \Re^N$ be a vector of expression levels observed in a micro-array experiment at time point, t. Expression levels for all time points can be written in a matrix form as follows;

$$D^{N\times M} = \left\{ x^1,...,x^t,...,x^M \right\}, \quad \textbf{(4.2)}$$

An extensive list of studies on gene regulatory network modeling, using time-series data, have focused on the continuous-time representation via differential equations. In our study, we also adopted a differential equation model for gene regulation.In this model, the expression level of a gene is assumed to be the concentration of its transcript. The rate of change in the concentration of a particular transcript x_i is given by a function f_i whose arguments are concentrations of other RNA species and parameters;

$$x_i(t) = f_i(x_1,...,x_N,\theta_i) \qquad i = 1,...,N, \quad \textbf{(4.3)}$$

where N is the number of transcripts in the network. In general, only a small subset of all RNA species participates in the regulation of a particular transcript. In other words, the networks are sparse [2]. Different forms for f_i can be adopted. We adopted a linear form for f_i, and discrete form of equation (4.3). The model becomes as follows;

$$x_i(t+1) = \sum_{j=1}^{N} a_{ij} x_i(t), \quad \textbf{(4.4)}$$

where $x_i(t)$ expression level of i^{th} gene, and a_{ij} is the regulatory strength between i^{th} and j^{th} genes. A positive value for a_{ij} indicates activation of j^{th} gene expression by gene i, a negative values corresponds to inhibition of expression level of j^{th} gene by i^{th}, and a zero value shows that there is no regulatory relationship between i^{th} and j^{th} genes. The model can be written for all expression levels.

$$x(t+1) = A \times x(t), \quad \textbf{(4.5)}$$

In equation (4.5), $x(t) \in \Re^N$ is a column vector of expression levels of N genes at time point, t , and $A^{N \times N} = \{a_{ij}\}$ is the regulatory strength matrix. For all time points, equation (4.5) becomes;

$X' = AX$,

$$X = \{x(1),..,x(M-1)\} \quad X' = \{x(2),..,x(M)\} \quad X, X' \in \Re^{N \times (M-1)} \qquad \textbf{(4.6)}$$

where X and X' are obtained from micro-array data matrix, $D^{N \times M} = \{x^1,..,x^t,...,x^M\}$ given in equation (4.2). M is the number of time points. (The number of micro-array experiments).

All data that is obtained from true model placed into a Bayesian estimation framework with appropriate probabilistic models. We seek to infer the posterior distribution for the parameters (elements of activity matrix), $A = \{a_{ij}\} \in \Re^{N \times N}$. Doing this we are trying to recover the presence, strength and nature of the linkages among the nodes in true network. Using the Bayes rule, the posterior distribution over the activity matrix**,** A can be represented as follows;

$$\Pr(A \mid D,W) = \frac{\Pr(D \mid A,W) \times \Pr(A \mid W)}{\Pr(D \mid W)}, \qquad \textbf{(4.7)}$$

where $\Pr(A \mid D,W)$ is the posterior distribution on regulatory strength matrix given micro-array data, D and observed binary connectivity, W. $\Pr(A \mid W)$ is the prior on regulatory matrix, A given the observed binary connectivity. $\Pr(D \mid A,W)$ is the likelihood of the micro-array data given the regulatory strength matrix, A. In the next several paragraphs, each term and their mathematical formulations will be explained in detail.

The prior term, $\Pr(A|W)$ is decomposed as follows;

$$\Pr(A|W) = \Pr(A|R) \times \Pr(R|W), \quad \textbf{(4.8)}$$

where $R = \{r_{ij}\}$ is the unknown true binary connectivity introduced in the Bayesian framework. $\Pr(R|W)$ is simply the error model between true and observed connectivity matrix which is given in equation (4.1). The model can be rewritten in matrix form as follows;

$$\Pr(R|W) = \prod_{i=1}^{N}\prod_{j=1}^{N} w_{ij}\left\{ p^{r_{ij}}(1-p)^{1-r_{ij}} \right\} + (1-w_{ij})\left\{ q^{1-r_{ij}}(1-q)^{r_{ij}} \right\}, \quad \textbf{(4.9)}$$

where p and q are the rates of true positives and false negatives respectively. It is assumed that error for each pairs of genes are independently distributed. $\Pr(A|R)$, on the hand, is the probability of regulatory strengths given the binary connectivity matrix. It models the regulatory strength in each pair of genes, a_{ij}, given the presence or absence of a connection between them. The term, a_{ij}, is assumed to have a Gaussian distribution with mean zero and variance, σ_1 and variance σ_2 for given $r_{ij} = 1$ and $r_{ij} = 0$ respectively.

$$\Pr(a_{ij} | r_{ij} = 1) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left\{ -\frac{a_{ij}^2}{2\sigma_1^2} \right\} \quad \Pr(a_{ij} | r_{ij} = 0) = \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left\{ -\frac{a_{ij}^2}{2\sigma_2^2} \right\}, \quad \textbf{(4.10)}$$

One can write the probability, $\Pr(a_{ij} | r_{ij} = 1)$ as follows;

$$\Pr(a_{ij} | r_{ij}) = r_{ij}\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left\{ -\frac{a_{ij}^2}{2\sigma_1^2} \right\} + (1-r_{ij})\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left\{ -\frac{a_{ij}^2}{2\sigma_2^2} \right\}, \quad \textbf{(4.11)}$$

$\Pr\left(a_{ij} \mid r_{ij}\right)$ is assumed to be independent for each pair of genes, then we obtain the following expression;

$$\Pr\left(A \mid R\right) = \prod_{i=1}^{N}\prod_{j=1}^{N} r_{ij} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{-\frac{a_{ij}^2}{2\sigma_1^2}\right\} + \left(1-r_{ij}\right)\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left\{-\frac{a_{ij}^2}{2\sigma_2^2}\right\} \tag{4.12}$$

Using equations (4.9-12), equation (4.8) is rewritten as follows;

$$\begin{aligned}\Pr\left(A \mid R\right) = &\prod_{i=1}^{N}\prod_{j=1}^{N} r_{ij} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{-\frac{a_{ij}^2}{2\sigma_1^2}\right\} + \left(1-r_{ij}\right)\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left\{-\frac{a_{ij}^2}{2\sigma_2^2}\right\} \\ &\times \prod_{i=1}^{N}\prod_{j=1}^{N} w_{ij}\left\{\, p^{\,r_{ij}}\left(1-p\right)^{1-r_{ij}}\right\} + \left(1-w_{ij}\right)\left\{q^{1-r_{ij}}\left(1-q\right)^{r_{ij}}\right\}\end{aligned} \tag{4.13}$$

The likelihood term in Bayesian formula, $\Pr\left(D \mid A\right)$ has Gaussian distribution with zero mean according to the linear discrete time model given in equation (4.6). The corresponding probabilistic model is given as follows;

$$\Pr\left(D \mid A\right) = \Pr\left(X' \mid A, X\right) \propto \prod_{i=1}^{N}\prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\left(x_{il} - \sum_{k=1}^{N} a_{ik} x_{kl}\right)^2}{2\sigma^2}\right\}, \tag{4.14}$$

The final form for the Bayesian framework is written as follows and contains the unknown true connectivity matrix, R ;

$$\Pr\left(A, R \mid D, W\right) \propto \Pr\left(X' \mid A, X\right) \times \Pr\left(A \mid R\right) \times \Pr\left(R \mid W\right)$$

$$\Pr\left(A \mid R\right)=\prod_{i=1}^{N}\prod_{j=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{\left(x_{il}-\sum_{k=1}^{N}a_{ik}x_{kl}\right)^{2}}{2\sigma^{2}}\right\}$$
$$\times\prod_{i=1}^{N}\prod_{j=1}^{N}r_{ij}\frac{1}{\sqrt{2\pi}\sigma_{1}}\exp\left\{-\frac{a_{ij}^{2}}{2\sigma_{1}^{2}}\right\}+\left(1-r_{ij}\right)\frac{1}{\sqrt{2\pi}\sigma_{2}}\exp\left\{-\frac{a_{ij}^{2}}{2\sigma_{2}^{2}}\right\} \quad \textbf{(4.15)}$$
$$\times\prod_{i=1}^{N}\prod_{j=1}^{N}w_{ij}\left\{\ p^{r_{ij}}\left(1-p\right)^{1-r_{ij}}\right\}+\left(1-w_{ij}\right)\left\{q^{1-r_{ij}}\left(1-q\right)^{r_{ij}}\right\}$$

Solution methodology

To solve the complicated expression of equation (4.15), a Gibbs Sampler approach is employed which samples each component, A and R at a time while fixing the other components. As maximum a posteriori network is sought, the formulation is modified accordingly. In the first step of each iteration, the binary connectivity matrix is held fixed and maximum a posterior estimation on A is calculated according to the following formula;

$$\arg\max_{A}\Pr\left(A \mid D,R\right)=\arg\max_{A}\left[\Pr\left(A \mid X',X\right)\times\Pr\left(A \mid R\right)\right] \quad \textbf{(4.16)}$$

One can take the natural log of the right hand side of equation (4.16) and drop the minus sign. The equation becomes as follows;

$$\arg\max_{A}\Pr\left(A \mid D,R\right)=\arg\max_{A}LOG\left[\Pr\left(A \mid X',X\right)\times\Pr\left(A \mid R\right)\right]$$

$$\arg\min_{A}\left\{\sum_{i}^{N}\sum_{l}^{M}\left\{\frac{\left(x_{il}-\sum_{k=1}^{N}a_{ik}x_{kl}\right)^{2}}{2\sigma^{2}}\right\}+\sum_{i}^{N}\sum_{i}^{N}r_{ij}\frac{a_{ij}^{2}}{2\sigma_{1}^{2}}+\sum_{i}^{N}\sum_{j}^{N}\left(1-r_{ij}\right)\frac{a_{ij}^{2}}{2\sigma_{2}^{2}}\right\} \quad \textbf{(4.17)}$$

Equation (4.17) is nothing but regularized least squares solution with weights proportional to the inverse square of variances.

In each iteration, the second step is the maximum a posteriori estimation of binary connectivity matrix, R when regulatory matrix, A is held fixed at its value calculated in the previous step.

$$\arg\max_R \left[\Pr(R \mid W, A)\right] = \arg\max_R \left[P(A \mid R) \times P(R \mid W)\right]$$

$$= \arg\max_R \left[\begin{array}{l} \prod_{i=1}^{N}\prod_{j=1}^{N}\left(r_{ij}\left(\frac{1}{\sigma_1}\exp\left\{-\frac{a_{ij}^2}{2\sigma_1^2}\right\}\right) + (1-r_{ij})\left(\frac{1}{\sigma_2}\exp\left\{-\frac{a_{ij}^2}{2\sigma_2^2}\right\}\right)\right) \\ \times \prod_{i=1}^{N}\prod_{j=1}^{N} w_{ij}\left\{ p^{r_{ij}}(1-p)^{1-r_{ij}}\right\} + (1-w_{ij})\left\{q^{1-r_{ij}}(1-q)^{r_{ij}}\right\} \end{array} \right] \quad \textbf{(4.18)}$$

4.4 Results

To evaluate the performance of the algorithm, we created an ensemble of 100 artificial networks. A linear discrete time equation model is assumed for the created networks. (See equation (4.4)). In each network, all genes are perturbed randomly, and while the system is reaching to a new steady state, measurements are taken for all genes at certain time points. We introduced false positive and negative error on the topology to simulate noisy prior connectivity data. Furthermore, measurement noise is added to the data with varying degree. This procedure is depicted in figure 4.3

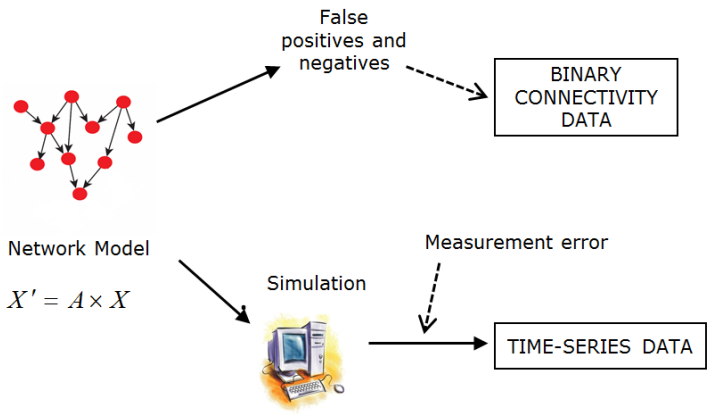


Figure 4.3 Illustration of data collection and simulating topological and measurement errors.

The prediction power of the method is calculated through two performance metrics. The first one is defined as the fraction of the eliminated topological errors. It is represented as follows;

$$\varepsilon_T = 1 - \frac{\sum_i^N \sum_j^N \left| w'_{ij} - r_{ij} \right|}{\sum_i^N \sum_j^N \left| w_{ij} - r_{ij} \right|} \qquad \textbf{(4.19)}$$

In equation (4.19), $R = \{r_{ij}\}$ is the true connectivity matrix, $W = \{w_{ij}\}$ is the initial observed topology and $W' = \{w'_{ij}\}$ is the estimated topology. This is represented as follows;

$$\begin{aligned} &\textit{if} \quad \left\| \hat{a}_{ij} \right\| > \delta \qquad w'_{ij} = 1 \\ &\textit{otherwise} \qquad w'_{ij} = 0 \end{aligned} \qquad \textbf{(4.20)}$$

where $\hat{a}_{ij}$ is the estimated regulatory strength. If it is greater than a certain threshold, δ then the estimated connection between i^{th} and j^{th} gene is equal to 1, otherwise it is zero. The second performance metrics is the relative error in parameters. This is defined as the Frobenius norm of the relative difference between calculated and true parameter matrices.

$$\varepsilon_F = \frac{\left\| \hat{A} - A \right\|}{\left\| A \right\|} \qquad \textbf{(4.21)}$$

The average percentage errors for the ensemble of 100 randomly created artificial networks are calculated and plotted against different number of measurements. Each network has 10 genes and 30 connections. For each network, topological error is introduced as follows; There are total of 12 errors. 3 random connections are deleted (false negatives) and 9 false connections (false positives) are added. The multiplicative error variance in samples is changed in the range of 5-20% of the signal. In each experiment, 8 time points are sampled and the number of experiments changed from 1-6.

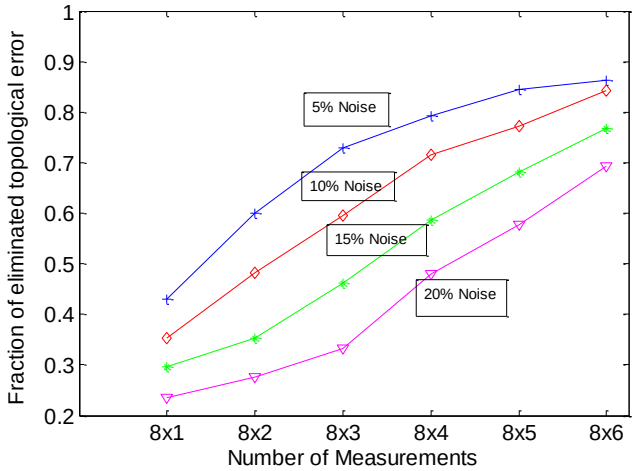
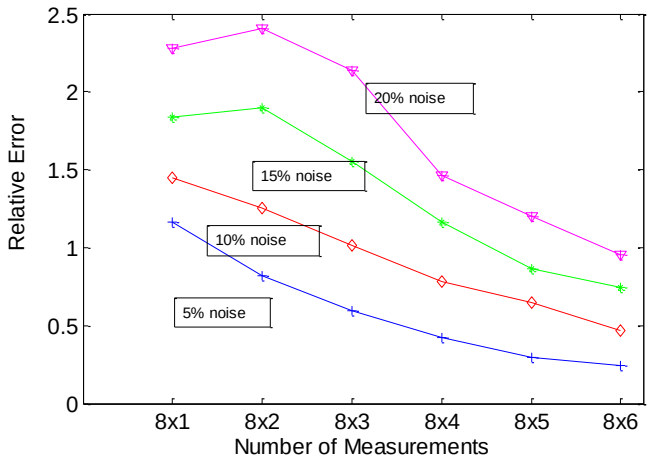


Figure 4.4 Fraction of eliminated topological error for different levels of measurement error and time points. There are 3 false negatives and 9 false positives.

It can be seen from figure (4.4) that as number of data points is increased the fraction of topological error that is eliminated increases. Furthermore, the prediction power decreases with the increasing noise levels, which is expected. For example, in 20% noise case, 6 parallel experiments with 6 time points in each are required to eliminate 80% percent of the topological errors.

In the next step of analysis, we looked into the performance in estimating the value of parameters. This is done by plotting the relative error in parameters with respect to the number of data and varying multiplicative noise levels.



.

Figure 4.5: Relative error with respect to various number of data points.

As it is seen from figure (4.5), relative error decreases with increasing number of data. Furthermore prediction power decreases with the increasing noise level. For 20% noise case, the relative error is approximately equal to 1 for 6 parallel experiments and 6 data point in each experiment. Though 80% of topological errors are eliminated for this case (see figure 4.4), relative error in parameters is 1.

In the second part of analysis total number of topological error is fixed at 12 again, however we introduced 9 false negatives and 3 false positives. Essentially we

increased the number of false positive errors. As it can be seen from figure 4.6 the performance of eliminating topological error has significantly decreased even though number of error remain constant at 12. Relative error in parameters stayed high and didn't change significantly.

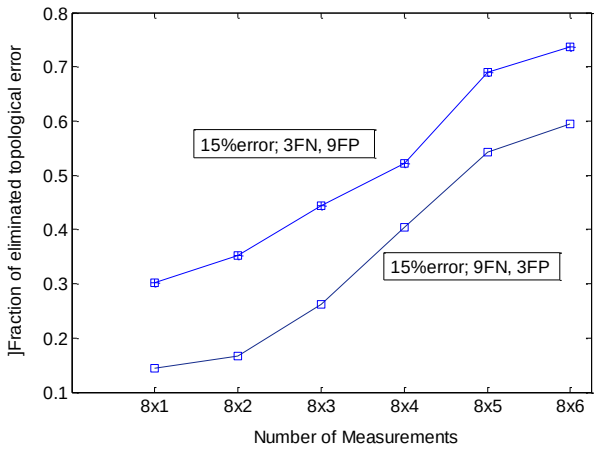


Figure 4.6 Fraction of eliminated error with 15% measurements error for two different cases of topological errors. The curve with lower error elimination rate has 9 false negatives and 3 false positives. The curve with higher error elimination rate has 3 false negatives and 9 false positives.

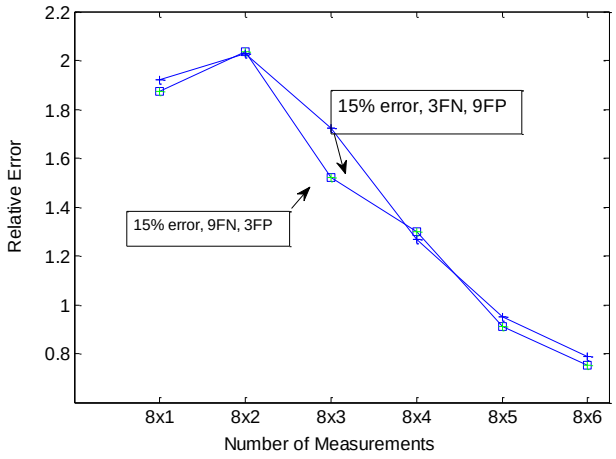


Figure 4.7 Relative error with 15% measurements error for two different cases of topological errors. The curve with lower error elimination rate has 9 false negatives and 3 false positives. The curve with higher error elimination rate has 3 false negatives and 9 false positives.

4.5 Conclusions

Binding data provides an initial topology of the gene networks. However, this data suffers from high rate of false positive and negative errors. In these results, we see that our algorithm is able to fuse connectivity data and micro-array data and to approach true network topology and dynamics provided that enough data is available. However, for reasonable levels of noise in the data (10-20% multiplicative noise) the error in parameters is still quite high. This raises the issues of additional prior information requirement and optimal experimental design. Furthermore, better error models for both micro-array and binding data may significantly decrease the data requirements. The major problem in topology data is the false negatives, in other words, missing links in the prior topology. It is important to be able to identify missing links in the prior information. To this end in next chapter we concentrated on link prediction techniques using network's topological distribution data. We introduced a novel link prediction method that is based on local connectivity information.

CHAPTER 5

LINK PREDICTION THROUGH NETWORK CONNECTIVITY USING LITERATURE MINING DATABASES

5.1. Summary

In the previous chapter we introduced a Bayesian method that integrates connectivity and time series data. We concluded that in order to utilize the network identification methods to their full extent more accurate prior connectivity data is needed. The major problem in identifying connectivity is false negatives, in other words, missing links in the network. In this chapter we focus on identifying missing links based on existing connectivity information. We demonstrate finding missing links in protein target identification using literature mining data. Target identification is very crucial in pharmaceutical research. The pharmaceutical industry frequently brings a new promising target to clinical trials only to find that it has serious safety concerns or lack of efficacy. A gene downstream or upstream in the targeted pathway can serve as a remedy, however, finding such an alternative target using existing in-silico or bench tools can be extremely labor-intensive. Recently, increasing amounts of information and observations have been compiled from different areas of biological research and deposited on databases. In this work we propose a novel computational method to quantify indirect relationships between the objects of biological research of interest by using existing relationships from text mining databases to automate the search for novel biological targets. We apply our method to analyze 10850 proteins in Ariadne database and created a rank-ordered list of protein pairs that are most similar to each other. This list can potentially guide researchers in the

effort of identifying novel targets that are most similar to the existing unsafe or inefficient targets. We compared the prediction power of our method with the Jaccard and Common Neighbors similarity scores. Our method outperformed both methods in predicting the links for 10850 proteins in the database.

5.2 Introduction

Biological processes are the result of interactions involving hundreds of thousands of molecular entities. These interactions form complex networks. In a biological network a node represents a biological entity and a link refers to association between two biological entities. This association can be a physical link, a functional similarity or an implicit relation. It is becoming increasingly important to approach biological problems from a perspective of networks and identify missing links in the network. To understand diseases and find new drug targets in a systematic way, it is critical to scrutinize the topology of these networks.

Link prediction methods in complex networks have attracted increasing attention from computer scientists and physicists [57-60]. These methods usually aim at estimating likelihood of the existence of a link between two nodes based on observed links and the attributes on the nodes [61]. Link prediction methods can be classified into two categories; the first is prediction of existing but unknown links; the second is prediction of future links. Biological link prediction falls into the first category [49]. Discovery of links in biological networks can be costly and time consuming through experimental means. Making predictions based on the existing links instead of blindly checking all links can considerably reduce the cost, both in time and money. Furthermore, missing links can be a major drawback in dynamic modelling of the networks. Prediction of

missing links can be potentially used as a tool to improve connectivity information for reverse engineering of networks. There are many biological applications of link prediction methods. One of them is the target identification. A biological target may refer to a protein that has been intended to be a target of a drug.

Drug-target interaction (interactions between drugs and target proteins) is a key area in drug discovery. Both number of new drugs and targets hasn't changed significantly in the last 20-25 years [51]. In target identification problem a protein in the downstream or upstream of the network might be an alternative to an existing inefficient target, however, not all pathways are known, and finding such an alternative target using existing in-silico or bench tools can be extremely labor-intensive. A method that can automatically find implicit relationships between network nodes (proteins, diseases, drugs, compounds etc.) can be invaluable in the search of new target. There are many studies on target identification problem. Yamanishi et al [52] integrated known drug-target information with target protein sequence data and drug chemical structure and proposed as a new target-identification tool. In another study, Campillos et al [53] used side effect similarity between drugs to predict novel targets for drugs. They based their method on the assumption that the drugs that share common side effects are more likely to share targets. They combined drug-target, drug-side effect information with drug chemical similarity. They also validated some of their predictions with experimental results. These studies are based on finding target space for existing drugs. In this study we try to find novel targets that are most similar to existing targets based on its connectivity in networks that are obtained from literature mining databases. Increasing amount of information is compiled into biological network format. Text mining is the

automated way of collecting the relationships between biological entities through co-occurrences within electronically available records [61]. Text mining aims at collecting and retrieving useful hidden relations from these resources of information. Therefore, text mining databases represent different sets of pre-compiled information on biological relationships and associations, interactions and facts which have been extracted from the biomedical literature.

In this chapter we propose a novel computational method of drug target discovery by quantifying indirect relationships between the nodes of biological networks using interactions retrieved from mining databases. This method can also be used to annotate diseases with similar etiology, reposition existing drugs, or discover adverse events for the targets.

5.3. Methods

Our model is based on a computational approach that quantifies the relevance of two biological objects such as genes, proteins, compounds, complexes, drugs, diseases (hereafter referred to simply as "objects" or "entities") by comparing their common connections, obtained through databases against a random network model obtained through the databases. Denoting an object of interest with 'A', one can identify other objects 'B' (See Fig.5.1).

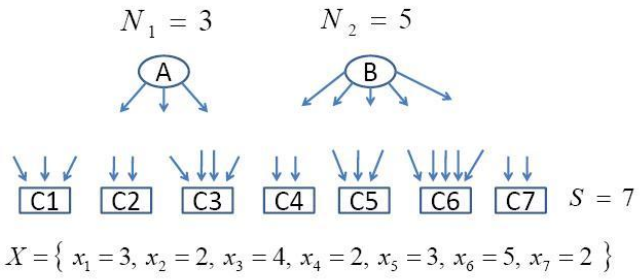


Figure 5.1. Random bipartite network model for entities A and B

The text mining database information can be represented as a directed bi-partite network. In graph theory a bipartite graph is graph whose vertices can be divided into two sets. This is a directed graph where the relations between the nodes are represented as arrows with originating from a source node and ending in a sink node. Out-degree of a source node in a directed graph is the number of edges (arrows) originating from the node and in-degree of a sink node is the number of arrows ending in a sink node. In other words an out-degree is the number of distinct objects that a source node (first set object) is effecting and in-degree is the number of distinct objects a sink node (second set object) is being effected by a source node(first set object). Figure (5.1) is a directed bi-partite graph model where the random network model is sought using the parameters of the network. It consists of two sets of nodes. The first set nodes are the source nodes A and B and the second set nodes are the sink nodes $C_1,...,C_7$. Each node in both sets refers to certain biological object. The parameters are the out-degrees of the pair of the entities A and B and in-degrees of objects in the second set along with the number of entities in this set. Out-degrees of A and B are represented with N_1 and N_2, whereas in-degrees are denoted as $\{x_1, x_2,.., x_S\}$. S is the total number of entities in the second set of the bi-

partite graph. Let us denote the parameter set that we obtain from the database with, $\theta = \{N_1, N_2, x_1, x_2, ..., x_S, S\}$.

Random graph construction is a method to model the possible ways for A and B to connect to the objects of the second set. This allows us to quantify the likelihood of A and B having common downstream objects when drawn from a random graph. We compare the observed common downstream connections in the database against this random graph model to quantify the similarity between A, and B. If A and B in the database graph share substantially more downstream connections than would be predicted by a purely random graph, then we have evidence to suggest that they are similar.

Let us define the two different events on this bi-partite graph. The first event is the number of common entities to which A and B are connected and the second event is the set of in-degrees of these common entities. The joint probability of these two events can be represented with following expression;

$$P(M = \{ m_1, m_2, ..., m_k \}, i = k \mid \theta) \quad \textbf{(5.1)}$$

,where M is the list of the in-degrees of the common downstream entities, and i is the number of common entities. Using the definition of joint probability distribution, one can write the following equation;

$$P(M = \{ x_1, x_2, ..., x_k \}, i = k \mid \theta) = P(i = k \mid \theta) . P(M = \{ x_1, ..., x_k \} \mid i = k, \theta) \quad \textbf{(5.2)}$$

In this equation $P(i = k \mid \theta)$ is the probability of first event given the parameters, and $P(M = \{ x_1, ..., x_k \} \mid i = k, \theta)$ is the conditional probability of second event conditional on the first event given the parameters.

The first term in this equation, $P(i=k\,|\,\theta)$ can be derived as a function of the parameters; N_1, N_2 and S. In figure (5.2), a random configuration of bipartite graph for A and B is shown. Connections that are common for the pair are represented with solid, whereas node specific connections are displayed by dashed lines.

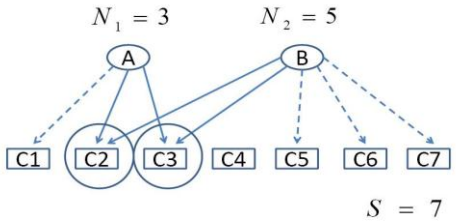


Figure 5.2. Example of A and B having common second set entities, C_2, C_3.

In order to derive the probabilistic distribution for the number of common objects that A and B share, we start with enumeration of different possibilities. The number of combinations of the N_1 connections that A can make with S different second set objects is calculated by the following;

$$C(S,N_1)=\frac{S!}{N_1!(S-N_1)!} \qquad \textbf{(5.3)}$$

One can obtain similar equation for B;

$$C(S,N_2)=\frac{S!}{N_2!(S-N_2)!} \qquad \textbf{(5.4)}$$

Let L denotes the total number of combinations of A and B connections to any N_1 and N_2 second set objects. L can be calculated as the multiplication of the combinations of both cases;

$$L = C(S, N_1) \,.\, C(S, N_1) = \frac{S!}{N_1!(S-N_1)!} \;\; \frac{S!}{N_2!(S-N_2)!} \tag{5.5}$$

The number of combinations for A and B having k common downstream objects (second set objects or sink nodes) can be represented as follows;

$$C(S,k) = \frac{S!}{k!(S-k)!} \tag{5.6}$$

Once k connections of A and B are fixed, they have $(N_1 - k)$ and $(N_2 - k)$ connections remaining respectively. The number of objects available in the second set is reduced to $(S-k)$. The number of ways that the remaining connections of A could be chosen out of $(S-k)$ entities can be calculated as follows;

$$C\,(\,(S-k),(N_1-k)) = \frac{(S-k)!}{(N_1-k)!(S-N_1)!} \tag{5.7}$$

This will fix the number of all N_1 connections of A and there will be $(S-N_1)$ objects left for (N_2-k) remaining connections of B. The number of combinations for remaining connections of B for the remaining objects is represented as follows;

$$C\,(\,(S-N_1),(N_2-k)) = \frac{(S-N_1)!}{(N_2-k)!\,(S-N_1-N_2+k)!} \tag{5.8}$$

The overall number of combinations that A and B are connected to k common objects will be the denoted with D. It can be written as;

$$\begin{aligned} D &= C\,(\,S,k\,)\,C\,(\,(S-k),(N_1-k))\,C\,(\,(S-N_1),(N_2-k)) \\ &= \frac{S!}{k!(S-k)!}\,\frac{(S-k)!}{(N_1-k)!(S-N_1)!}\,\frac{(S-N_1)!}{(N_2-k)!\,(S-N_1-N_2+k)!} \end{aligned} \tag{5.9}$$

The probability that A and B are connected to k common objects is the ratio of the total number of combinations of A and B are connected to k objects in common to the total number of combinations that the pair is connected to objects in any possible way. This probability is written as;

$$P(i=k\mid\theta)=\frac{D}{L}=\frac{C(S,k).C((S-k),(N_1-k)).C((S-N_1),(N_2-k))}{C(S,N_1).C(S,N_2)}$$
$$=\frac{\frac{S!}{k!(S-k)!}\frac{(S-k)!}{(N_1-k)!\cdot(S-N_1)!}\frac{(S-N_1)!}{(N_2-k)!\cdot(S-N_1-N_2+k)!}}{\frac{S!}{N_1!(S-N_1)!}\frac{S!}{N_2!(S-N_2)!}} \qquad \textbf{(5.10)}$$

After cancelations, we obtain;

$$P(i=k\mid\theta)=\frac{N_1!.\,N_2!\,(S-N_1)!.(S-N_2)!}{S!\,k!\,(N_1-k)!\,(N_2-k)!\,(S-N_1-N_2+k)!} \qquad \textbf{(5.11)}$$

This expression can be approximated by a Poisson distribution.

$$P(i=k\mid\theta)=\frac{1}{\alpha}\frac{\lambda^{i}e^{-\lambda}}{i!} \qquad \alpha=\sum_{i=0}^{\min(N_1,N_2)}\frac{\lambda^{i}e^{-\lambda}}{i!} \qquad \lambda=\frac{N_1N_2}{S} \qquad \textbf{(5.12)}$$

,where λ is a function of N_1, N_2 and S while α is the normalization factor. It normalizes the cumulative distribution to one at $i=\min(N_1,N_2)$ as the probability is not defined beyond this point. This approximation allows us to obtain a compact representation for the probability term. It is less computationally intensive. The aim is to derive a compact similarity score function between two objects that makes sense intuitively starting from a formal probabilistic framework.

To check the validity of the approximation we calculated sum of absolute deviation of the equation (11) from the Poisson approximation for all possible values of

$i = \{0,1,...,\min(N_1, N_2)\}$ at different values of N_1 and N_2 . This corresponds to the deviation of cumulative distributions for Poisson and equation (5.11). We defined the percentage deviation as follows;

$$E(N_1, N_2, S) = 100 * \sum_{i=1}^{\min(N_1, N_2)} \left| \left(\frac{1}{\alpha} \frac{\lambda^i e^{-\lambda}}{i!} \right) - \left(\frac{N_1! \cdot N_2! \,.\, (S-N_1)! \,.\, (S-N_2)!}{S! \,.\, i! \,.\, (N_1 - i)! \,.\, (N_2 - i)! \cdot (S - N_1 - N_2 + i)!} \right) \right| \quad \textbf{(5.13)}$$

In figure (3), we illustrated the calculation the value of E on an example. The absolute deviation of Poisson approximation corresponds to the sum of lengths of the dotted lines.

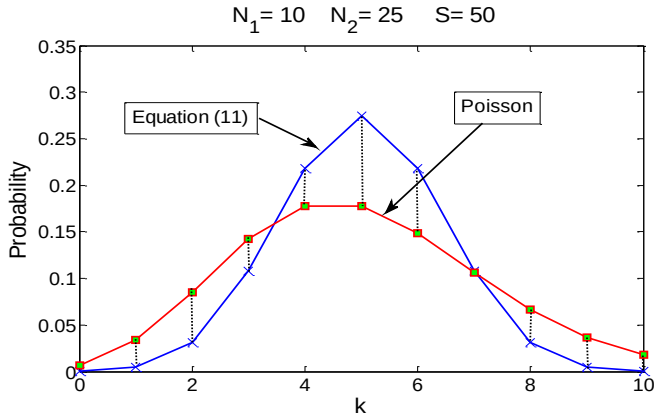


Figure 5.3. Deviation of the model from Poisson distribution for $N_1 = 10, N_2 = 25, S = 50$

In figure (5.4), the curves for various E values are shown at different values of N_1 and N_2. The area under each curve shows the region for the values of N_1 and N_2 where Poisson approximation exceeds the given percentage deviation. For example the deviation of Poisson approximation is less than 10% when one of the objects has four or more connections ($N_1 \geq 4$) and the other object connected to less than 34% of all second layer objects ($N_2 \leq 0.34 \times S$). N_1 and N_2 can be used interchangeably and the area under

the curves remain same for different values of S . This figure shows that Poisson distribution is a reasonably good approximation for a large span of N_1 and N_2 values.

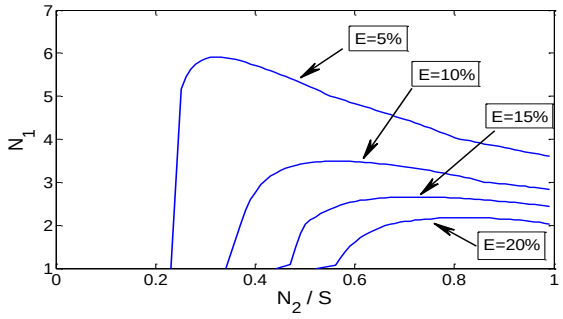


Figure 5.4. Deviation of model fromPoisson approximation for different N_1 and N_2 values.

One can also derive the conditional probability term on the right hand side of equation (2). In figure (5.5), a possible connection pattern is shown for illustration purposes. k is the number of shared entities between A and B (in this example there are two common entities) , M shows the list of common entities and X is the set of in-degree values for these commonly shared objects.

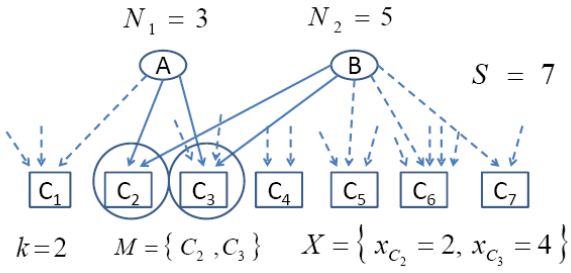


Figure 5.5. Example of A and B having common second set entities, C_2, C_3 with their in-degrees.

Let us consider the general case for k common objects. The number of possible ways for A and B to be both connected to a particular second set object (sink node) with

a given in-degree of x_i is equal the number of 2-combinations of x_i . In other words, it is the number of combinations that two objects (A and B) can be connected to a particular object that is known to have x_i objects connected to it. It can be calculated as;

$$c_i = C(x_i, 2) = \frac{x_i(x_i - 1)}{2} \quad \textbf{(5.14)}$$

In this equation the number of combinations where A and B are both connected i^{th} object is denoted by c_i .

The number of possible connections of A and B to any k objects with known in-degrees in the downstream is written as follows;

$$Z = \sum_{i=1}^{S} c_i \sum_{j=1}^{S} c_j \ldots \sum_{z=1}^{S} c_z \quad \textbf{(5.15)}$$

In this equation there are k embedded summation terms corresponding to k common objects. Each common object can be chosen out of S different objects. For large S , this summation term would be difficult to calculate. Therefore we introduce the following approximation.

$$c_1 = c_2 = \ldots = c_S = \hat{c} \quad \textbf{(5.16)}$$

Here we assume that all c_i terms are equal to an average $\hat{c}$ term. If (16) is plugged into expression (15), we obtain the following approximation,

$$Z \approx \sum_{i=1}^{S} \hat{c} \sum_{j=1}^{S} \hat{c} \ldots \sum_{z=1}^{S} \hat{c} = S^k \hat{c}^k \quad \textbf{(5.17)}$$

One can represent the number possible ways that A and B are connected to k particular objects as follows;

$$T = \prod_{i=1}^{k} c_i \qquad \textbf{(5.18)}$$

The probability that A-B pair are connected to k particular objects is calculated as the ratio of the number of combinations that this pair is connected to k particular objects to the number of different ways that they are connected to any k objects.

$$P\left(M = \{ m_1, m_2, ..., m_k \} | , i = k, \theta \right) = \frac{T}{Z} = \frac{\prod_{i=1}^{k} c_i}{S^k \hat{c}^k} \qquad \textbf{(5.19)}$$

Plugging expression (5.19) and (5.12) into expression (5.2), we obtain

$$P\left(M = \{ m_1, m_2, ..., m_k \}, i = k \mid \theta \right) = \frac{1}{\alpha} \frac{\lambda^k e^{-\lambda}}{k!} . \frac{\prod_{i=1}^{k} c_i}{S^k \hat{c}^k} \qquad \textbf{(5.20)}$$

This equation gives us the probability of two entities having k common downstream objects from the set M. It is derived based on a random bi-partite network model using the parameter set, θ. The similarity between the pair of entities; A and B is assumed to be based on the statistical significance of their common connections according to the probability of occurrence in a random network model. To quantify the significance of an observed connectivity structure of the pair that has common downstream entities, we defined the following score function;

$$Score = -\log\left(P\left(M = \{ m_1, m_2, ..., m_k \}, i = k \mid \theta \right)\right) = -\log\left(\frac{1}{\alpha} \frac{\lambda^k e^{-\lambda}}{k!} . \frac{\prod_{i=1}^{k} c_i}{S^k \hat{c}^k} \right) \qquad \textbf{(5.21)}$$

Hence, the lower the probability of occurrence for a random model is, the more significant the event is and therefore the higher the score. One can write score in an open form as follows;

$$Score = \log(\alpha) + \log(k!) - k\log(\lambda) + \lambda + k\log(S.\hat{c}) - \sum_{i=1}^{k} \log(c_i) \quad \textbf{(5.22)}$$

One can use Sterling approximation for the term, $\log(k!)$;

$$\log(k!) \approx k\log(k) - k \quad \textbf{(5.23)}$$

Using (5.23), expression for λ in (5.12) and rearranging the terms, expression (5.22) can be rewritten as follows;

$$Score = (\log(\alpha) + \lambda - k) + \sum_{i=1}^{k} \log\left(\frac{S}{N_1}\frac{S}{N_2}\frac{\hat{c}}{c_i}k\right) \quad \textbf{(5.24)}$$

This equation can be further simplified by the following assumption;

$$(\log(\alpha) + \lambda - k) << \sum_{i=1}^{k} \log\left(\frac{S}{N_1}\frac{S}{N_2}\frac{\hat{c}}{c_i}k\right) \quad \textbf{(5.25)}$$

Finally, one can obtain the following expression;

$$Score = \sum_{i=1}^{k} \log\left[\frac{S}{N_1}\frac{S}{N_2}\frac{\hat{c}}{c_i}k\right] = k.\log\left(S^2\hat{c}\right) + \sum_{i=1}^{k} \log\left[\frac{k}{c_i N_1 N_2}\right] \quad \textbf{(5.26)}$$

This function gives us the similarity score between A and B based on the network structure and properties. In this expression, $\log\left(S^2\hat{c}\right)$ is a network domain-dependent constant. A network domain can be defined as part of the network with all biological interactions of a certain type. Examples of such domains can be transcriptional regulation, protein binding, protein modification and any other biological function that connects one biological entity to another. Each domain might have different number of second layer entities (S) and connectivity structure (c).

One can see that the similarity score is directly proportional to number of common downstream objects, k. This is an expected result as one expects two entities to be similar when they have more common downstream effects. Score is also inversely

proportional to both N_1 and N_2. This can be interpreted as the more connected the species are the more likely they have common downstream effect by chance. Finally, the score is inversely proportional to in-degree of the common objects connected to the pair. This is the result of the fact that the pair will more likely to have common downstream entities that have high in-degree by chance. Hence, this commonality gives relatively lower significance for the similarity.

5.4 Results

We applied our algorithm to the Ariadne database. Ariadne is aSystems Biology software that consists of computational methods to generate databases from the literature. Ariadne database represent different sets of biological relationships which have been extracted from the biomedical literature [63]. To test our algorithm we extracted 10850 proteins and approximately 200,000 protein-protein associations from Ariadne. Each association corresponds to a biological mechanism, such as; binding, regulation, activation, inhibition, modification, etc. These proteins are grouped into two. First group is the regulators and second group represents the regulatees. We applied our algorithm on each pairs of regulators. Note that our algorithm quantifies the relationship between two regulator proteins depending on the number of common regulates they share, in-degrees of these regulates as well as out-degree of the regulator proteins.(See equation 5.26). A rank list of regulator protein pairs are created, starting from most similar regulator pairs going through the least similar ones. Some regulators are actually connected to each other. This constitutes the probe set. Probe set is not used in prediction algorithms and it is treated as missing links in the network. Prediction methods are compared based on

prediction power on this probe set. To quantify the prediction accuracy we employed receiver operating characteristics (ROC) curve approach. ROC curve quantifies sensitivity and specificity of a prediction method in a systematic way and it is one of the standard methods to measure performance of prediction methods. In the decreasing-order rank list of protein pairs, a threshold value for the similarity score is chosen. The pairs with similarity score higher than the threshold value are taken as positives and the remaining pairs are the negatives. The positive set and negative set is compared to probe set to determine true and false positive rates.

We compared our method with existing similarity index measures. Common Neighbors (CN) and Jaccard (J) methods are the most commonly used similarity scores in link prediction. Common Neighbors method assumes that two nodes are more likely to form or have a link if they have many common neighbors.

$$Score\,(CN) = |\Gamma(x) \cap \Gamma(y)| \qquad \textbf{(2.14)}$$

Here, $\Gamma(x)$ is the set of nodes connected to node-x and similarly $\Gamma(y)$ is the set of neighbors for node-y .

Jaccard is another quantification of similarity index between two nodes in a network. It is a function of out-degrees of the node pair as well as number of common neighbors they have.

$$Score\,(J) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \qquad \textbf{(2.16)}$$

ROC curve comparison of the methods is given in figure (5.6). A roc curve plots true positive rate with respect to false positive rate. The area under the curve shows the prediction power of the corresponding method.

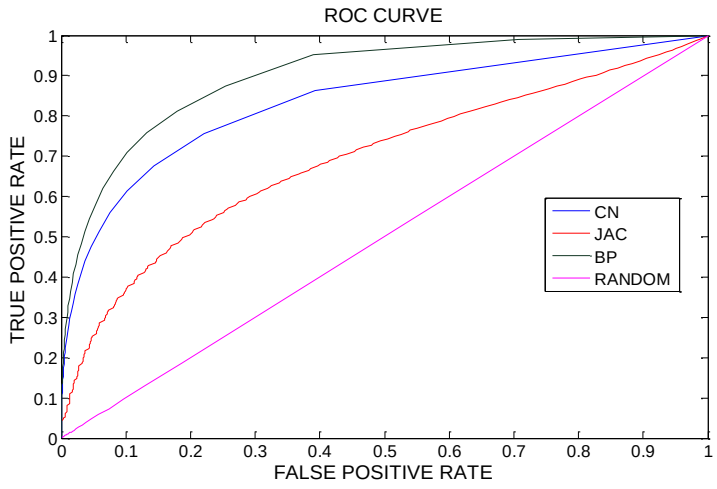


Figure 5.6. Receiver operating characteristics curve for Common Neighbors (CN), Jaccard (JAC) and our bipartite approach (BP). 'Random' curve refers to prediction is made based on completely naïve random approach.

It can be seen from figure (5.6) that our bipartite approach outperforms Jaccard and Common Neighbors methods on the protein-protein interaction network obtained from Ariadne database.

5.5 Conclusions

The contribution of this work can be summarized in two ways. First, our method is a novel computational algorithm to quantify indirect relationships between the objects of biological research of interest by using existing relationships from text mining databases to automate the search for novel drug targets. This method can also be used for different purposes such as; annotating diseases with similar aetiology, reposition of existing drugs, or discovering adverse events for the targets. Secondly, in a case study involving 9575 proteins in the Ariadne database, our method outperformed the Jaccard method for the

prediction of existing links for all proteins. This illustrates its prediction capability for biological networks.

CHAPTER 6

COMPUTATIONAL ADVERSE EVENT PREDICTION THROUGH A NETWORK BASED APPROACH

We introduced a link prediction method in the last chapter and demonstrated its prediction power for protein-protein association networks obtained from literature mining databases. However, this method is applicable to the bipartite networks. In this chapter we will look into the link prediction methods for multipartite networks. One immediate application of this is the adverse event prediction.

6.1 Summary

Adverse event prediction is becoming increasingly important as health authorities focuses on the obligation of the pharmaceutical industry to ensure that marketed drugs have acceptable benefit-risk profiles. Therefore, it is critical for pharmaceutical companies to identify potential toxicity risks of the drugs during early phases of the lengthy drug development process. To date the adverse event prediction methods are mostly concentrated on finding novel targets for the drugs as side effects may be due to the unintended targets of the drugs. Biological pathway knowledge is a crucial source of information that can help predict the side effect profiles of the drugs. Moreover, animal models can give clues on possible adverse events in the early phases of drug development. To the best of our knowledge there has been no systematic network based study for finding significant associations between biological pathways and side effects as well as mouse phenotypes and side effects. In this study we introduced a computational framework for side effect prediction from pathway and mouse phenotype information.

We integrated MGI, KEGG pathway, Drugbank and SIDER information on a multilevel network representation. A p-value based approach is introduced to find significant associations between *pathway-side effect* as well as *mouse phenotype-side effect* pairs. We demonstrated the biological relevance of these associations with two examples. Finally, we validated the prediction power of our method using ROC curves.

6.2 Introduction

An adverse event (side effect) is an unwanted response to a drug that has happened during treatment of patients or clinical trials. Increasing scientific, regulatory research is focused on the obligation of the medical community, pharmaceutical industry and health authorities to ensure that marketed drugs have acceptable benefit-risk profiles. In that regard adverse event prediction methods for drugs become increasingly important.

Drugs bind to target proteins and affect biological pathways, and these pathways cause phenotype effect. An adverse event can be caused by drugs known targets (target effect) or it can be due to proteins that are not yet identified as the targets of the drug (off target effects). Adverse events vary from simple symptoms, such as nausea, to critical symptoms, such as torsades de pointes. Most side effects are harmful to humans, but side effects can also be utilized to find new uses for known drugs. Therefore, it is highly desirable to automatically discover new targets for known drugs and to understand the mechanisms that cause side effects for target-specific treatments. There are several studies concentrated on finding drug targets integrating various information resources. Yamanishi et al [52] integrated known drug-target information with protein sequence data and drug chemical structure to find novel drug targets. In another study, Campillos et

al [53] utilized side effect similarity between drugs to predict novel targets for drugs. Their method based on the assumption that drugs that share common side effects are more likely to shared targets. They validated some of their predictions through experimental results.

Yamanishi's and Campillos's studies were focused mostly on finding off-target proteins causing the side effects. However it is important to consider biological pathways that are affected by the drugs. Proteins in the downstream of a drug's known targets can lead to side effects. The knowledge of a pathway allows separate targeting of upstream or downstream targets. Inhibition or modulation of selected targets in the same pathway could lead to the same therapeutic with fewer side effects or better druggability. Furthermore these targets can crosstalk with other pathways which may be potential sources of the observed side effects. Therefore the knowledge of pathways and their relation to each other helps researchers understand side effect profiles [71]. To the best of our knowledge there has been no systematic study of integrating *pathway-target-side effect* relationships on a network framework to find significant *Target-Side Effect* or *Pathway-Side Effect* relations.

It is essential to identify adverse events in the early phases of drug development. Two of these early phases include target discovery and animal models. Animal models have specific characteristics that mimic human diseases. The technologies for the creation of transgenic animals, where certain genes are deleted, modulated, or added, have progressed tremendously in the last decade. As a result, the predictive power of animal models for human disease and pharmacology is improving. It is crucial to note that some experts in the pharmaceutical industry and the U.S. Food and Drug Administration (FDA)

believe that inadequate animal models, or the lack of animal models altogether, are a major obstacle in drug discovery and development. Pharmaceutical companies have long used model organisms in preclinical efficacy [71]. The laboratory mouse is the premier animal model for understanding the genetic and molecular basis of human biology and disease [72]. Therefore, mice models can be a useful resource to understand potential side effect of the drugs. To date, there has been no networks based systematic study to understand *Mouse Phenotype -Human Side Effect* relationships.

In next section we will list the data sources that we used in this research and give a brief background for each. We will also introduce the network based methods that we employed for finding significant links in resulting multilevel networks created by integration of these databases. In the results and discussion section we will validate our method through receiver operating characteristics (ROC) curves and point out major findings. Finally we will give most significant findings and future extensions of this work in the last section

6.3 Methods

In this study we obtained *drug-target* relationships from Drugbank database [56]. This database provides detailed drug (i.e. chemical, pharmacological and pharmaceutical) data with comprehensive drug target (i.e. sequence, structure, and pathway) information. The database contains 6826 drug entries including 1431 FDA-approved small molecule drugs, 133 FDA-approved biotech (protein/peptide) drugs, 83 nutraceuticals and 5211 experimental drugs.

Drug-side effect relationships are obtained through SIDER database (Side Effect Resource) [53]. This data source is a collection of side effects for marketed drugs that are obtained through the drug package inserts[53]. It has 888 drugs and 1450 side effect terms associated with them.

A valuable resource for biological pathways and target association is the KEGG pathway database (Kyoto Encyclopedia of Genes and Genome) [54]. This is a collection of manually drawn pathway maps representing the collected knowledge on the molecular interaction and reaction networks [54]. In this data source there are 203 distinct pathways associated with hundreds of protein targets.

We combined SIDER databases with Drugbank to obtain *target-drug-side effect* relationships. To do that we matched SIDER drug names with Drugbank drug names and obtain 708 matching drugs out of 888 SIDER drugs. There are 653 distinct targets associated with 708 matching drugs. Furthermore, these 653 targets are matched in KEGG database to obtain pathway-target relationships. The integration of these databases forms a multipartite network that is shown in figure (6.1).

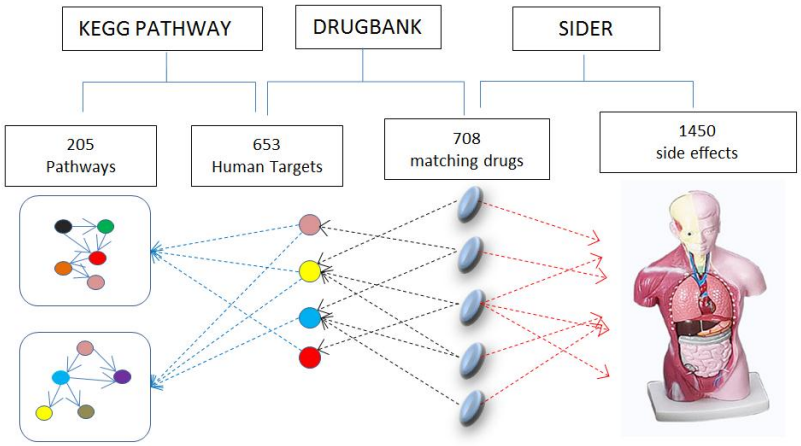


Figure 6.1. Integration of KEGG, Drugbank and SIDER databases on a multilevel network [53,54].

A valuable information resource that can be obtained from the MGI database is the *mouse phenotype-mouse gene* associations. The MGI database is a comprehensive information source that primarily provides genetic and genomic data to support laboratory mouse a model organism. [73] To achieve this goal, MGI maintains a comprehensive catalog of mouse genes and other genome features and associates these features with orthologous genes in other mammals, human diseases, functional annotation, mouse phenotype descriptions, DNA and protein sequence data and developmental gene expression information. We matched 429 mouse genes that are orthologues of 653 human targets. There are 3637 distinct mouse phenotypes corresponding to these 429 mouse genes. We combined MGI database with Drugbank and SIDER resources. In figure (6.2) all relationships from mouse phenotypes to human side effects are casted on a multipartite network frame.

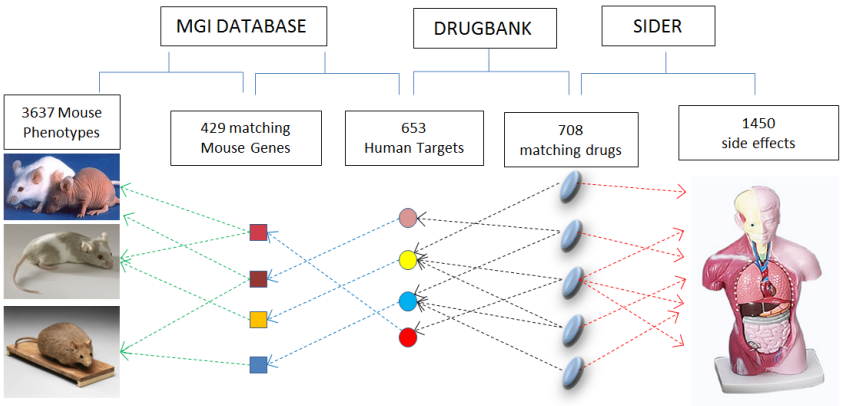


Figure 6.2 Integration of MGI, Drugbank and SIDER databases on a multilevel network.

Our aim is to find significant target-side effect, pathway-side effect and mouse phenotype-side effect relationships. In order to identify the significant links in a network,

the next logical step is to introduce a random model. By doing this observed structures in the network can be scored based on the random model and those with higher scores will give the significant links. We employed a p-value based approach to find important links. Each level in this multilevel network is randomized as follows; the degree of the nodes (number of connection at each node) in each level is kept constant and edges are shuffled for a large number of times. This procedure is repeated for each level. By integrating the random networks for each level we obtained a random multilevel network at each iteration. The resulting ensemble of networks constitutes random multilevel network space. Using this random network space the next step is to create a p-value for each observed *target-side effect*, *pathway-side effect* and *mouse phenotype-side effect* pairs. We based p-value on the count of drugs connecting these each pairs. One can obtain the distribution for the number of drugs connecting each pair of association from the random network space.

$$p_{ij}(k) = \frac{N_{ij}^{k}}{N_{total}} \qquad \textbf{(6.1)}$$

In equation (6.1) $p_{ij}(k)$ is the probability of observing k drugs connecting the i^{th} target (or mouse phenotype, or pathway) with the j^{th} side effect. N_{ij}^{k} refers to the number of random networks that have k drugs connecting the i^{th} target (or mouse phenotype, or pathway) with the j^{th} side effect. N_{total} is the total number of random networks in the ensemble. A p-value for each pair of association can therefore be calculated as the complementary cumulative distribution function.

$$\rho_{ij} = 1 - \sum_{k=0}^{k=k^*} p_{ij}(k) \qquad \textbf{(6.2)}$$

In this equation k^* is the number of observed drugs connecting the i^{th} target (or mouse phenotype, or pathway) with the j^{th} side effect. ρ_{ij} is the p-value for association between the i^{th} target (or mouse phenotype, or pathway) and the j^{th} side effect.

In the next step of our analysis we aim at validating our method by predicting each drug's side effects through their known pathway information. We utilized receiver the operating characteristics curve (ROC). We employed leave-one-out validation method. In this method each drug left out one at a time while creating the random network space. Significant pathway-adverse event association pairs are obtained using this random model. These pairs are then ranked in an increasing order of p-value. A p-value threshold is chosen and the pairs that have p-value lower than the threshold are chosen as significant pairs. From known targets of the drug that is left out one can find the related pathways. These pathways are matched in the significant pathway-adverse event pairs and the union of corresponding adverse events is the predicted adverse event set. Predicted adverse events are then compared with observed adverse events. Observed adverse events are the side effect list of the drug that is left out. Comparing predicted adverse events with observed ones gives false positive and false negative rates for the side effect prediction. By increasing the p-value threshold one can obtain ROC curves for each drug that is left out. This procedure is demonstrated in figure (6.3).

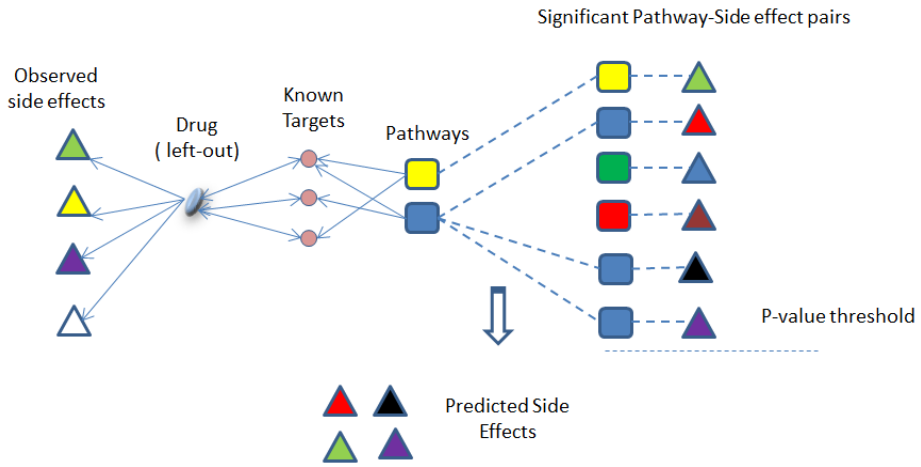


Figure 6.3 Procedure for predicting side effect for each drug that is left out.

A similar procedure can be applied for the prediction of adverse events through mouse phenotypes. Significant pathway-Side effect pairs are replaced with mouse phenotype and side effect pairs. Moreover from known targets of each drug that is left out one can find the corresponding mouse orthologous genes of these targets. From mouse genes relevant mouse phenotypes are extracted and considered as relevant mouse phenotypes to the drug that is left out. Next section will summarize our findings and validate prediction power of the method.

6.4 Results and Discussion

We have selected one example of significant link for each of the *pathway-side effect* and *mouse phenotype-side effect* networks. Calcium is a common signaling mechanism, as once it enters the cytoplasm of a cell it exerts regulatory effects on many enzymes and proteins. Calcium can act in signal transduction after influx resulting from activation of ion channels. It takes part in maintaining the balance of electrical system of the heart.

Heart block is an adverse event term that refers to dysfunction of the electrical system of the heart. It can cause syncope and palpitations. Calcium channel and Heart Block are found to be significantly related (giving a p-value<0.0001) in our framework. The targets and drugs connecting this pair are shown in figure (6.4).

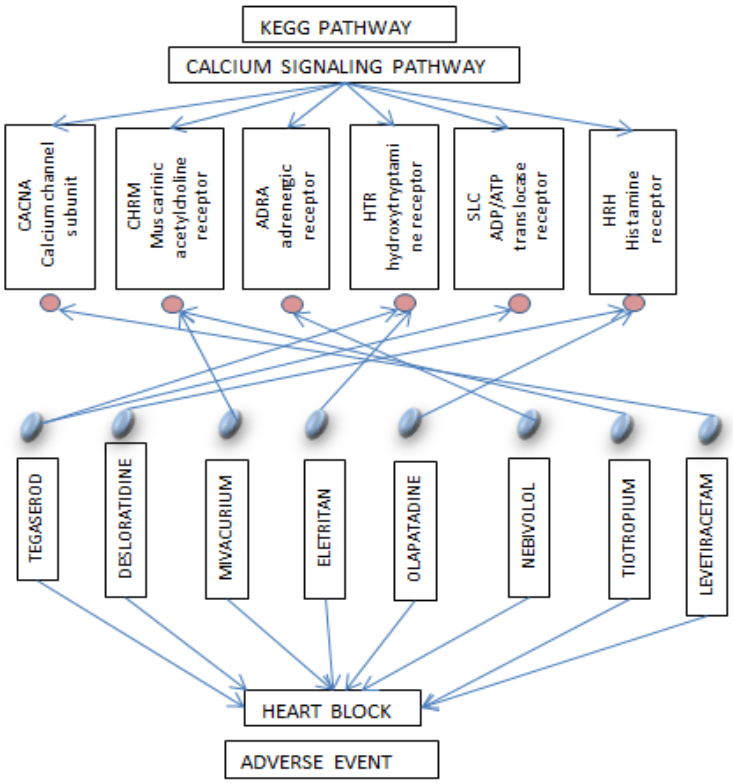


Figure 6.4 Association of Calcium signaling pathway with Heart Block through targets and drugs.

For mouse phenotype-adverse event association we found abnormal cardiovascular system physiology for mouse is significantly associated with congestive heart failure as human adverse event (figure (6.5)).

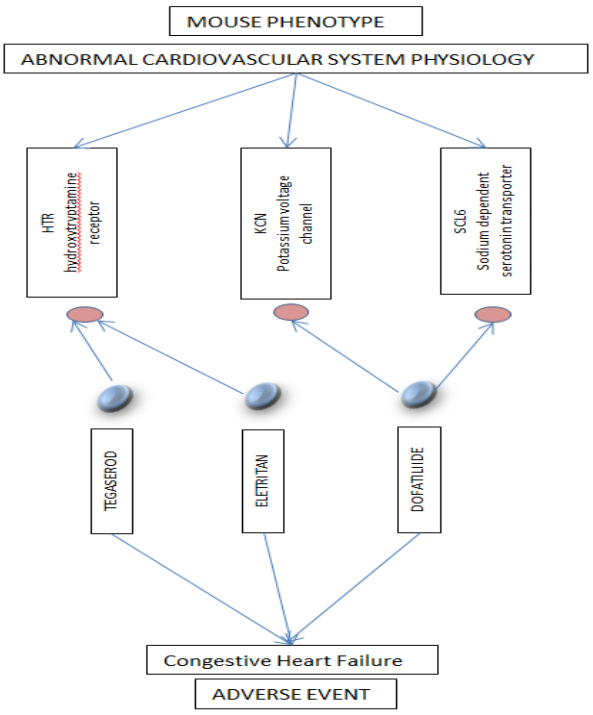


Figure 6.5 Association of abnormal cardiovascular system physiology for mouse with Congestive Heart failure adverse event through targets and drugs.

To validate our method we calculated ROC curve for 708 drugs when each left out at a time as it is outlined in section 6.3. Our aim is to predict side effects of a given drug using its pathway information. The ROC curve for each drug is averaged over 708 drugs. The resulting average ROC curve is shown in figure (6.6).

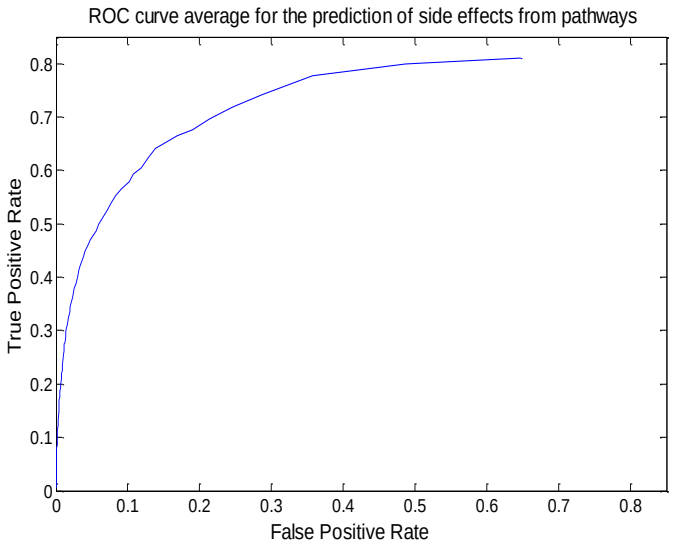


Figure 6.6 ROC curve average for the prediction of side effects from pathways.

Similarly we predicted side effects from mouse phenotypes that are relevant to the drug targets.

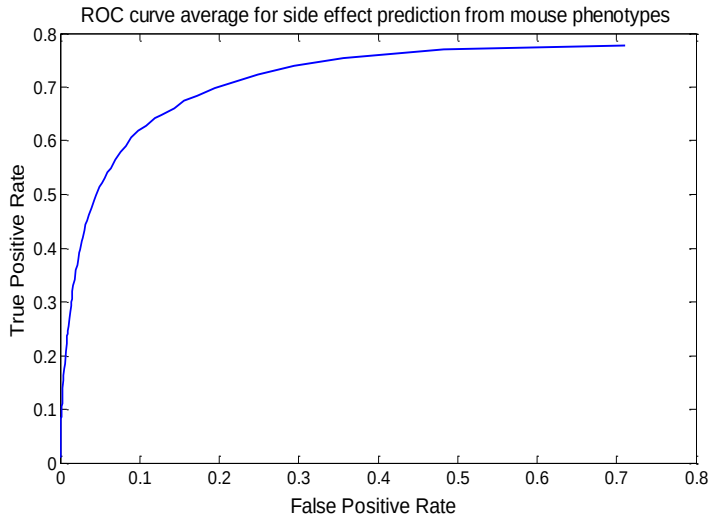


Figure 6.7 ROC curve average for the prediction of side effects from mouse phenotypes.

As it can be seen from figure (6.6) and figure (6.7) our method is capable of predicting side effects from pathways and mouse phenotypes. Furthermore, biologically relevant *pathway-side effect* and *phenotype-side effect* pairs are found to be significantly associated in our method. This study is capable of providing a framework for side effect prediction in the early phases of drug development using pathway and animal model information. It is very critical to design clinical trial to observe any potential side effect risks. At this junction our framework for estimating possible toxicity can be a useful tool. This framework can further be extended to include different databases, drug chemical structure information and genomic information on targets.

6.2 Conclusions

An adverse event (side effect) is an unwanted and harmful response to a drug that has happened during treatment of patients or clinical trials. Increasing regulatory research is focused on the obligation of the medical community, pharmaceutical industry and health authorities to ensure that marketed drugs have acceptable benefit-risk profiles. It is essential to identify adverse events in the early phases of drug development, therefore side effect prediction is critical for pharmaceutical research. Two of these early phases include target/pathway discovery and animal models. An adverse event can be caused by drugs known targets (target effect) or it can be due to proteins that are not yet identified as the targets of the drug (off target effects). Most adverse events are harmful to humans, but they can also be utilized to find new uses for known drugs. The knowledge of a pathway allows separate targeting of upstream or downstream targets. Inhibition or modulation of selected targets in the same pathway could lead to the same therapeutic

with fewer side effects or better druggability. Furthermore these targets can crosstalk with other pathways which may be potential sources of the observed side effects. Therefore the knowledge of pathways and their relation to each other helps researchers understand side effect profiles [71]. Animal models have specific characteristics that mimic human diseases.]. The laboratory mouse is the premier animal model for understanding the genetic and molecular basis of human biology and disease [72].

In this study we introduced a computational framework for side effect prediction from pathway and mouse phenotype information. We integrated MGI, KEGG pathway ,Drugbank and SIDER information on a multilevel network representation. A p-value based approach is introduced to find significant associations between *pathway-side effect* as well as *mouse phenotype-side effect* pairs. We demonstrated the biological relevance of these associations with two examples. Finally, we validated the prediction power of our method using ROC curves.

It is very critical to design clinical trial to observe any potential side effect risks. Our approach can provide a framework for side effect prediction in the early phases of drug development using pathway and animal model information. Our framework for estimating possible toxicity can be a useful tool. This work can further be extended to include different databases, drug chemical information and other genomic resources.

CHAPTER 7

CONCLUSIONS AND FUTURE EXTENSIONS

It is immensely important to understand biological networks in order to understand complex diseases, identify novel, safer protein targets for therapies and design efficient drugs. Computational approaches for identifying these networks become crucial and have been growing in parallel with the increasing amount of genomic data. 'Systems biology' has emerged as an interdisciplinary science that has as one of its foci revealing biological networks through genomic data.

The contribution of this thesis to Systems Biology can be stated in two ways; Predicting biological network topology and dynamics to understand complex machinery of biology and finding missing or significant links that have many important applications in getting a better picture of the network wiring of the biological systems.

In chapter 3 we addressed the problem of network identification from noisy measurements. It is known that biological data has significant levels of noise. In regression from dynamic data the resulting estimation model has noise term in both dependent and independent variable. Total Least Squares (TLS) is capable of taking error in independent variables into account. Constrained Total Least Squares (CTLS) is a further improvement on TLS that can incorporate the correlation in the noise.

We demonstrated the superior performance of our novel CTLS framework over other estimation methods on examples with a wide range of data points and noise levels. Though CTLS methods seem to improve parameter estimation significantly over the existing methods, the error levels are still high despite reasonable noise levels. Therefore,

it is necessary to use network connectivity data with a combination of optimal experimental design to obtain high accuracy parameter estimation.

In chapter 4 we demonstrated our approach for incorporating prior connectivity data with time series data. Binding data provides an initial topology of the gene networks. However, this data suffers from high rate of false positive and negative errors. We showed that our algorithm is able to fuse connectivity data and micro-array data to approach true network topology and dynamics provided that enough data is available. However, for reasonable levels of noise in the data (10-20% multiplicative noise) the error in parameters is still quite high. This highlights the importance of additional prior information. The major problem in topology data is the false negatives, in other words, missing links in the prior topology. It is important to be able to identify missing links in the prior information.

Possible extensions: In this method the likelihood term can be improved to include noise structure. As we demonstrated in Chapter 3, noise in resulting network models is correlated along the time domain. Furthermore noise can be multiplicative in nature. This information can be used to improve likelihood expression.

In chapter 5 we concentrated on link prediction techniques using network's topological distribution data to address the question of filling in possible false negative connections. We introduced a novel link prediction method that is based on local connectivity information. The contribution of this work can be summarized in two ways. First, our method is a novel and effective computational algorithm to quantify indirect relationships between the objects of biological research of interest by using existing relationships from text mining databases to automate the search for novel drug targets.

This method can also be used for different purposes such as; annotating diseases with similar aetiology, reposition existing drugs, or discovering adverse events for the targets. Second, in a case study involving 9575 proteins in the Ariadne database, our method outperformed the Jaccard method for the prediction of existing links for all proteins. This illustrates its prediction capability for biological networks.

Possible extensions: In this work the missing links between pairs of nodes are predicted through local connectivity information for the pair of the nodes based on a score that is derived from a probabilistic approach. This score function can be further extended to consider complementary cumulative distribution for the probability of observing shared common nodes with particular in-degree distribution. Complementary cumulative function will give one-sided p-value and score function can be based on this p-value.

In chapter 6 we integrated MGI, KEGG pathway, Drugbank and SIDER information on a multilevel network representation. A p-value based approach is introduced to find significant associations between *pathway-side effect* as well as *mouse phenotype-side effect* pairs. We demonstrated the biological relevance of these associations with two examples. Finally, we validated the prediction power of our method using ROC curves.

It is very critical to design clinical trial to observe any potential side effect risks. Our approach can provide a framework for side effect prediction in the early phases of drug development using pathway and animal model information. Our framework for estimating possible toxicity can be a useful tool.

Possible extensions: This work can further be extended to include different databases, drug chemical information and other genomic resources. AERS database for drug-side effect relationships can be used in combination with SIDER. AERS includes drug-side effect profiles that have exposure to a larger population compared to clinical trial data as in the case of SIDER. Therefore, it can give valuable information on rare side effects. Drug Chemical information can give a larger target space possibly including off-targets.

APPENDIX A

STOCHASTIC APPROACHES TO THE GENE NETWORK INFERENCE

A.1 Bayesian Networks

Bayesian network is a graphical model that represents the causal relationship in random variables. Suppose that we have *N* genes in a graph represented by an array of *N* random variables (expression levels), $X = \left(X_1, X_{2,\ldots,} X_N \right)$. Bayesian networks then enable us to compute the joint probability by the product of conditional probabilities [20].

$$P(X) = \prod_{j=1}^{N} P\left(X_j \mid Pa_j \right) \tag{A.1.1}$$

Where Pa_j is the set of random variables corresponding to the direct parents of X_j in a given graph, G. By the Bayes theorem, the posterior probability of the graph can be represented as

$$P(G \mid X) = \frac{P(G)P(X \mid G)}{P(X)} \tag{A.1.2}$$

where $P(G)$ is the prior probability of the graph, $P(X \mid G)$ is the likelihood of the data X. $P(X)$ is the normalizing constant [20].

A.2 Dynamic Bayesian Networks

Dynamic Bayesian networks represent the dependency in gene expression levels based on time-course data. Suppose that $X(t) = \left(X_1(t), X_2(t), ..., X_P(t) \right)$, each being a random variable, are the expression levels of genes at time point t. $(t = 1, ..., N)$ [20]. This can be

formulated as a bipartite graph with P nodes that allows edges from $X(t)$ to $X(t+1)$, where, $t = 1,...,N-1$. The directed graph, G_T of the causal relationship among P random variables is then constructed by estimating the bipartite graph. Under this topology, one can have the decomposition as follows;

$$\Pr\left(X(1),..X(t).,X(N)\right) = \prod_{t=1}^{N}\prod_{j=1}^{P}\Pr\left(X_j(t) \mid Pa_j(t-1)\right), \qquad \text{(A.2.1)}$$

Where $Pa_j(t)$ is the set of random variable at time t corresponding to the direct parent of $X_j(t)$ in bipartite graph, G_T [20]. In this equation, the distributions are assumed to have Markov property with independence along time points and genes.

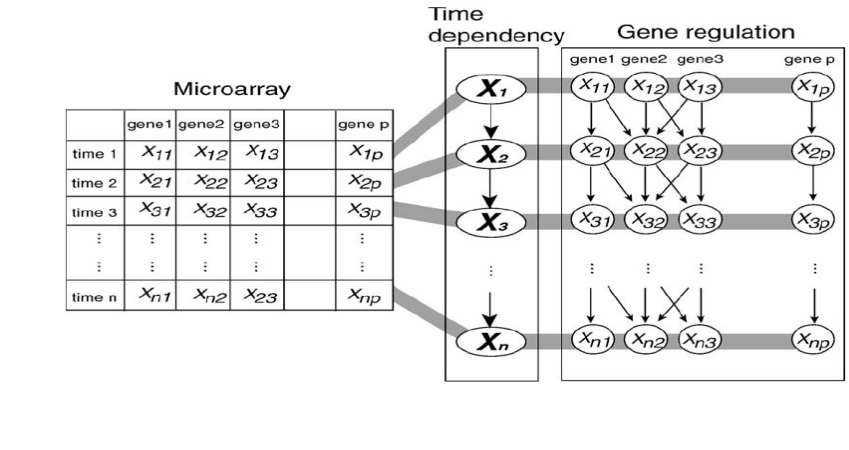


Figure 10. Graphical view of Dynamic Bayesian Network Model.

In Figure A.2.1, the structure of DBN in case of gene regulation is depicted. Micro-array data for different time points are shown in a table form. Each row of the table corresponds to a single micro-array experiment in a particular time point.

The aim of learning the topology of dynamic Bayesian network is to determine the topology, G that is most probable given data D (See Equation A.1.2). The notion of the most probable network is made formal by the Bayesian scoring metric (BSM), which is simply the log posterior probability of G given D:

$$BSM(G:D) = \log(G \mid D) = \log(D \mid G) + \log(G) + c \qquad \text{(A.2.2)}$$

This is simply derived by taking the logarithm of bayes rule employed in equation A.1.2 A common choice for the log prior over structures, $\log(G)$, is to assume that it uninformative.: every structure is equally likely: in this case, the prior term can be safely ignored since it is same for all structures.

APPENDIX B

DETERMINISTIC APPROACHES TO THE GENE NETWORK INFERENCE

B.1 Systems of Differential Equations

A set of ODEs, one for each gene, describes gene regulation as a function of other genes.

$$\dot{X}_i = f_i(X_1, X_2, ..., X_N, \theta_i, U_i) \tag{B.1.1}$$

where θ_i is a set of parameters describing the interactions among genes (the edges of the graph), $i = 1...N$, N is the number of genes and U_i is the amount of external perturbation applied to the gene.

To reverse-engineer a network using ODEs , a functional form for f_i is chosen and then the parameters θ_i are estimated from the gene expression data using some optimization or regression techniques.

Linear form

The linear discrete ODE models for gene network can be written as

$$\dot{X}_{il} = \sum_{j=1}^{N} A_{ij} X_{jl} + U_{il} \tag{B.1.2}$$

where $\dot{X}_{il}$ is the mRNA concentration of gene i following the perturbation experiment, l;

A_{ij} represent the influence of gene j on gene i; U_{il} is the external perturbation to the expression of gene i in experiment l [22]. One can assemble the expressions in a matrix form as follows;

$$\dot{X} = AX + U \tag{B.1.3}$$

B.2 S-Systems Approach

S-Sytems approach for gene network can be formulated as follows;

$$\frac{dX_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} X_j(t)^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j(t)^{h_{i,j}} \tag{B.2.1}$$

where N is the number of state variables (gene expression levels), X_i. The terms, $g_{i,j}$ and $h_{i,j}$ define the effect of X_j on X_i. The first term includes all effects that increase X_i, whereas second term includes the influences that decrease X_i. S refers to synergism and saturation. These are two fundamental properties of biological systems.

APPENDIX C

C.1 Artificial Gene Networks

Mendes et al [16] proposed a nonlinear differential equation system that generates random artificial networks according to well-defined kinetic properties.

In their model, a nonlinear form for f_i (right hand side of differential equations describing expression level of each gene) is assumed and it is decomposed into two components, namely; synthesis and degradation rates.

$$\dot{x}_i(t) = S_i(x_1, ..., x_N) - D_i(x_i) \tag{C.1.1}$$

where S_i stands for nonlinear synthesis term and D_i is the linear, first order degradation. x_i is the expression level of i^{th} gene. S_i term encompasses the nonlinear relation between transcription rate and inhibitor and activator expression levels. One can write this relation as;

$$S_i = v_i \prod_j \left(\frac{K_j^{n_j}}{x_j^{n_j} + K_j^{n_j}} \right) \times \prod_k \left(1 + \frac{x_k^{n_k}}{x_k^{n_k} + L_k^{n_k}} \right) \tag{C.1.2}$$

In this formulation, x_j stands for the inhibitor concentrations and j is the inhibitor index for the regulation of gene i. Similarly, x_k accounts for activator concentration and k is the activator index. The exponents n_j and n_k indicates the sigmoidicity of the curves. K_j and L_k are the constants, and v_i is the synthesis rate constant.

Along with first order degradation term, gene regulation model becomes;

$$\dot{x}_i(t) = v_i \prod_j \left(\frac{K_j^{n_j}}{x_j^{n_j} + K_j^{n_j}} \right) \times \prod_k \left(1 + \frac{x_k^{n_k}}{x_k^{n_k} + L_k^{n_k}} \right) - d_i x_i \ , \qquad \textbf{(C.1.3)}$$

where d_i is the degradation constant.

C.2 Simulation of artificial gene networks and obtaining data

Time-course data

Here we illustrate how we create, simulate and obtain data from the artificial gene network on a simple example. In figure 7, a simple network example is shown. In this graph, circles (nodes) indicate the genes and an arrow shows activation.

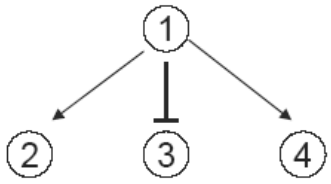


Figure 11: A simple 4-gene network. Gene 1 is the regulator for the rest. It is a single-input motif. Gene 1 is activating gene number 2 and 4, however it inhibits gene A number 3.

A connectivity matrix for the example network can be obtained as follows;

$$C = \begin{bmatrix} 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \textbf{(C.2.1)}$$

Where rows show the indices of the regulator and column numbers correspond to the indices of regulated genes. The system of nonlinear differential equations for this graph can be written down as follows [16];

$$\dot{x}_1(t) = v_1 - d_1 x_1$$

$$\dot{x}_2(t) = v_2\left(1 + \frac{x_1^{n_1}}{x_1^{n_1} + L_1^{n_1}}\right) - d_2 x_2 \qquad \textbf{(C.2.2)}$$

$$\dot{x}_3(t) = v_3\left(\frac{K_1^{n_1}}{x_1^{n_1} + K_1^{n_1}}\right) - d_3 x_3$$

$$\dot{x}_4(t) = v_4\left(1 + \frac{x_1^{n_1}}{x_1^{n_1} + L_1^{n_1}}\right) - d_4 x_4$$

Simulating this system with an appropriate numerical method, one can obtain time-course data in different time intervals. Different levels of Gaussian noise can be added to the data in order to mimic experimental and biological variations.

REFERENCES

[1] DRISCOLL, M. E. and GARDNER, T.S., "Identification and control of gene networks in living organisms via supervised and unsupervised learning," *Journal of Process Control,* vol. 16, pp. 303-311, 2006.

[2] GARDNER, T.S. and FAITH, J.J., "Reverse-engineering transcription control networks," *Physics of Life Reviews,* vol. 2, pp. 65-88, 2005.

[3] LOCKHART, D.J., DONG, H., BYRNE, M.C., FOLLETIE, M.T., GALLO, M. V., CHEE, M.S., MITMANN, M., WANG, C., KOBAYASHI, M., HORTON, H. and BROWN, E.L., "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nature Biotechnology,* vol. 14, pp. 1675-1680, 1996.

[4] BROWN, B.O. and BOTSTEIN, D., "Exploring the new world of the genome with DNA micro-arrays," *Nature Genetics*, vol. 21, pp. 33-37, 1999.

[5] FUENTE, A., BRAZHNIK, P. and MENDES, P., "Linking the genes: inferring quantitative gene networks from micro-array data," *TRENDS in Genetics,* vol. 18, no.8 pp. 395-398, 2002.

[6] SONTAG, E., KIYATKIN, A. and KHOLODENKO, B.N., "Inferring dynamics architecture of cellular networks using time series of gene expression, protein and metabolite data," *Bioinformatics,* vol. 20, no.12, pp. 1877-1886, 2004.

[7] GAO, F., FOAT, B.C. and BUSSEMAKER, H.J., "Defining transcriptional networks through integrative modeling of mRNA expression and transcription binding data," *BMC Bioinformatics,* vol. 5, no.31, 2004.

[8] SALGADO, H., SOCORRO, C., ANTONIO, M.A., PEREDO, E.D., SOLANO, F.S., GIL, M.P., ALONSO, D.G., JACINTO, V.J., ZAVALETA, A.S., MARTINEZ, C.B. and VIDES, J.C. "RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12," *Nucleic Acids Research,* vol.32, pp.72-74, 2004.

[9] LEE, T.I., RINALDI, N.J., ROBERT, F., ODOM, D.T., JOSEPH, Z., GERBER, K.G., HANNETT, N.M., HARBISON, C.T., THOMPSON, C.M., SIMON, I., ZEITLINGER, J., JENNINGS, E.G., MURRAY, H.L., GORDON, D.B., REN, B., WYRICK, J.J., TAGNE, J.B., VOLKERT, T.L., FRAENKEL, E., GIFFORD, D.K. and YOUNG, R.A. "Transcriptional Regulatory networks in Saccharomyces Cerevisae," *Science,* vol.298, no.5594, pp.799-804, 2002.

[10] ROGERS, S., KHANIN, R. and GIROLAMI, M., "Bayesian model-based inference of transcription factor activity," *BMC Bioinformatics,* vol. 8, no.2, 2007.

[11] TEGNER, J., YEUNG, M.K.S., HASTY, J. and COLLINS, J.J., "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *PNAS,* vol. 100, no.10, 2003.

[12] LOCKHART, D.J. and BROWN, P.O., "Genomics, gene experession and DNA arrays *Nature,* vol. 405, no.6788, pp. 827-836, 2000.

[13] DERISI, J.L., IYER, V.R. and WINZELER, E.A., "Genomics, gene experession and DNA arrays," *Science,* vol. 278, pp. 680-686, 1997.

[14] WEN, X., FUHRMAN, S., MICHAELS, G.S., CARR, D.B., SMITH, S., BARKER, J.L. and SOMOGYI, R., "Large-scale temporal gene expression mapping of central nervous system development," *PNAS,* vol. 95, no.1, pp. 334-339, 1998.

[15] YEUNG, M.K.S., TEGNER, J. and COLLINS, J.J., "Reverse engineering gene networks using singular value decomposition and robust regression," *PNAS,* vol. 99, no.9, 2002.

[16] MENDES, P., SHA, W. and YE, KEYING., "Artificial gene networks for objective comparison of analysis of algorithms," *Bioinformatics,* vol. 19, no.2, pp. 122-129, 2003.

[17] BERNARDO, D.D., GARDNER, T.S. and COLLINS, J.J., "Robust Identification of Large Genetic Networks," *Pacific Symposium on Biocomputing*, vol. 9, pp. 486-497, 2004.

[18] FRIEDMAN, N., LINIA, M., NACHMAN, I. and PE'ER, D., "Using Bayesian Networks to analyze expression data," *Journal of Computational Biology*, vol. 7, pp. 601-620, 2000.

[19] HARTEMINK, A., GIFFORD, D., JAAKOLA, S. and YOUNG, R., "Using Graphical models and genomic expression data to statistically validate models of genetic regulatory networks," *Pacific Symposium on Biocomputing,* 2001.

[20] IMOTO, S., TAMADA, Y., ARAKI, H., YASUDA, K., PRINT, C.G., JONES, S.D., SANDERS, D., SAVOIE, C.J., TASHIRO, K., KUHARA, S. and MIYANO, S., "Computational Strategy for discovering druggable gene networks from genome-wide RNA expression profiles," *Pacific Symposium on Biocomputing,,* 2006.

[21] BANSAL, M., GATTA, G.D. and BERNARDO, D., "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics,* vol.22, no. 7, pp. 815-822, 2006.

[22] GARDNER, T.S., BERNARDO, D.D., LORENZ, D. and COLLINS, J.J., "Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling,"

[23] BRAZHNIK, P., “Inferring gene networks from steady-state response to single gene perturbations,” *Journal of Theoretical Biology,* vol. 237, pp. 427-440, 2005.

[24] SAVAGEAU, M.A., “Biochemical System Analysis: a Study of Function and Design in Molecular biology,” *Addison-Wesley,* Reading, MA, 1976.

[25] ALVES, R. and SAVAGEAU, M.A., “Systematic properties of ensembles of metabolic networks: application of graphical and statistical methods to simple unbranched pathways,” *Bioinformatics,* vol.16, pp. 534-547, 2000.

[26] VOIT, E.O., “Computational Analysis of Biochemical Systems,” *Cambridge University Press*, 2000.

[27] VOIT, E.O. and RADIVOYEVITCH, T., “Biochemical Sytems analysis of genome-wide expression data,” *Bioinformatics,* vol.16, pp.1023-1037, 2000.

[28] KIKUCHI, S., TOMINAGO, D., ARITA, M., TAKAHASHI, K. and TOMITA, M., “Dynamic modeling of genetic networks using genetic algorithm and S-System,” *Bioinformatics,* vol.19, no.5, pp. 643-650, 2003.

[29] MAKI, Y., TOMINAGO, D., OKAMOTO, M., WATANABE, S. and EGUCHI, Y., "Development of a system for the inference of large scale genetic networks," *Pacific Symposium on biocomputing,* 6, pp. 446-458, 2001.

[30] AKUTSU, T., MIYANO, S. and KUHARA, S., "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," *Pacific symposium on biocomputing,* pp.17-28, 1999.

[31] LIANG, S., FUHRMAN, S. and SOMOGYI, R., "REVEAL, a general reverse engineering algorithm for inference of gene network architectures," *Pacific symposium on biocomputing,* 3, pp.18, 1998.

[32] LIAO, J.C., BOSCOLO, R., YANG, Y.L., TRAN, L.M., SABATTI, C., ROYCHOWDHURY, V.P., "Network component analysis: reconstruction of regulatory signals in biological systems," *PNAS,* vol.100, pp.15522-15527, 2003.

[33] LI, Z., SHAW, S.M., YEDWABNICK, M,J. and CHAN, C., "Using a state-space model with hidden variables to infer transcription factor activities," *Bioinformatics,* vol.22, no.6. pp.747-754, 2006.

[34] BRYNILDSEN, M.P., TRAN, L.M. and LIAO, J.C., "A Gibbs sampler for the identification of gene expression and network connectivity consistency," *Bioinformatics,* vol.22, no.24. pp.3040-3046, 2006.

[35] BOULESTEIX, A.L. and STRIMMER, K., "Predicting transcription factor activities from combined analysis of microarray and ChIP data: a partial least squares approach," *Theoretical Biology and Medical Modeling*, vol.2, no.23, 2005.

[36] SANGUINETTI, G., LAWRENCE, N.D., and RATTRAY, M., "Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities," *Bioinformatics*, vol.22, no.22, pp.2775-2781, 2006.

[37] SABATTI, C. and JAMES, G.M., "Bayesian sparse hidden components analysis for transcription regulation networks," *Bioinformatics*, Vol.22, no.6, pp.739-246, 2006.

[38] SUN, N., CARROLL, R.J. and ZHAO, H., "Bayesian error analysis model for reconstructing transcriptional regulatory networks," *PNAS*, vol.103, no.21, pp.7988-7993, 2006.

[39] BERNARD, A. and HARTEMINK, A.J., "Informative structure priors: Joint learning of Dynamic Regulatory networks from Multiple Types of Data," *Pacific symposium on biocomputing*, 10, pp.459-470, 2005.

[40] ALTER, O. and GOLUB, G.H., "Integrative analysis of genome-scale data by using pseudo-inverse projection predicts novel correlation between DNA replication and RNA transcription," *PNAS*, 101, pp.16577-16582, 2004.

[41] ERNST, J., VAINASS, O., HARBISON, C. T., SIMON, I., BAR-JOSEPH Z., *"*Reconstructing dynamic Regulatory maps*," Molecular Systems Biology*, vol.3. no.74 pp.1-13, 2007.

[42] POURNARA, I. and WERNISCH, L., *"*Reconstruction of gene networks using Bayesian Learning and manipulation experiments*," Bioinformatics*, vol.20. no.17 pp.2934-2942, 2004.

[43] WILDENHEIN, J. and CRAMPIN, E.J., *"*Reconstructing gene regulatory networks: from random to scale-free connectivity,*" IEE Proceedings-Systems Biology*, vol.153, no.4, pp.247-256, 2006.

[44] GASCH, A.P., SPELLMAN, P.T., KAO, C.M., CARMEL-HARREL, O., EISEN, M.B., STORZ., G., BOTSTEIN, D. and BROWN., P.O. *"*Genomic expression programs in the response of yeast cells to environmental changes*" Molecular Biology of the Cell*, vol.11, pp.4241-4257, 2000.

[45] GEIER, F., TIMMER, J. and FLECK, C. "Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge" *BMC Systems Biology*, vol.1, no.11, 2007.

[46] STEINKE, F., SEEGER, M. and TSUDA, C. “Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models” *BMC Systems Biology*, vol.1, no.51, 2007.

[47] HARBISON,C.T., GORDON,B.D., LEE,T.I., RINALDI,N.J., MACISAAC,K.D.,DANFORD,T.W., HANNETT,N.M., TAGNE,J.B., REYNOLDS,D.B., YOO,J., JENNINGS,E.G., ZEITLINGER,J., POKHOLOK,D.K., KELLIS,M., ROLFE,A.P., TAKUSAGAWA,K.T., LANDER,E.S., GIFFORD,D.K., FRAENKEL,E. AND YOUNG,R.A. “Transcriptional regulatory code of a eukaryotic genome.” *Nature*, vol.431 (2004), pp. 99–104, 2004

[48] LI ,S., WU,L., AND ZHANG, Z. “Constructing biological networks through combine literature mining and microarray analysis: a LMMA approach” *Bioinformatics*, vol.22, no.17, 2006.

[49] LU, L. AND ZHOU,T. “Link prediction in weighted networks: The role of weak ties” *Europhysics Letters*, vol.89, no.1, 2010.

[50] LIEBEN-NOWELL, D. AND KLEINBERG, J. “Link prediction Problem for Social Networks” *Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management*, 2003.

[51] ZOLTAN, S., KOVACS, I.A., AND CSERMELY, P. "Drug-therapy networks and prediction of novel targets" *Journal of Biology*, vol.7, no.6, 2008.

[52] YAMANISHI, Y., ARAKI, M., GUTTERIDGE, A., HONDA, W. AND KANEHISA, M. "Prediction of drug-target interaction networks from the integration of chemical and genomic spaces" *Bioinformatics*, vol.24, pp. i232–i240, 2008.

[53] CAMPILLOS, M., KUHN, M., GAVIN, A., JENSEN, L.J., AND BORK, P. "Drug-target identification Using Side-Effect Similarity" *Science*, vol.321, 2008.

[54] KANEHISA, M., KUHN, M., GAVIN, A., JENSEN, L.J., AND BORK, P. "From Genomics to chemical genomics: new developments in kegg." *Nucleic Acids Research*, vol.34, 2006.

[55] SCHOMBURG, I, KUHN, M., GAVIN, A., JENSEN, L.J., AND BORK, P. "Brenda the enzyme database: updates and major new developments" *Nucleic Acids Research*, vol.32, 2004.

[56] WISHART, D.S., M., GAVIN, A., JENSEN, L.J., AND BORK, P. "Drugbank: a knowledge base for drugs, drug actions and targets" *Nucleic Acids Research*, vol.36, 2008.

[57] GETOOR, L AND DIEHL, C.P. *Explor. Newsletter*, vol.7, no.2, 2005.

[58] HUANG, Z., LI, X. AND CHEN, H. "Link prediction approach to collaborative filtering" *Proceeding of the 5th ACM/IEEE-CS Joint conference on Digital Libraries,* 2005.

[59] CLAUSET, A., MOORE, C. AND NEWMAN, M.E.J *Nature,* vol.453, 2008.

[60] CLAUSET, A., MOORE, C. AND NEWMAN, M.E.J. *Nature,* vol.453, 2008.

[61] WREN, J.D., BEKEREDJIAN, R., STEWART, J.A., SHOHET, R.V AND GARNER, H.R. "Knowledge Discovery by Automated Identification of Implicit relationsips" *Bioinformatics,* vol.20, no.3, 2004.

[62] NOVICHKOVA, S., EGOROV, S. AND DARASELIA,N. "MedScan a natural Language Processing engine for Medline abstracts" *Bioinformatics,* vol.19, no.3, 2003.

[63] HUFFEL, S. V. (1991). *The total least squares problem: computational aspects and analysis,* Society for Industrial and Applied Mathematics, Philadelphia.

[64] KIM, J., BATES, D. G., POSTLETHWAITE, I., HARRISON, P., AND CHO, K. (2007).' A least squares approach for identification of biochemical regulatory networks from noisy measurements' *BMC Bionformatics,* vol.8, no.8.

[65] JI, R., LIU, AND ZHANG, W. (2010).' The Application of Hidden Markov Model in building genetic regulatory networks ' *Journal of Biomedical Science and Engineering*, vol.3, pp633-637

[66] IDEKER, T., THORSSON, V., SIEGEL, A.F. AND HOOD, L. (2000).' Test for differentially-expressed genes by maximum likelihood analysis of microarray data ' *Journal of Computational Biology*, vol.7, no.6, pp805-817

[67] MORGAN, D. O. (2007).' The Cell Cycle: Principles of Control ' *Sinauer Publishing Associates.*

[68] IMAGE COURTESY OF AFFYMETRIX www.affymetrix.com.

[69] KNUDSEN, S. (2004).' Guide to Analysis of DNA microarray data ' *Wiley-Liss.*

[70] PEARL, J. (1998).' Probabilistic Reasoning in Intelligent Systems ' *Morgan Kaufmann Publishers Inc.*

[71] ECKSTEIN, J. (2011).' Alzheimer Research Forum: ISOA/ARF Drug Development Forum ' *Institute for the Study of Aging.*

[72] ROSENTHAL, N. and BROWN, S (2007).' The mouse ascending:perspectives of human-disease models ' *Nature Cell Biotechnology*, vol.9, pp993-999.

[73] BULT, C.J. et al. (2010).' The mouse Genome Database: enhancements and updates' *Nucleic Acids Research (Database Issue)*, vol.38, pp586-592.

[74] HAUBEN, M., MADIGAN,D., GERRITS, C.M., WALSH, L. and PUIJENBROEK, E.P.V. (2005). 'The role of data mining in pharmacovigilance' *Expert Opinion in Drug Safety*, vol.4, no.5, pp929-948.

[75] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, G. (2009). 'The elements of Statistical Learning' *Spring Series in Statistcs.*

[76] GAMERMAN, D. and LOPES, H.F. (2006). 'Markov Chain Monte Carlo Stochastic Simulation for Bayesian Inference' *Texts in Statistical Science.*