Modeling and Learning Realistic Genetic Interactions Using Dynamic Bayesian Network and Information Theory

Nizamul Morshed

A dissertation submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Supervisor

Associate Professor Madhu Chetty

Associate Supervisors

Associate Professor Manzur Murshed

Professor Terry Caelli

Gippsland School of Information Technology

Monash University

December, 2013

Typeset in Palatino by TEX and LATEX 2ε

Modeling and Learning Realistic Genetic Interactions Using Dynamic Bayesian Network and Information Theory

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Nizamul Morshed
December 11, 2013

Dedicated to my parents and family for their endless love and inspiration

Acknowledgments

First of all, I would like to thank Allah, the most gracious, for blessing me with the intellect and patience needed to carry out this research. My biggest thanks go to my parents, Md. Zahir and Nilufa Zahir, for their love and unconditional support throughout all these years. Loving thanks also to my sisters, Zohora Fathema and Asma Naznin, for their inspiring discussions and support when I needed them.

I gratefully acknowledge the encouragement and direction of my supervisors, A/Prof Madhu Chetty, A/Prof Manzur Murshed and Professor Terry Caelli. A work like this would have never been possible without the encouragement, help and constant guidance given by my supervisors.

Permission is given by authors/publishers of published articles to reproduce relevant figures and illustrations from their work in this thesis. I am grateful to them for their permission and support.

Thanks to all the Bangladeshi people here in Churchill, for making sure that I have a cultural and social life besides my Ph.D. I would also like to thank my friend Md Anit Khan for his help and cooperation. I would also acknowledge Ms Gillian Fulcher for proofreading the thesis. I wish to express my gratitude to all the staff and postgraduate students at Gippsland School of IT, Monash University, for their cordial support and inspirations throughout my candidature.

I would like to acknowledge scholarship support from Monash University, Gippsland School of IT and National ICT Australia (NICTA). Last but not the least, I wish to thank Monash University for providing me with the excellent opportunity to conduct my research in such a peaceful and friendly environment.

Nizamul Morshed

Monash University

December 2013

Abstract

Deciphering genetic interactions is of fundamental importance in computational systems biology, with wide applications in a number of other associated areas. Realistic modeling of these interactions poses novel challenges while dealing with the problem. Further, learning these interactions using computational methods becomes increasingly complex with the adoption of advanced and more realistic modeling techniques. In this thesis, we propose methods to address this challenge using a graphical model having sound probabilistic underpinnings, commonly known as dynamic Bayesian networks.

Inference of genetic interactions is usually carried out using DNA microarray data. This data provides snapshots of mRNA expression levels of a large number of genes from a single experiment. However, the number of samples from such experiments is small, and additionally, they contain missing values and noise. Bayesian networks are considered as one of the most promising ways by which these issues can be tackled. However, traditional Bayesian networks have their own limitations; for example, they neither take time information into account nor can they capture feedback. Further, accurate determination of the direction of regulation requires a significant number of tests to be performed. Dynamic Bayesian networks (DBN) are extensions of Bayesian networks that can effectively address these limitations.

In this thesis, we develop novel techniques for gene regulatory network reconstruction using DBN based modeling approach. We start with a basic DBN based model, and improve it so that it can represent and model both instantaneous and time-delayed genetic interactions. Initially, we aim to detect the occurrence of instantaneous and single-step time-delayed interactions, and subsequently this approach is further extended to model the instantaneous and multi-step time-delayed interactions. This approach of

modeling both instantaneous and multi-step time-delayed genetic interactions is superior to traditional DBN based GRN reconstruction techniques, where only the time delayed interactions are learnt.

In addition to modeling interactions, one needs a learning mechanism for inferring genetic interactions. To facilitate detection of nonlinear gene to gene interactions (in addition to linear interactions), which are prevalent in all genetic networks, we propose using well known properties, including fundamental results related to information theoretic measures for testing conditional independence relations in a DBN. This enables us to formulate efficient learning techniques for reconstructing GRNs. Using these theoretical underpinnings, we first implement simple hill-climbing techniques that enable detection of various types of interactions among genes. Subsequently, we use these results to devise novel score and search based evolutionary computation techniques, which can effectively explore a significantly larger search space.

We carry out investigations using both synthetic networks as well as real-life networks. For real-life network study, we use four different microarray data sources, covering three organisms, namely, yeast, *E. coli* and cyanobacteria. We use networks of varying sizes, ranging from five-gene small networks (yeast) to large scale networks of cyanobacteria (730 genes). The evaluation of the performance is carried out using four widely used performance measures. For some networks where we do not have sufficient information for calculating these performance measures, we use literature mining for performing comparative evaluations of the proposed approaches. For the large scale network of cyanobacteria, we use gene ontology (GO) based analysis of gene functionalities, in addition to degree distribution analysis of the inferred network.

Due to the inherent difficulties associated with inferring GRNs using DNA microarray data, it is often supplemented by other sources of data; for example, genomic data and protein-protein interaction data. In this thesis, we propose a framework that jointly learns the structure of a GRN and a protein-protein interaction network (PPIN). Using this process, the GRN reconstruction technique can effectively make use of the vast wealth of knowledge available from these external sources of data. This knowledge is fed to the GRN reconstruction process probabilistically, thereby enabling it to weigh each different data source according to the reliability of that source. The approach is applied on yeast networks where four different interaction data sources and a number of genomic data sources are used. Together with the novel modeling and learning techniques proposed in this thesis, the probabilistic integration of different types of knowledge sources and the

co-learning of GRN with PPIN represents a significant step towards the reconstruction of GRNs using DBNs.

Contents

List of Tables

List of Figures

Publications

Publications arising from this research include:

- **Peer reviewed journal articles**:
  - N. Morshed, M. Chetty, and V.X. Nguyen. *Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique*. **BMC Systems Biology**, *6*(1):62, 2012. [**Impact Factor: 3.15**].

- **Peer reviewed conference articles**:
  - N. Morshed, M. Chetty, N. Vinh, and Terry Caelli. *mDBN: Motif Based Learning of Gene Regulatory Networks Using Dynamic Bayesian Networks*. In **Genetic and Evolutionary Computation Conference (GECCO'2013)**, Accepted. [**ERA Rank A**].
  - N. Morshed, M. Chetty, and N. Vinh. *FusGP: Bayesian Co-learning of Gene Regulatory Networks and Protein Interaction Networks*. In **Neural Information Processing (ICONIP'2012)**, pages 369–377. Springer, 2012. [**ERA Rank A**].
  - Nizamul Morshed, Madhu Chetty, and Nguyen Xuan Vinh. *Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique*. In **Proceedings of the 18th international conference on Neural Information Processing - Volume Part II, ICONIP'11**, pages 248–257, Berlin, Heidelberg, 2011. Springer-Verlag. [**ERA Rank A**].
  - N. Morshed and M. Chetty. *Reconstructing genetic networks with concurrent representation of instantaneous and time-delayed interactions*. In **2011 IEEE Congress on Evolutionary Computation (CEC)**, pages 1840–1847. IEEE, 2011. [**ERA Rank A**].
  - N. Morshed and M. Chetty. *Combining instantaneous and time-delayed interactions between genes-a two phase algorithm based on information theory*. **AI'2011: Australasian Joint Conference on Artificial Intelligence**, pages 102–111, 2011. [**ERA Rank B**].
  - Nizamul Morshed and Madhu Chetty. *Information Theoretic Dynamic Bayesian Network Approach for reconstructing genetic networks*. **AIA'2011: International Conference on Artificial Intelligence and Applications**, pages 236–243, 2011. [**ERA Rank B**].

Notations

Symbol	Description
D	Microarray dataset
X	Gene (random variable)
$\boldsymbol{X}$	Set of genes
x	Value taken by a gene
N	Number of samples in aami dataset
n	Number of genes
$\boldsymbol{V}$	Set of all genes in the GRN
$\boldsymbol{E}$	Set of edges in the GRN
Ω	Sample space for probability calculation
$H(X)$	Entropy of X
$H(X,Y)$	Joint entropy of X and Y
$H(X\|Y)$	Conditional entropy of X given Y
$P(\cdot)$	Probability of an event
$Pa(X)$	Parent set of gene X
$MI(X,Y)$	Mutual information between X and Y
$MI(X,Y\|Z)$	Conditional mutual information between X and Y given Z
$D_{KL}(P\|\|Q)$	Kullback-Leibler divergence between two distributions P and Q
$r_{x.y}$	Correlation between variables X and Y
$r_{xy.z}$	Partial correlation between variables X and Y given variable Z
$(X \perp Y\|Z)$	X and Y are conditionally independent given Z
$\boldsymbol{Markov}(G)$	The set of local Markov assumptions about a graph G
$\boldsymbol{\theta}$	Parameter set storing the conditional joint probability distribution
$MB(X)$	Markov Blanket of X

Continued on next page...

Notations – Continued

Symbol	Description
$\Gamma(\cdot)$	Gamma function
r_i	Number of states of the variable X_i
q_i	Number of possible configurations of the parent set of X_i
w_{ij}	Configuration of $Pa(X_i)$
N_{ijk}	Number of instances in D where the $X_i = x_{ik}$ and $Pa(X_i) = w_{ij}$
N_{ij}	Number of instances where $Pa(X_i)$ take their j-th configuration w_{ij}
N_{ik}	Number of instances where X_i takes its k-th value x_{ik}
DI_{XY}	Directionality Index between genes X and Y
$Z(\theta)$	Normalizing constant for binary Markov networks
ω	Search space for evolutionary algorithms
D_p	Protein-protein interaction dataset
D_r	Dataset for gene regulatory network
G_p	Protein-protein interaction network
$L(i, j)$	Likelihood ratio

Chapter 1

Introduction

The diverse functions of any living cell are carried out through the concerted activity of many genes and gene products, coordinated through molecular networks involving interacting proteins, RNA, and DNA molecules. With the advent of high-throughput and functional genomics, a system level view of the complex biological systems responsible for the successful functioning of all living organisms has now become a possibility. The understanding of these complex systems is of tremendous importance in developing new methods for various applications, such as treating complex diseases, creating environment-friendly green fuels, and designing new drugs [16, 65]. Inevitably, this discipline, known as systems biology, has therefore become a recent trend in molecular cell biology research [38, 75, 83, 137, 205, 239].

In the cell of any living organism, there are thousands of genes interacting with each other, at any given time, to accomplish complicated biological tasks. Gene regulatory networks represent gene-gene regulatory relations in a genome and are models that display causal relationships between gene activities [29]. A regulatory network can be viewed as a cellular input-output device, and typically contains the following components at minimum (see Figure 1.1) [165, 201]:

1. An input signal reception and transduction system (component 1 in the figure) that mediates intra and extracellular cues. Usually, more than one signal affects a given target gene.

2. A core "GRN component" complex (component 2 in the figure) composed of trans-acting regulatory proteins and cognate cis-acting DNA sequences. Functionally similar components may be associated with more than one target gene, which results in similar gene-expression patterns.

3. Primary molecular outputs (RNA and protein) from target genes, as shown in component 3 in the figure. The net effects are changes in cell phenotype and function (component 4 in the figure). Direct and indirect feedbacks also typically exist.

   More realistic networks often contain multiple levels of regulation, with the first-level gene products regulating expression of another group of genes, and so on. Beyond GRN boundaries are signalling responses and feedbacks. These do not involve regulation of gene expression, rather act directly on proteins and protein machine assemblies (indicated by dashed arrows). It may be noted here that there can be feedback loops from primary outputs (RNAs) directly onto the mechanisms of gene expression themselves, and in these special cases, there can be arrows from component 3 to component 2 directly.

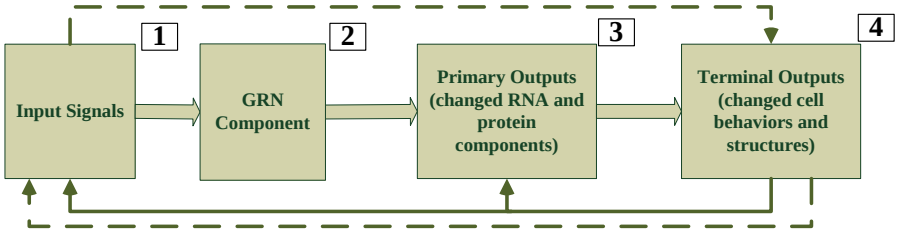

Figure 1.1: Schematic of a gene regulatory network

Over the past decades, molecular biology has been extremely successful at identifying and cataloging the functional components of cells, and also genes, RNA transcripts, proteins, metabolites etc. Despite extensive knowledge of individual components, we are far from understanding how cells work, and how their functioning could be easily manipulated for the betterment of mankind [189]. As has been said, gaining an understanding of how the genetic interactions occur and how they form networks of regulations is of tremendous importance to understanding the overall mechanisms of living cells. Ubiquitously, the understanding is facilitated by developments in the DNA microarray technology, which enables us to measure expression levels for a large number of genes at the

same time. These expression levels, under certain conditions, are basically a snapshot of the expression levels of genes at a given instant of time. With a large enough set of snapshots, it should theoretically be possible to uncover the underlying gene regulatory network (GRN). Researchers have applied microarray technology to study diseases such as Huntington's disease [249], HIV [186], and cancer [240]. However, numerous factors within the biological networks, e.g., various levels of time delays, stochastic behavior and feedback loops, complicate their modeling and inference mechanism. The modeling process is also plagued by the well known dimensionality problem of microarrays, namely, large number of genes and very few samples available from microarrays. Moreover, the gene-expression measurements are noisy, due to variations among different individuals, low quantities of some RNAs and measurement errors. As a result, the solution to this problem cannot depend solely on approaches from experimental biology; rather, it calls for a synergy of multiple disciplines that includes biology, computer science, and so forth. This thesis is devoted to addressing some of these challenges by developing sophisticated statistical machine learning approaches using dynamic Bayesian networks for reconstructing gene regulatory networks.

1.1 Motivation

Early approaches related to understanding genetic interactions essentially worked on a gene by gene basis. They consisted of gathering information about a single gene (or a single chemical reaction related to the genes under study) via biological experiments, and analysis of the obtained information afterwards. Although limited successes were achieved by such lab-based approaches, they were time and resource consuming, and hence could not be scaled for large sized genomes. Moreover, prediction of unobserved properties could not be done using these techniques. With the development of DNA microarray technology and other high-throughput experiments, understanding of genetic regulations has entered the next stage, from being simply a molecular biology approach to a data driven, computational approach.

Computational approaches to learning genetic regulations can be model based (e.g., BN based methods [76,176], differential equation based methods [17,78]); and also there can be approaches having no specific model (relevance network based methods [139,261],

clustering based methods [71,178]). Early approaches used relatively simple techniques, like clustering and boolean network based methods. Although simple techniques like clustering can work on large scale genomes, they work on a coarse level, and thus it becomes difficult to obtain deep levels of understanding about a particular organism under study. Techniques having medium complexity (compared to simpler clustering based methods), such as boolean network based methods and linear differential equation based methods, have also been used for reconstructing gene networks. However, they were also limited in scope due to various deficiencies. For example, boolean model based methods [132, 207] assume that genetic interactions can be described by boolean logic, which is hardly practical. Linear differential equation based methods [226, 253] assume linearity of interactions. This is also not true for genetic regulations, in general. Hence, recent approaches to understanding genetic interactions usually de advanced techniques such as BN (Bayesian network) based methods and nonlinear ODE (ordinary differential equation) based methods. Artificial neural network based methods [112, 137] have also been used, but they usually require lots of data samples for accurate reconstruction, and because microarray data samples are small in length, they generally do not perform well for GRN reconstruction.

Advanced computational techniques for reconstructing GRNs, e.g., nonlinear ODE based methods [115, 162], have the benefit that they can model detailed quantity information changing over a period of time, but these suffer due to the need of learning large number of model parameters. For large scale networks, the number of parameters that these methods need to estimate becomes significantly high, and due to the low number of samples and limited computational resources, correctly estimating these models becomes increasingly difficult. Also, due to these constraints, often the results produced by these methods bear no qualitative difference from linear approximation based techniques [176]. Following the seminal work by Friedman *et al.* [76], there has been great interest in inferring genetic interactions using Bayesian network based methods. However, using basic BN based models for reconstructing GRNs also has its own limitations:

1. BN based methods are static, and they do not take the dynamics information present in time series microarray data into account.

2. Techniques based on BN based models cannot model feedback loops.

3. When BN based methods use linear dependency measures like PCC (Partial Correlation Coefficient), they fail to discover nonlinear genetic interactions.

4. Most BN based methods restrict studies to smaller scale networks. However, biological networks being large scale, there is a need to assess the performance of algorithms on large scale networks.

5. Because microarray data is limited, it would be advantageous to integrate various prior knowledge available from different sources. Currently there is a dearth of techniques that can integrate multiple sources of prior knowledge.

6. They fail to effectively search the large and multi modal search space.

The first two problems are specific to static BN based methods, whereas the other problems apply to most GRN reconstruction techniques in general. The BN specific problems can be solved using its extension called dynamic Bayesian networks (DBN). Solving the rest of the problems listed above requires a careful consideration of the steps involved in reconstructing GRNs. Computationally, reconstructing a GRN from data consists of two steps: (i) a modeling technique[1], and (ii) a learning technique. In this thesis, we focus on both these steps. We use the improved DBN model rather than using BN as the modeling technique, and also address other limitations listed above to devise a reconstruction technique that improves the current state of GRN inference.

This proposed research is highly significant for at least three reasons. Firstly, it represents a significant step towards a systems-level understanding of the eukaryots such as the yeast cell cycle. When we are able to infer system dynamics, protocols, and the design principles of the eukaryotes, we will be able to extend to other larger datasets such as the cyanobacteria dataset. Secondly, it provides novel methods that will underpin the modeling of systems-level biological processes and their underlying mechanisms. Thirdly, the statistically supported gene network model established in this study will seed similar studies for other organisms and could expedite the discovery of new strategies for a wide variety of diseases and production of biofuels. Very recently, genes that are responsible for nerve related problems such as migraine have been discovered [30]. If we can deduce the regulatory relationships between these genes, we can cure those diseases by understanding the interactions amongst different genes. Moreover, application of the methods

[1]This excludes relevance network based methods, where there is no model to choose.

to real-life biological data like cyanobacteria will help us to stimulate carbon-dioxide sequestration and biofuel production capability of these organisms and this should have profound impacts on both the energy crisis and natural stability.

1.2 Aims and Objectives

The specific aims and objectives that we set out for this thesis are as follows:

1. To develop novel techniques that harness the power of DBN modeling, and use MI/CMI based detection of nonlinear regulatory interactions in a manner such that inferred interactions using the technique are statistically significant. Quantitative assessment of the method using real-life biological datasets.

2. To the best of our knowledge, for BN/DBN based GRN reconstruction, either instantaneous or time-delayed interactions (but not both) based modeling have been used ubiquitously. Our second objective is to devise a *modeling framework* to enhance the basic modeling techniques used by earlier approaches so that it can handle both instantaneous and time-delayed interactions among genes. Exploring different avenues by which this model can be used for GRN reconstruction.

3. To devise novel techniques for *joint learning* of multiple-step time-delayed and instantaneous interactions to realize the biological fact that multiple regulators can regulate genes simultaneously, with varying time delays.

4. To develop stochastic search methods using local heuristics for optimizing the network search so that it can explore the huge search space efficiently.

5. To enhance the GRN model to include the influence of neighborhood proteins by fusing the knowledge from protein-protein interaction data into the process of GRN reconstruction.

6. To study large scale gene regulatory networks. Although gene regulatory networks are significantly large, in practice, GRN reconstruction algorithms are usually assessed using smaller sub-networks. While these sub-networks aid in proving the concepts behind the reconstruction technique, in real-life scenarios, larger networks are of greater importance. One objective of this thesis is the study and mining of

meaningful biological insights from large scale biological networks of important organisms, e.g., cyanobacteria.

7. Although quantitative analysis techniques are straightforward for GRNs, due to scarcity of exact knowledge about the networks, it sometimes becomes difficult to assess the performance of reconstruction techniques quantitatively. Hence, one of the objectives of this research is to compile knowledge sources which can aid in the assessment of large scale GRNs, and use this knowledge for assessing reconstruction algorithms.

1.3 Contributions

In order to achieve the aims and objective outlined in the previous section, the research work reported in this thesis makes the following contributions:

1. A novel DBN based GRN reconstruction technique, that uses information theoretic quantity based conditional independence tests to infer regulatory interactions among genes. Due to the conditional independence tests embedded into the algorithm, the interactions inferred by the approach are statistically significant, unlike various threshold based techniques. Moreover, the use of MI/CMI for regulatory interaction assessment ensures that the algorithm can detect nonlinear regulatory interactions. This algorithm, called Bayesian Information Theoretic GRN Reconstruction (BITGRN), and the associated preliminary results for both synthetic and real-life data, were presented and published in the Proceedings of the 11th International Conference on Artificial Intelligence and Applications (AIA'2011) [157].

2. We propose a novel modeling framework that can represent both instantaneous and to begin with - single step time-delayed genetic interactions, for providing a better representation of the biological processes. To harness the benefits of the proposed model, we present two structure learning algorithms. The first of these operates in two stages, and uses a methodology similar to BITGRN; the other algorithm employs a score and search based evolutionary approach. Both these algorithms use the proposed modeling framework, and sequentially learns the two types of genetic interactions. Preliminary results using the first approach, i.e., using the CI

test based technique of BITGRN, was presented and published in the Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence (AI'2011) [152]. The contributions from the second approach, which uses genetic algorithm based search and an information theory based scoring metric, was presented and published in the Proceedings of the 2011 IEEE Congress on Evolutionary Computation (IEEE CEC'2011) [153].

3. Based on the modeling framework developed in the previous contribution, we primarily focus on improving the learning aspect of reconstructing a GRN in this contribution. The first part of this work is modeling related, where we extend the framework proposed in the previous contribution to incorporate higher order time-delayed interactions. Subsequently (and more importantly in the second part of this work), we propose a novel scoring function (called the CCIT metric) that enables detection of both instantaneous and time-delayed interactions simultaneously rather than sequentially. This results in reconstruction techniques which are more realistic and accurate. The initial results using this approach were presented and published in the Proceedings of the 18th International Conference on Neural Information Processing (ICONIP'2011) [158]. Due to the very good performance of the approach on different genetic networks, we later published a thorough study of the proposed approach in a journal article in *BMC Systems Biology* [154].

4. The results from the CCIT-based scoring metric in 3. above use a hill-climbing local search approach for exploring the network space. Due to the incorporation of the multiple step time-delayed interactions, the already large search space for the BN-based modeling becomes even larger. To efficiently explore this huge search space, we propose an evolutionary optimization technique incorporating network motifs whose identification enables us to effectively explore a significantly larger search space. This algorithm, and preliminary results using the algorithm, have been accepted for presentation at the Genetic and Evolutionary Computation Conference (GECCO 2013) [155].

5. Microarray data is noisy and the number of samples from microarray is usually very low. Due to these problems, it should be beneficial to the reconstruction process to use various alternate sources of knowledge in addition to using microarray

data. For this, we propose to incorporate the knowledge obtained from protein-protein interaction networks and other diverse data sources. We propose an iterative, co-learning based fusion algorithm where the present estimate of the gene regulatory network is used for the estimation of the protein-protein interaction network for the next stage, and vice versa. The method and results obtained by applying the method were presented and published in the Proceedings of the 19th International Conference on Neural Information Processing (ICONIP'2012) [156].

6. In order to attain the previously mentioned objective of studying larger scale gene regulatory networks, we study a 730 gene network of *Cyanothece* sp. ATCC 51142. This cyanobacteria is capable of carbon-dioxide sequestering, and it is also capable of producing environment-friendly biofuels. Typical for microarray experiments, the data for this network contains 24 samples. The genes were selected based on a 2-fold change cutoff, and they are responsible for performing some major tasks of the organism such as energy metabolism, respiration, nitrogen fixation, protein translation and photosynthesis, along with several other tasks.

7. One of the ways to qualitatively assess large scale GRNs is to use ontological information from diverse sources. However, the ontological resources are scattered and thus it is difficult to combine ontology information to make assessments. The Systems Biology group at Monash University has built a GO database for cyanobacteria. This GO database has been used in this research for functional category analysis of the large cyanobacteria network mentioned in the previous contribution. The analysis of this important organism using the database has been presented in our journal article [154]. The database is publicly accessible and can be used by any researcher for analyzing cyanobacteria genes.

1.4 Organization of the Thesis

The thesis is organized in seven chapters. While Chapter 1 (Introduction) and Chapter 7 (Conclusion), respectively, provide the introduction and the conclusion to the thesis, organization of the remaining chapters is as follows.

Chapter 2 **Background and Literature Review**. In this chapter, we provide a detailed discussion covering both biological and mathematical background. A literature review, which reviews various modeling and reconstruction techniques of GRNs, is also presented. Further, we also describe the basics of the modeling technique that we use in this thesis, i.e., Bayesian networks, and methods that are used for reconstructing GRNs using Bayesian networks.

Chapter 3 **Information Theoretic Bayesian Approach for Genetic Network Reconstruction**. This chapter uses DBN based modeling to propose a novel information theory based GRN reconstruction technique. Rather than using linear dependency measures such as PCC, we use MI/CMI based regulatory interaction assessment in a DBN framework, and apply important mathematical properties of MI/CMI to formulate conditional independence tests. These rigorous statistical tests ensure that the arcs inferred by the proposed algorithm are highly accurate with correct direction of the arcs, and they can detect nonlinear interactions. We test and show the effectiveness of the algorithm using both synthetic and real-life GRNs.

Chapter 4 **Realistic Modeling of Genetic Interactions**. Chapter 4 proposes a more realistic modeling approach to biological interactions, by allowing both instantaneous and time-delayed arcs in DBN based modeling of GRNs. For this, first we propose a novel framework that can model both instantaneous and single-step time-delayed genetic interactions. Subsequently, we present two different learning algorithms that make use of the proposed modeling framework and sequentially learn the two types of genetic interactions. Both these algorithms are assessed using both synthetic networks and real-life networks of yeast and *E. coli*.

Chapter 5 **Joint Learning of Instantaneous and Multi-Step Time-Delayed Interactions**. This chapter focuses primarily on learning GRNs, and specifically, it extends the modeling framework proposed in Chapter 4 to model both instantaneous and *multi-step* time-delayed genetic interactions. Further, it proposes a novel scoring metric that can score both instantaneous and time-delayed genetic interactions *simultaneously*. This scoring metric, with the instantaneous and multi-step time-delayed interactions based modeling, is applied to synthetic networks, and also real-life networks

of yeast, *E. coli* and cyanobacteria. Finally, we also propose an evolutionary framework (GA based) using the proposed modeling technique and the scoring metric, for effectively exploring a significantly larger search space. The effectiveness of the algorithm is shown using both synthetic networks and real-life networks of yeast and *E. coli*.

Chapter 6 **Co-Learning of GRN and PPIN**. This chapter proposes a fusion based co-learning algorithm for learning GRNs and PPINs. We employ a disjunctive approach to the co-learning using the Bayes theorem. An iterative fusion of information among the GRN and PPIN structures is proposed, that makes novel use of Bayesian marginalization for integrating multiple sources of prior knowledge. Different yeast networks are used for assessing the effectiveness of the proposed algorithm.

In the next Chapter, we provide necessary background information for this thesis work and also carry out an extensive review of the available literature related to GRN modeling and reconstruction.

Chapter 2

Background and Literature Review

One of the major objectives in systems biology research is the elucidation of genetic regulatory interactions occurring within the cell. The work of Kauffman *et al.* [110] is one of the first attempts aimed at this objective, which provided a mathematical formalism for describing GRNs. With the advent of DNA microarray and other data sources, early approaches based on simulation [200] have been replaced by more robust techniques which reconstruct the gene networks from the expression dynamics of the associated genes. Broadly speaking, there are two classes of GRN reconstruction algorithms [16, 79]: those based on the 'physical interaction' approach that aim to identify interactions among transcription factors and their target genes (gene-to-sequence interaction), and those based on the influence interaction approach that relate the expression of a gene to the expression of the other genes in the cell (gene-to-gene interaction), rather than relating it to sequence motifs found in its promoter. Typically, the ensemble of influence interactions are used to model genetic networks.

In this chapter, we discuss the biological background behind gene regulations and also mathematical fundamentals that are used for the reconstruction of GRNs. Further, we review various techniques available in the literature which have been used for modeling and reconstructing these networks. In this thesis, since we use Bayesian networks and its derivative, dynamic Bayesian networks for the modeling of genetic regulatory networks, a detailed discussion on different BN based strategies is also presented. The rest of the chapter is organized as follows. In Section 2.1, we provide a brief introduction to the biological concepts related to gene regulatory networks. Section 2.2 describes

relevant mathematical quantities commonly employed by different GRN reconstruction techniques. Next, in Section 2.3, we review different modeling and learning methods which are commonly used for reconstructing GRNs. Finally, we review the techniques used by BN/DBN based reconstruction methods in Section 2.7.

2.1 Fundamentals of Microbiology

We begin with a brief overview of the basic concepts of molecular biology that are relevant to this thesis. More details can be found in other molecular biology textbooks (for example, see [6, 118, 133]).

2.1.1 Cell

Cells (see Figure 2.1) are the fundamental working units of every living system. The cell was discovered by Robert Hooke in 1665. There are two types of cells: eukaryotic and prokaryotic. Eukaryotic cells (e.g., plants, animals, fungi, protozoa, algae) are characterised by the presence of membrane enclosed subcellular organelles (plasma membrane, nucleus, mitochondria, ribosomes, Golgi apparatus etc.). A basic eukaryotic cell also contains plasma membrane, glycocalyx (components external to the plasma membrane), cytoplasm (semifluid, salty; takes up most of the cell volume), cytoskeleton (microfilaments and microtubules that suspend organelles, give shape, and allow motion). Prokaryotes (e.g., bacteria and archaea), on the other hand, are molecules surrounded by a membrane and cell wall. Prokaryotic cells lack characteristic eukaryotic subcellular membrane enclosed "organelles," but may contain membrane systems inside a cell wall. Prokaryotic cells may also have photosynthetic pigments, (e.g. in cyanobacteria). Some prokaryotic cells have external whip-like flagella (see figure) for locomotion or hair like pili for adhesion. Prokaryotic cells come in multiple shapes like cocci (round), baccilli (rods) and spirochetes (helical cells).

Although a cell is the fundamental unit of all living organisms, it is complicated in terms of both its structure and function. Such complexities are mainly embodied in and regulated by three biological sequences: DNA, RNA and Protein.

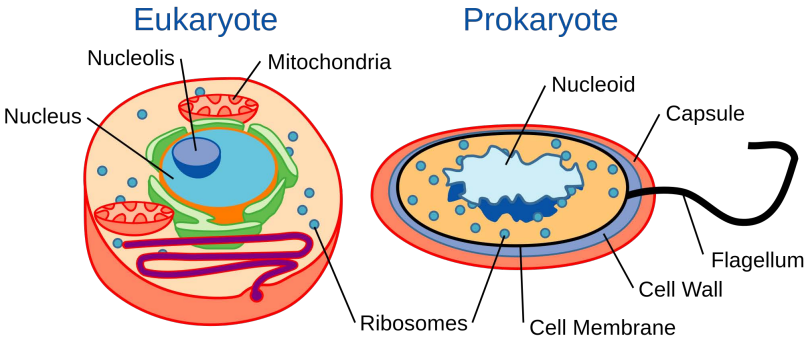

Figure 2.1: Illustration of a Cell. Source: [72].

2.1.2 DNA

DNA (see Figure 2.2) stands for deoxyribonucleic acid which is a structure of sugar, phosphate and a base combined into a complex double helix. The two strands in the helix are complementary to each other, which means that each DNA strand contains the template information for synthesis of a new copy of the other strand. The DNA is situated in the nucleus, and organized into chromosomes. Every cell must contain the genetic information and the DNA is therefore duplicated before a cell divides (a process called replication).

The chemical structure of DNA is shown in Figure 2.3. The building blocks of DNA are the 5-carbon sugar deoxyribose linked together by phosphodiester bonds forming two strands of sugar-phosphate backbones on the outside of the double helix. Each ribose also binds one of four alternative bases: adenine (A), guanine (G), cytosine (C) or thymine (T). The opposing strands are held together by base-pairing between the two strands: G is always paired with C by three hydrogen bonds and A is always paired with T by two hydrogen bonds.

DNA carries the genetic information of a cell and consists of thousands of genes. Each gene serves as a recipe on how to build a protein molecule (via an intermediate step where RNA is produced). Proteins perform important tasks required for proper functioning of the cell, or serve as building blocks. The flow of information from the genes determines the protein composition and thereby the functions of the cell.

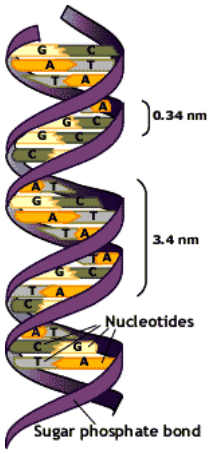

Figure 2.2: Illustration of DNA. Source: [144].

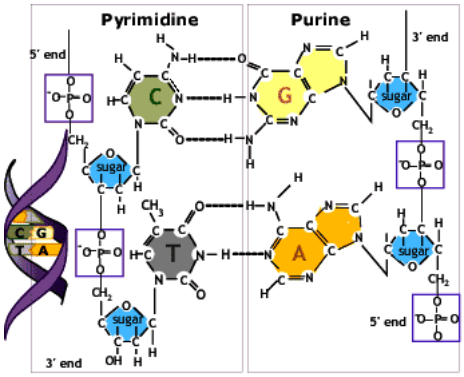

Figure 2.3: Chemical Structure of DNA. Source: [143].

2.1.3 RNA

The chemical structure of RNA is very similar to that of DNA, with two major differences: (1) RNA contains the sugar ribose, while DNA contains the slightly different sugar deoxyribose (a type of ribose that lacks one oxygen atom), and (2) RNA has the nucleobase uracil while DNA contains thymine. Unlike DNA, most RNA molecules are single-stranded and can adopt very complex three-dimensional structures (see Figure 2.4).

RNA can be of three types: 1. tRNA (transfer RNA), 2. rRNA (ribosomal RNA) and 3. mRNA (messenger RNA). Transfer RNA (tRNA) serves as the physical link between the nucleotide sequence of nucleic acids (DNA and RNA) and the amino acid sequence of proteins. Ribosomal RNA (rRNA) is the RNA component of the ribosome, and it is essential for protein synthesis in all living organisms. Finally, messenger RNA (mRNA) is a large family of RNA molecules that convey genetic information from DNA to the ribosome, where they specify the amino acid sequence of the protein products of gene expression. Following transcription of mRNA by RNA polymerase, the mRNA is translated into a protein. However, there are other types of RNAs that do not translate to proteins. These are called non-coding RNAs. A special type of small, non-coding RNA, called miRNA (micro RNA), has recently been discovered [263] to play important roles in post-transcriptional gene regulation.

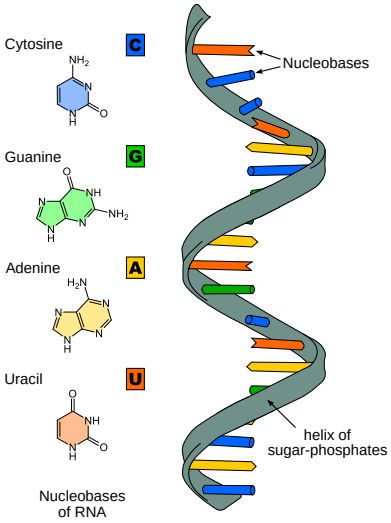

Figure 2.4: Illustration of RNA. Source: [245].

2.1.4 Protein

Proteins are the most versatile and powerful molecules in the body. Proteins are the building blocks of all cells and execute nearly all cell functions. The multiplicity of functions performed by proteins arises from the huge number of different three-dimensional shapes they adopt. Structurally, proteins are polymers of amino acids, joined together by peptide bonds in a long chain, also called a polypeptide chain. Some proteins consist of more than one polypeptide chain and they frequently associate with each other to form larger protein complexes. An example of a protein (hemoglobin) is shown in Figure 2.5.

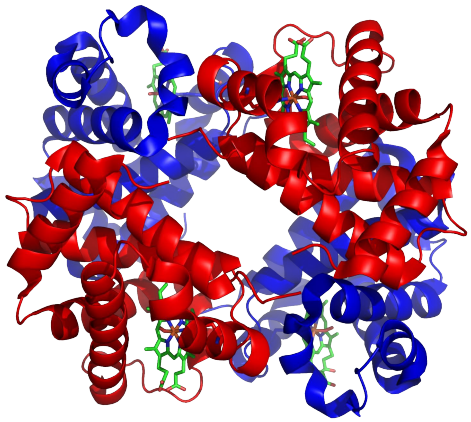

Figure 2.5: Example of a protein (hemoglobin). Source: [243].

2.1.5 Gene

A gene is a piece of DNA fragment which contains genetic information. Genes hold the information to build and maintain an organism's cells and pass genetic traits to offspring. Genes tell cells how to work, control our growth and development, and determine what we look like and how our bodies work. They also play a role in the repair of damaged cells and tissues. The instructions contained in genes get implemented via the proteins. A diagram of a gene in relation to the DNA and chromosome [1] is shown in Figure 2.6.

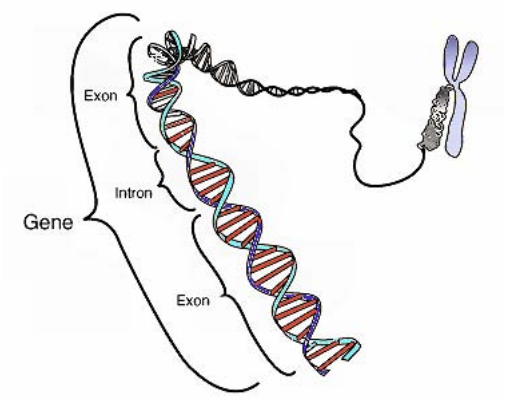

Figure 2.6: Gene. Source: [242].

2.1.6 Central Dogma of Molecular Biology

Although a cell is the fundamental unit of all living organisms, it accomplishes various tasks needed for the successful functioning of the respective organisms using the above mentioned biopolymers, i.e. DNA, RNA and protein. The central dogma of molecular biology describes the flow of genetic information within a biological system, in particular among DNA, RNA and proteins. According to the central dogma, the process of conversion of DNA to protein constitutes the following two major steps (see Figure 2.7).

1. *Transcription*: In all eucaryotic cells a DNA never leaves the nucleus, instead the genetic code (the genes) is copied into RNA which then is decoded (translated) into proteins in the cytoplasm. Transcription is the process whereby the DNA serves as a template to make RNA. The resultant messenger RNA has a nucleotide sequence that is complementary to the DNA from which it was transcribed.

[1] The DNA is ribbon-like in structure, but normally exists in a condensed form called chromosomes.

2. *Translation*: Translation is the actual synthesis of a protein under the direction of mRNA. During this process, the nucleotide sequence of an mRNA (messenger RNA) is translated into the amino acid sequence of a protein.

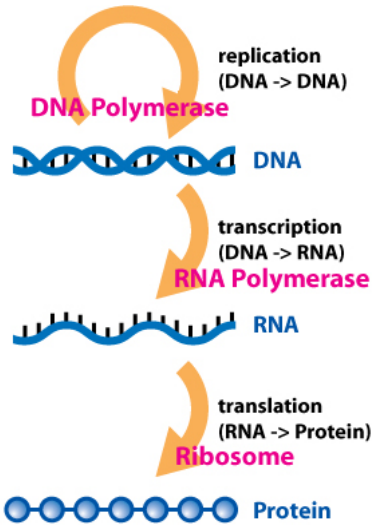

Figure 2.7: Illustration of the Central Dogma of Molecular Biology. Source: [241].

2.1.7 Gene Expression

In all organisms, there are two major steps separating a protein-coding gene from its protein: first, the DNA segment on which the gene resides must be transcribed from DNA to messenger RNA (mRNA); second, it must be translated from mRNA to protein[2]. The process by which information from a gene is used in the synthesis of a biologically functional gene product (RNA or protein) is called gene expression.

Gene expression differs both temporally and spatially [25, 234]. The temporal expression of a gene refers to the process that a gene expresses (or is regulated) at the appropriate time and keeps itself silent otherwise [19, 234]. It also indicates a gene has different expression patterns at different times [4]. For example, the expression patterns of tumor suppressor gene p53 are different at different stages, in modulating cellular functions such as DNA repair, cell cycle arrest, and apoptosis. There is also spatial control of gene expression [198, 234]. Although cells from the same organism have identical genomes,

[2]RNA-coding genes must still go through the first step, but are not translated into protein.

cells in the different parts of an organism may have different gene expression patterns due to the various functions they fulfill [51,63,220].

2.1.8 Gene Regulation

Gene Regulation is the process by which the expressions of a gene is regulated by the expression of a set of parent genes (called transcription factors). There are two types of regulation: positive and negative. Given two genes X and Y, if an expression level of Y is affected by the expression level of X, we say X regulates Y. If an increase in the expression level of X leads to an increase in the expression level of Y, it is a positive regulation; otherwise, it is a negative expression. DNA microarray technology allows us to measure the amount of RNA associated with many genes in parallel, thus facilitating the discovery of genetic regulations.

It may be noted here that epigenetic modifications (histone modifications, DNA methylation), in addition to direct transcription factor based regulations, ultimately regulate gene activity and expression during development and differentiation, or in response to environmental stimuli. However, the effect of these modifications is not direct, and hence they are not directly factored into the discovery of genetic regulations.

2.1.9 Gene Regulatory Networks

Gene regulatory networks (see Figure 2.8 for a schematic representation) are collections of regulatory relations among genes in a genome and they serve as the models that display causal relationships between gene activities [29]. As we know from the central dogma of molecular biology, genes act as the source for producing proteins. These proteins execute various cellular functions. One of the many functions that these proteins (called transcription factors[3]) perform is that they promote or inhibit the expressions of other genes. Thus, we see that although genes do not interact directly with each other, their products (synthesized proteins) in conjunction with other components of the cell regulate the expression of genes in the network. A network depicting these regulatory relationships among genes, where the intermediate components are not taken into account, is called a gene regulatory network .

[3]Some proteins are not involved in regulation, and thus are not transcription factors.

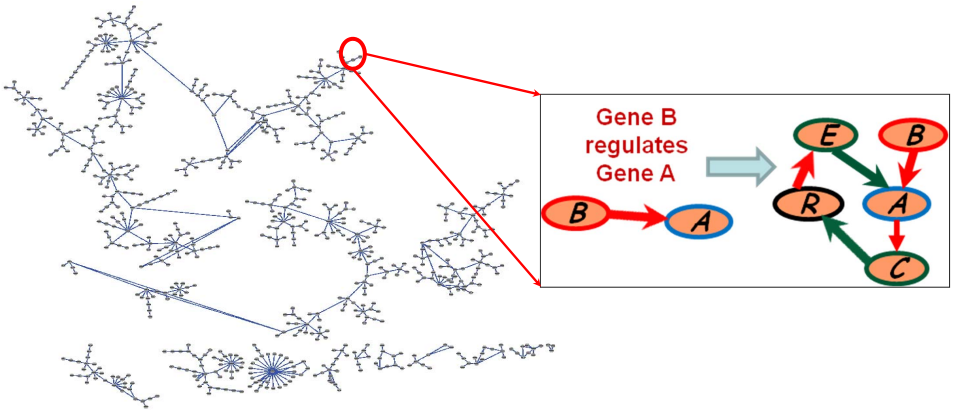

Figure 2.8: Schematic view of a gene regulatory network

Although GRNs are composed primarily of genes regulating other genes, recent discoveries show that [263] miRNAs can downregulate gene expression by base-pairing with the 3′ untranslated regions (3′UTRs) of target messenger RNAs (mRNAs). It was found that in animal cells, miRNAs regulate their targets by translational inhibition and mRNA destabilization. However, it is still unclear whether all organisms show this same behavior. Together with the fact that miRNA regulation data is scarce, it is yet not a fully viable technique for regulatory network reconstruction.

2.1.10 DNA Microarray data

DNA microarrays are 2D arrays of different DNA sequences that allows us to measure in parallel (using gene chips) the amount of RNA associated with many genes and determine which genes are expressed in a particular cell type (see Figure 2.9). There are two main types of microarrays. One is the spotted microarray where two different experimental conditions (each with its own label) are hybridized to one array. With this fabrication method, only relative gene expression values can be estimated. The other type of microarray is the oligonucleotide array where each different condition is hybridized to one array. The absolute values of gene expression can be estimated with this fabrication method. In each cell of this array there is a single stranded DNA sequence or collection of DNA sequences. If the contents of one cell is taken which has been stocked with a given condition, after washing the microarray, the reading of what mRNAs were transcribed can be obtained. This is because the mRNAs that were transcribed at that condition will

bind to its specific cell. By knowing what each cell represents, the level of transcription of every gene in the genome in a particular cell and at a particular condition can be measured. Thus, using DNA microarray based gene expression profiling experiments, the expression levels of thousands of genes can be simultaneously monitored. Microarray datasets are commonly very large (contains thousands of genes' expression values), and analytical precision is influenced by a number of variables. Statistical challenges include taking into account effects of background noise, missing observations and appropriate normalization of the data.

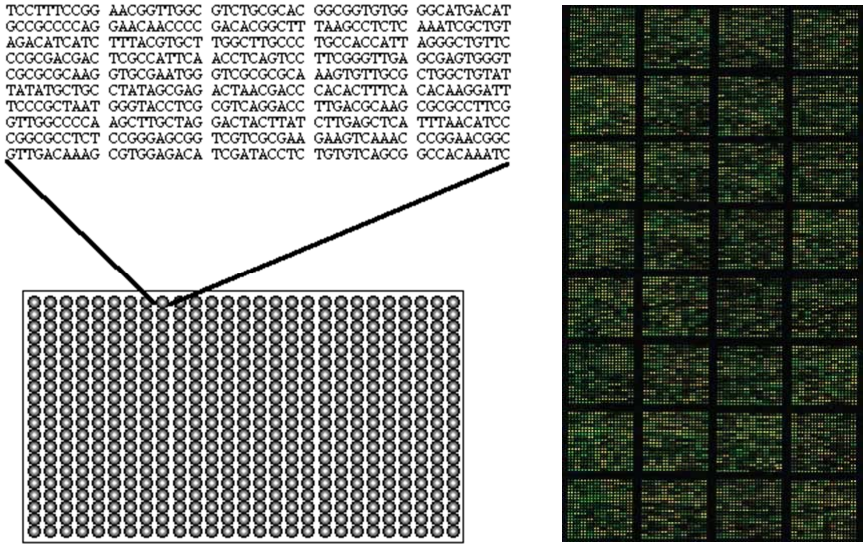

Figure 2.9: A DNA Microarray

It may be noted here that some of these limitations of traditional DNA microarray technologies can be alleviated with a recent technology called NGS (Next Generation Sequencing), which builds on the characteristic that the bases of a small fragment of DNA can be sequentially identified from signals emitted as each fragment is re-synthesized from a DNA template strand. NGS based techniques usually deliver higher quality data with better sensitivity, accuracy and broader dynamic range compared to microarray data. On the other hand, microarray based techniques are widely used, well known, and relatively inexpensive. Integrated genomics approaches can be used to combine NGS and gene expression data, which can then be used to interrogate for genes with both a

regulation event and a change in the mRNA abundance. This way, we can find the instances where regulation patterns affect gene expression levels. However, this integrated approach is still not well established, but this is a promising future direction.

In the next subsection, we introduce functional enrichment analysis using gene ontology, which can complement traditional microarray analysis.

2.1.11 Gene Ontology and Functional Enrichment Analysis

Gene Ontology (GO) is a set of associations from biological phrases to specific genes that are either chosen by trained curators or generated automatically. GO is designed to rigorously encapsulate the known relationships between biological terms and and all genes that are instances of these terms. The GO associations allow biologists to make inferences about groups of genes instead of investigating each one individually.

To centralize and disseminate a wealth of prior knowledge about known genes, the Gene Ontology [14] database was formed, which allows researchers to assign attributes to groups of genes that emerge from their experiments or analyses. The Gene Ontology (GO) project [23] is a collaborative effort to address the need for consistent descriptions of gene products in different databases. It provides an ontology of defined terms representing gene product properties.

GO terms are organized hierarchically such that higher level terms are more general and thus are assigned to more genes, and more specific terms are related to parents by either “is a” or “part of” relationships. The relationships form a directed acyclic graph (DAG), where each term can have one or more parents and zero or more children [53] (see Figure 2.10). The terms are separated into three categories/ontologies:

1. Cellular Component, which describes where in the cell a gene acts, what organelle a gene product functions in, or what functional complex an enzyme is part of.

2. Molecular Function, which describes activities, such as catalytic or binding activities, that occur at the molecular level. It defines the function carried out by a gene product (one product may carry out many functions).

3. Biological Process, which is a series of events accomplished by one or more ordered assemblies of molecular functions. Examples of biological process terms are cellular physiological process or signal transduction, DNA replication, limb formation, etc.

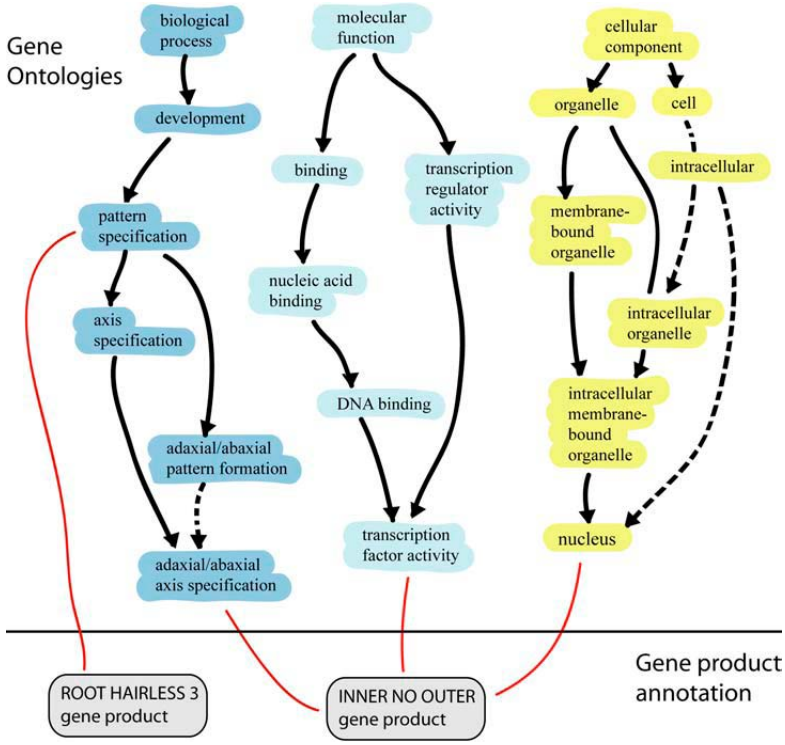

Figure 2.10: Gene Ontologies. Sections of the three ontologies are represented schematically with only term names shown. The biological process ontology is shown on the left side (dark blue background), the molecular function ontology is shown in the center (light blue background), and the cellular component ontology is shown on the right side (yellow background). Source: [53].

GO annotations can be used to complement traditional microarray analysis. Once low level analysis is complete and a group of differentially expressed or significantly affected genes is selected, enrichment of GO attributes within the group can be assessed. Many tools exist to address this problem. Given a background gene set (i.e., all genes on the array), and a subset of interesting genes (e.g., all those that are differentially expressed), the task of these analysis programs is to identify which GO terms are most commonly associated with this subset and test the claim that this association (enrichment) is significantly different from what would be expected by chance, based on the proportions of genes out of the total having each attribute. Examples of tools to determine whether such over-representation is significant in general can be found at [130].

2.1.12 Protein-Protein Interaction Networks

Interactions between proteins are important for the majority of biological functions. Many of the most important molecular processes in the cell, such as DNA replication, are carried out by large molecular machines that are built from a large number of protein components organised by their protein-protein interactions. Protein-protein interactions occur when two or more proteins bind together (see Figure 2.11). There are various types of protein interactions depending on various criteria. If a protein can form stable crystal structure of its own (without any other associated protein) in vivo, then the complexes formed by such proteins are called "non-obligate protein interaction". On the other hand, some proteins cannot create a crystal structure alone, but can be found as a part of a protein complex which creates a stable crystal structure. Such protein complexes are called "obligate protein interaction". Also, there are transient protein interactions which form and break down transiently in vivo, whereas permanent complexes don't show such behavior but is typically dissociated by proteolysis. Typically, the obligate interactions (protein-protein interactions in an obligate complex) are permanent, whereas non-obligate interactions have been found to be either permanent or transient [10].

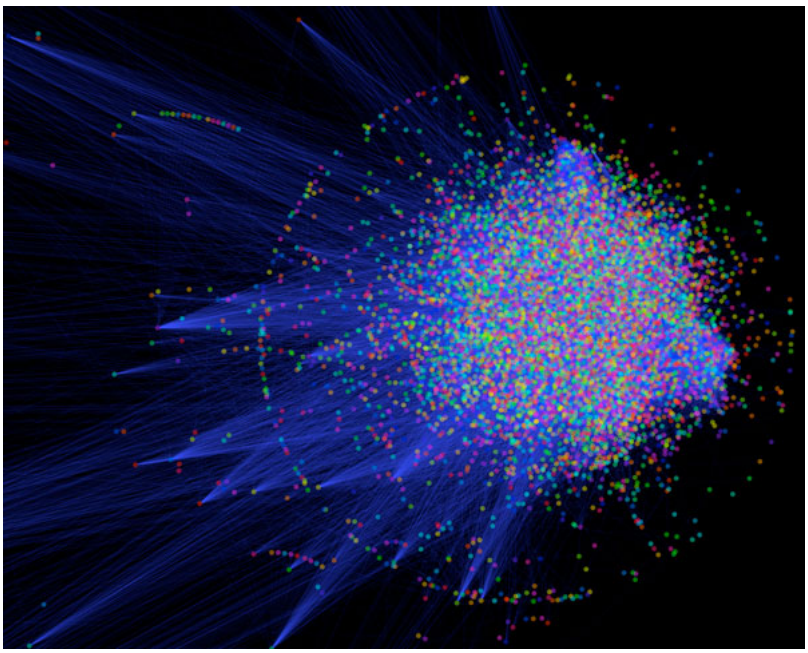

Figure 2.11: Protein-Protein Interaction Network (the human interactome). Each point in the figure represents a protein and each blue line between them is an interaction. Source: [244].

2.1.13 Protein Interaction Data

There are a multitude of methods to detect protein-protein interactions. Each of the approaches has its own strengths and weaknesses, especially with regard to the two performance metrics, sensitivity and specificity, of the method. Some of the most used methods include Yeast two hybrid assay (Y2H), phage display, and protein microarray [254].

Two-hybrid assay (Y2H) is a molecular biology technique that can be used to discover protein-protein interactions [254] by testing for physical interactions (such as binding) between two proteins. The premise behind the test is the activation of downstream reporter genes by the binding of a transcription factor onto an upstream activating sequence (UAS). For two-hybrid screening, the transcription factor is split into two separate fragments, called the binding domain (BD) and activating domain (AD). The BD is the domain responsible for binding to the UAS and the AD is the domain responsible for the activation of transcription [107, 254]. Phage display is a laboratory technique for the study of protein-protein interactions that uses bacteriophages to connect proteins with the genetic information that encodes them [254]. Phage display [209] was first described by George P. Smith in 1985, when he demonstrated the display of peptides on filamentous phage by fusing the peptide of interest on to gene III of filamentous phage. Like the two-hybrid system, phage display is used for the high-throughput screening of protein interactions. A protein microarray (or protein chip) is a high-throughput method used to track the interactions and activities of proteins [145]. Its main advantage lies in the fact that large numbers of proteins can be tracked in parallel. The chip consists of a support surface such as a glass slide, nitrocellulose membrane, bead, or microtitre plate, to which an array of capture proteins is bound [73]. Probe molecules, typically labeled with a fluorescent dye, are added to the array. Any reaction between the probe and the immobilised protein emits a fluorescent signal that is read by a laser scanner [74]. Protein microarrays are rapid, automated, economical, and highly sensitive, consuming small quantities of samples and reagents [148]. The high-throughput technology behind the protein microarray is comparatively easy to develop since it is based on the previously-developed DNA microarray technology [89].

Protein-protein interaction information is collected in several databases that make the data and the evidence behind it easily accessible and allow different mechanisms

to query and display the data [12, 36, 100, 166, 188, 193, 216]. These resources are very useful for researchers interested in checking a small number of particular proteins of interest. However, PPI data can also be used globally for systematic network analyses, prediction of protein properties, and evaluation of novel datasets of PPIs produced in a high-throughput fashion. With the objective of creating a resource allowing the selection of PPIs by experimental confidence cut-offs, Schaefer *et al.* [193] generated HIPPIE (Human Integrated Protein-Protein Interaction rEference), a scored human PPI collection integrated from multiple sources. Following Ceol *et al.* [36], Schaefer *et al.* [193] developed an expertly curated scoring scheme that takes into account the reliability of different experimental evidence in the definition of a PPI combining three types of information: experimental techniques used, number of studies finding the PPI, and reproducibility in model organisms. HIPPIE currently integrates $72,916$ interactions from several public PPI resources (scored according to confidence). For comparison, the complete human interactome map has been estimated to contain between $2,00,000$ and $4,00,000$ interactions (according to [26] and [179], respectively) suggesting that the knowledge of the human interactome is still incomplete.

Several other resources have been created that, like HIPPIE, integrate PPI data from multiple sources, but do not have a focus on distributing a simple scored dataset (e.g., iRefWeb [230]), or do not focus on experimentally verified interactions (e.g., STRING [225]). Some other databases offer a continuous confidence scoring scheme, (e.g., MINT [36] and HAPPI [40]), but they do not allow batch scoring of PPI sets or the exclusive retrieval of high confidence interactions and lack the integration of several important high-throughput experimental datasets.

2.2 Mathematical Preliminaries

In this section, we present relevant mathematical quantities that are commonly used for assessing regulatory interactions among genes, thus aiding the reconstruction of gene regulatory networks.

2.2.1 Random Variables and Entropy

In probability and statistics, a random variable or stochastic variable is a variable whose value is subject to variations due to chance (i.e. randomness, in a mathematical sense). As opposed to other mathematical variables, a random variable conceptually does not have a single, fixed value (even if unknown); rather, it can take on a set of possible different values, each with an associated probability. Formally,

Definition 2.1. ***Random Variable.*** *Given a probability space (Ω, P), a random variable X is a function whose domain is Ω. The range of X is called the* ***space*** *of X.*

For a random variable X, we use $X = x$ to denote the subset containing all elements $e \in \Omega$ that X maps to the value of x. The Shannon information content of an outcome x is defined to be the logarithm of the multiplicative inverse of the probability of the[4] outcome:

$$h(x) = log_2 \frac{1}{P(x)} \tag{2.1}$$

and it is measured in bits. The entropy of a random variable X is defined to be the average Shannon information content of all the outcomes:

$$H(X) = \sum_x P(x) log \frac{1}{P(x)} \tag{2.2}$$

with the convention that for $P(x) = 0$, $0 \times log\frac{1}{0} \equiv 0$ (considering limiting values). Shannon entropy is the average unpredictability in a random variable, which is equivalent to its information content. The concept was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication" [202, 203].

Next, we move on to the definition of joint entropy and conditional entropy:

Definition 2.2. ***Joint Entropy.*** *Joint entropy of a pair of two discrete random variables X and Y is:*

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{2.3}$$

Similarly, conditional entropy is defined as follows:

[4] In future, unless otherwise stated, we will consider 2-based logarithm.

Definition 2.3. ***Conditional Entropy.*** *The conditional entropy Y given a random variable X (average over X) is:*

$$H(Y|X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)[\log p(y|x)] \tag{2.4}$$

One very important property of H is, that we always have $H(X) \geq 0$.

2.2.2 Kullback-Leibler (KL) Divergence

In probability theory and information theory, the Kullback-Leibler divergence (or relative entropy) [124] is a non-symmetric measure of the difference between two probability distributions P and Q. Specifically, the Kullback-Leibler divergence of Q from P is a measure of the information lost when Q is used to approximate P. Formally,

$$D_{KL}(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \tag{2.5}$$

Similar to entropy, $D_{KL} \geq 0$, with equality only if $P = Q$. One interesting property of the divergence is that in general the relative entropy is not symmetric under interchange of the distributions P and Q (i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$).

2.2.3 Mutual Information (MI)

MI measures the amount of information that can be obtained about one random variable by observing another variable [37, 55]. Formally, it is defined by Equation 2.6.

$$MI\,(X,Y) = \sum_{x,y} P\,(x,y) \log \frac{P\,(x,y)}{P\,(x)\,P\,(y)} \tag{2.6}$$

where $P(x,y)$ is the cross-time joint probability and $P(x)$, $P(y)$ are the marginal probabilities. In terms of entropy, MI can also be defined as:

$$\boldsymbol{MI(X,Y) = H(X) + H(Y) - H(X,Y)} \tag{2.7}$$

Consider the below data for which, the calculation of MI will be as follows:

$$H(X) = -\left(0.3 \log 0.3 + 0.7 \log 0.7\right) = 0.8813$$

X	1	1	0	1	0	1	1	1	0	1
Y	0	0	1	0	1	1	1	1	0	1

Figure 2.12: Example of calculation of Mutual Information (MI)

$$H(Y) = -\left(0.4 \log 0.4 + 0.6 \log 0.6\right) = 0.971$$

$$H(X,Y) = -\left(0.1 \log 0.1 + 0.2 \log 0.2 + 0.3 \log 0.3 + 0.4 \log 0.4\right) = 1.8464$$

$$\boldsymbol{MI(X,Y) = H(X) + H(Y) - H(X,Y) = 0.0059}$$

2.2.4 Conditional Mutual Information (CMI)

CMI is the reduction in the uncertainty of X due to knowledge of Y when Z is given [37], [55]. The CMI of random variables X and Y given Z is defined in Equation 2.8.

$$MI\left(X,Y|Z\right) = \sum_{x,y,z} P\left(x,y,z\right) \log \frac{P\left(x,y|z\right)}{P\left(x|z\right) P\left(y|z\right)} \tag{2.8}$$

where $P(x,y,z)$ is the cross-time joint probability, $P(x,y|z)$ is the conditional cross-time joint probability and $P(x|z)$ and $P(y|z)$ represent conditional marginal probabilities.

In terms of entropy, CMI can also be defined as:

$$\boldsymbol{MI(X,Y|Z) = H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z)} \tag{2.9}$$

Consider the below data, for which the calculation of CMI will be as follows:

X	1	1	0	0	0	1	1	1	0	1
Y	0	1	1	0	1	1	0	1	0	1
Y	1	1	1	0	0	1	1	0	0	1

Figure 2.13: Example of calculation of Conditional Mutual Information (CMI)

$$H(Z) = -\left(0.4 \log 0.4 + 0.6 \log 0.6\right) = 0.971$$

$$H(X,Z) = -\left(0.3 \log 0.3 + 0.1 \log 0.1 + 0.1 \log 0.1 + 0.5 \log 0.5\right) = 1.685$$

$$H(Y,Z) = -\left(0.2 \log 0.2 + 0.2 \log 0.2 + 0.2 \log 0.2 + 0.4 \log 0.4\right) = 1.922$$

$$H(X,Y,Z) = -(0.2\log 0.2 + 0 + 0.1\log 0.1 + 0.1\log 0.1 + 0 + 0.2\log 0.2 + 0.1\log 0.1 + 0.3\log 0.3) = 2.446$$

$$\boldsymbol{MI(X,Y|Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) = 0.19}$$

2.2.5 Other Measures for Assessing Associativity

Other measures like Pearson Correlation Coefficient and Partial Correlation Coefficient (PCC) can also be used for the analysis of genetic regulations.

Pearson Correlation Coefficient

Correlation between variables is a measure of how well the variables are related. The most common measure of correlation in statistics is the Pearson Correlation (also called the Pearson Product Moment Correlation or PPMC), which shows the linear relationship between two variables [238]. Mathematically,

$$r_{xy} = \frac{N\sum xy - \sum x \sum y}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}} \tag{2.10}$$

where N is the number of samples. Correlation values are between -1 and 1. A result of -1 means a perfect negative correlation, while a result of 1 means that there is a perfect positive correlation between the two variables. A result of 0 means that there is no linear relationship between the two variables. In practice, we rarely get a correlation of 0, -1 or 1, and the results fall somewhere in between. The closer the value of r gets to zero, the greater the variation the data points are, around the line of best fit.

- High correlation: 0.5 to 1.0 or -0.5 to 1.0
- Medium correlation: 0.3 to .5 or -0.3 to .5
- Low correlation: 0.1 to 0.29 or -0.1 to -0.29

PPMC does not differentiate between dependent and independent variables. For example, if the correlation between a high caloric diet and diabetes is 0.85, switching the variables around will also result in the same PPMC. This would mean that diabetes causes a high caloric diet, which would make no sense.

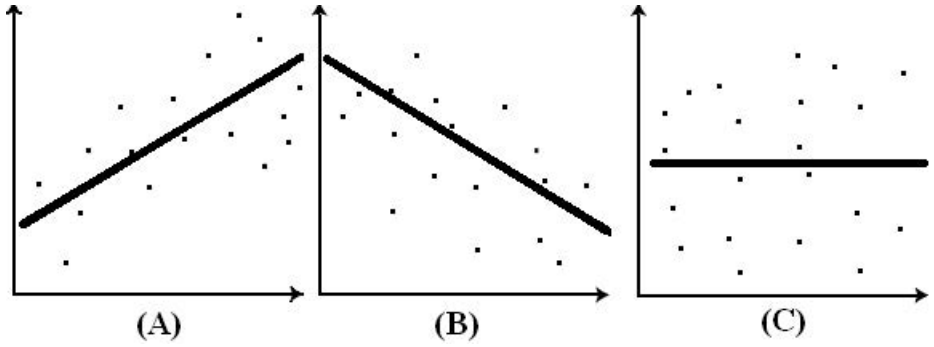

Figure 2.14: Different types of Pearson Correlations. (A) Positive correlation, (B) Negative correlation, (C) Zero correlation.

Partial Correlation Coefficient

A partial correlation corresponds to the correlation between two variables when one variable controls the effect of the other variable. Partial correlations are significant because they help in determining whether correlated variables are linked directly or otherwise and to detect whether the correlation is spurious [177]. Formally, the partial correlation coefficient ($r_{xy.z}$) between gene X and gene Y given a controlling gene, Z is calculated using the following equation:

$$r_{xy.z} = \frac{r_{xy} - r_{xz}r_{zy}}{\sqrt{(1 - r_{xz}^2)(1 - r_{zy}^2)}} \tag{2.11}$$

where r_{xz}, r_{xy} and r_{yz} are Pearson correlation coefficients over the expression profiles of pairs of genes. A zero or a small partial correlation coefficient indicates that the variables

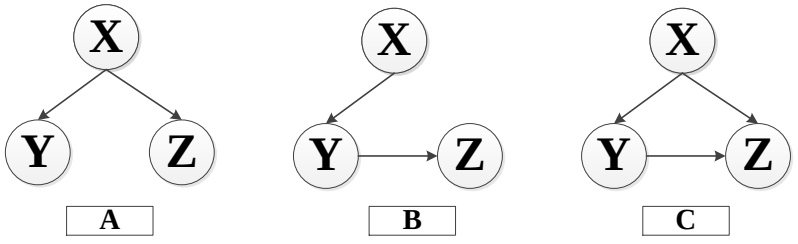

Figure 2.15: Possible inference results of the causal relations among three variables using PCC. (A) True/direct interactions, (B) indirect interaction inference, (C) bivariate inference [256].

are connected by a path that does not have a third variable involved. In most cases a

partial correlation of the general form $r_{xy.z}$ will turn out smaller than the original correlation r_{xy}. In those cases where it turns out larger, the third variable, Z, is typically spoken of as a supressor variable on the assumption that it is suppressing the larger correlation that would appear between X and Y if Z were held constant. As an example, consider Figure 2.15. Variable Y is highly correlated with Z because of the causal effects from X (Figure 2.15(A)). Pearson correlation may give rise to many false positives as in Figure 2.15(C), and Figure 2.15(B), which may be probable for methods that do not account for conditional independence. However, partial correlation tests the correlation between two variables after the linear effects from the rest of the data are removed, hence no relationship exists between Y and Z after the effect from X is removed[5]. Conditional independence, although by itself is insufficient to denote a causal link, can be a powerful tool for removing indirect relationships. Therefore, when inferring the relationship between two gene expression profiles, the other expression profiles can be taken into account to discriminate between direct (Figure 2.15(A)) and indirect (Figure 2.15(B) and (C)) interactions [256].

2.3 Reconstructing Gene Regulatory Networks

Having introduced the mathematical notions required, in the next section we review different techniques used for the modeling and reconstruction of gene regulatory networks. We start with comparatively simple techniques such as clustering based algorithms and boolean network based techniques, and then describe approaches with medium-to-high complexity like graphical Gaussian models, relevance network based methods, differential equation based models and finally, Bayesian network based methods.

2.3.1 Clustering Based Approaches

One of the main problems that hinder research on gene network reconstruction is the curse of dimensionality, i.e. there are many genes with only a few samples. A useful approach to overcome this is to cluster genes with similar expression patterns into clusters, then infer the regulatory relationship among the clusters. Researchers believe genes with similar expression patterns have similar functions or are involved in the same biological

[5]Note that partial correlation only infers undirected relationships.

events [68]. Clustering based approaches attempt to locate groups of genes that have similar expression patterns over a set of experiments [9, 22, 71, 147]. The genes in each group are then postulated to have similar mechanisms of regulation, and are therefore assumed to be functionally related.

Often, a sequence motif finding procedure is applied to the promoter regions of the genes in each cluster, in order to find putative binding sites of the clusters common regulators [257]. The most common clustering approach is hierarchical clustering by Eisen *et al.* [16, 71], where relationships among genes are represented by a tree whose branch lengths reflect the degree of similarity between genes, as assessed by a pairwise similarity function such as Pearson correlation coefficient. For a set of n gene profiles, all the pairwise correlation coefficients are computed; the highest value (representing the most similar pair of genes) is selected and a node in the tree is created for this gene pair with a new expression profile given by the average of the two profiles. The process is repeated by replacing the two genes with a single node, and all pairwise correlations among the $n - 1$ profiles are computed [16]. The process stops when only one element remains. Clusters are obtained by cutting the tree at a specified branch level. Other techniques for clustering also exist. Someren *et al.* [210] reduced 2467 yeast genes into clusters and represented each cluster by a 'prototype' gene calculated from the cluster. A linear model of the prototype genes is then generated by linear regression. D'Haeseleer and others [33, 44, 64, 71, 80] proposed grouping genes into clusters, and then find the representative genes for the clusters. Ram *et al.* [178] proposed a fuzzy logic based clustering approach for searching regulatory triplets by means of predicting changes in expression level of the target over interval time points based on input expression level, and comparing them with actual changes. They applied the method on *Saccharomyces cerevisiae* data and 548 activator/repressor regulatory triplets were inferred from the data.

2.3.2 Boolean Network Based Methods

In boolean networks based models [109, 211] each gene is assumed to be a boolean variable, which can be in one of the two states: *on* or *off*. The dynamics are modeled over a discrete series of time points. One of the main objectives of boolean network models is to study the logical interactions of genes without knowing specific details [113, 207]. It

uses boolean functions (AND, OR, NOR, NAND etc.) to define the gene relationships. The state of each gene is determined by these boolean functions of some of the other genes at the previous time step. Different algorithms have been proposed for inferring the network structure of such models from observations [3, 132], typically by employing information-theoretic entities.

Liang *et al.* [132] proposed REVEAL (REVerse Engineering ALgorithm) which uses information theoretic principles to reduce the search space and establish how the given genes are connected in the networks, and then determines the functions that specify the interactions among genes. To decrease complexity in enumerating all possible state transitions, a maximum fan-in, k ($1 \leq k \leq n$ where n is the number of genes in the dataset), is applied to each gene. An implementation of the algorithm proved to be capable of reliably reproducing networks with $n = 50$ and $k = 3$ given 100 state transition pairs (out of 1015 possible pairs). Akutsu *et al.* [3] later proved that only $O(\log n)$ state transition pairs (from $2n$ pairs) are necessary and sufficient to identify the original Boolean network, and extended the Boolean network model to a qualitative network. Although the simplicity of a Boolean network allows analysis of large networks efficiently, it does not utilise a lot of useful information such as information related to detailed quantity and time delay. Several improvements of noolean networks, such as Generalized Logical Networks [5], Fuzzy Logic Models [252] and Probabilistic Boolean Networks [208], continue to be limited by similar constraints.

2.3.3 Differential Equation Based Methods

Differential equations (DE) are the starting point for quantitative modeling of complex systems. DEs are continuous and deterministic modeling formalisms, capable of describing non-linear and emerging phenomena of complex dynamical systems. DE models of gene networks are based on rate equations, quantifying the rate of change of gene expression as a function of the expressions of other genes (and possibly other quantities). The general form of the equations, one for each of n genes, is:

$$\frac{dx_i}{dt} = f_i(x_{i_1}, x_{i_2}, \ldots x_{i_m}) \tag{2.12}$$

where each x_k is a continuous function, representing the gene expression of gene k. Each $f_i(\cdot)$ quantifies the combined effect of its arguments, or regulators, on x_i, and it subsumes all the biochemical effects of molecular interactions and degradation.

The simplest interesting form that the $f_i(\cdot)$ quantities can take are linear additive functions [68, 69], for which the above general equation 2.12 becomes:

$$\frac{dx_i(t)}{dt} = b_i u(t) + w_{i1}x_1(t) + \ldots + w_{in}x_n(t) \tag{2.13}$$

where the term $b_i u(t)$ indicates a controlled external influence on gene i, like an external perturbation (b_i represents the effect of the external pertrurbation on x_i and $u(t)$ represents the external perturbation at time t). The weight parameters (w_{ij}) indicate the degradation rate of mRNAs or any environmental effects on gene i's expression. Other network models exist, which are based on extensions of the linear model (e.g. co-expression networks [136, 226]). The major limitation of such models is the assumption of linear relationship because in reality, biological relations are usually highly nonlinear.

Chen *et al.* in [41] translated the problem of finding a solution to finding the weights when the number of nonzero weights w_{ij} for any given i is at most a fixed constant k, into a combinatorial problem called Minimum Weight Solutions to Linear Equations, and showed that it is polynomially solvable in general, although they offered a computationally expensive algorithm. Yeung *et al.* [253] used Singular Value Decomposition on time-course experiments to generate an initial solution and then refined it by using an optimization technique called robust regression. The solutions were much better than those from using SVD alone. Experimentally, the authors used very fine sampling times in that study which allowed them to approximate various $\frac{dx_i}{dt}$ values. In [226], Tegner *et al.* used the same linear model, but with advanced gene perturbation (over-expression) technology (described in [78]), and measurements at steady-states. Someren *et al.* [233] used clustering of the time-course expression matrix to reduce the dimensionality of the weight matrix. Hierarchical (progressive) clustering was performed until the resulting linear system had the smallest error in explaining the whole data, i.e., was close to being over-constrained. Their approach drew a lot from the success of clustering in identifying

coregulated clusters of genes through coexpression, but it has its limitations too: the resulting gene network is a network of gene clusters and not genes, and the interpretation is non-trivial.

The NIR algorithm [78] computes the edges w_{ij} from steady-state gene expression (and based on the observation that for steady state, the rate of change of concentration will be zero). NIR needs, as input, the gene expression profiles following each perturbation experiment, knowledge of which genes have been directly perturbed in each perturbation experiment and optionally, the standard deviation of replicate measurements. NIR is based on a network sparsity assumption, that is, a maximum number of incoming edges per gene (i.e., maximum number of regulators per gene), which can be chosen by the user. The output is in matrix format, where each element (i, j) corresponds to the edge between genes i and j. Under the steady state assumption, the inference algorithm reduces to solving equation 2.13 for the unknown parameters w_{ij}, that is, a classic linear regression problem.

The TSNI (Time Series Network Identification) algorithm [17] identifies the gene network (w_{ij}) as well as the direct targets of the perturbations. TSNI is applied on time-series data. To solve equation 2.13, we need the values of the derivatives of the concentrations of the genes, for each gene i and each time point t. This can be estimated directly from the time-series of gene expression profiles. TSNI assumes that a single perturbation experiment is performed (e.g., treatment with a compound, gene over-expression) and N time points following the perturbation are measured (rather than N different conditions at steady-state as for NIR). For small networks (tens of genes), it is able to correctly infer the network structure. For large networks (hundreds of genes), its performance is best for predicting the direct targets of a perturbation (i.e., b_i) (for example, finding the direct targets of a transcription factor from gene expression time series following over-expression of the factor).

S-System, proposed by Savageau [191], is a well known system for biochemical network and attracted attention for GRN inference from late 90s [115]. For n genes, S-System model is given by

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{n} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} x_j^{h_{ij}} \tag{2.14}$$

where, X_i is the expression level of the i-th gene. Non-negative parameters α_i and β_i are called rate constants and real-valued exponents g_{ij} and h_{ij} are referred as Kinetic Orders. If $g_{ij} = 0$, it implies that there is no activation or inhibition from gene j to gene i. If $g_{ij} > 0$, gene j activates gene i and if $g_{ij} < 0$, gene j inhibits gene i. Compared to g_{ij}, the term h_{ij} has an opposite effect on the genes i and j. To infer a GRN of n genes using S-System model, $2 * n(n + 1)$ parameters must be estimated. For inferring GRN using S-System, Noman *et al.* [162–164] used an approach called Trigonometric Differential Evolution (TDE) approach [162], which could infer a relatively large network of 30 genes. An improvement was proposed in Chowdhury *et al.* [50], where the authors use several features for accurate network inference, namely a Prediction Initialization (PI) algorithm to initialize the individuals, a Flip Operation (FO) for better mating of values, a restricted execution of Hill Climbing Local Search over few individuals and a refinement technique which utilizes the fit solutions of the genetic algorithm for improving sensitivity and specificity of the inferred network.

Although an attractive approach for GRN reconstruction would be the deduction of a detailed mathematical description of the entire system in terms of a set of coupled nonlinear differential equations from dynamic (time series) data, in reality, however, multiple parameter sets of nonlinear systems of differential equations can offer equally plausible solutions, and standard optimization techniques in high-dimensional multimodal parameter spaces are not robust and do not provide a reliable indication of the confidence intervals. More importantly, model selection would be impeded by the fact that more complex pathway models would always provide a better explanation of the data than less complex ones, rendering this approach intrinsically vulnerable to over-fitting. Finally, although S-system and overall differential equation based techniques model detailed quantities changing over time, due to having many network parameters, they need more measurements and detailed kinetic information. As a result, they can usually model small biochemical networks [141, 163], e.g., a group of genes of interest.

2.3.4 Relevance Network Based Methods

Relevance networks or correlation networks are essentially model free methods. Relevance networks are networks of highly correlated genes [34, 151]. Edges connect pairs of

genes with correlation coefficient over a certain threshold. Given a measure of association and a threshold value defined, for all pairs of domain variables (X, Y), association $A(X, Y)$ is computed. Variables X and Y are connected by an undirected edge when association $A(X, Y)$ exceeds the predefined threshold value. One of the measures of association is the mutual information (MI) [33, 91]. Correlation networks cluster genes naturally without a pre-assigned cluster number. Different from the classic clustering methods, correlation networks keep the strongest pair-wise association between genes, which contain relevant information for functional interpretation of the genes and their relationships. However, like clustering, the relationships between genes in a correlation network are mostly co-regulation, and not causal relationship. Relevance network based algorithms using information theoretic quantities have been widely used in the literature [37,258,261]. However, they are mostly threshold based techniques, and determining the appropriate threshold is difficult and error prone.

The ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) algorithm has been proposed in [139], and it is essentially a model free method. It is a two-step algorithm; the first step is for network construction and second step is for network pruning. ARACNE identifies candidate interactions by estimating pairwise gene expression profile mutual information, $MI(X, Y)$ that is zero if the joint distribution between the expression level of gene X and gene Y satisfies $P(X, Y) = P(X)P(Y)$. ARACNE estimates MI using a Gaussian kernel estimator. Since MI is reparameterization invariant, ARACNE copula-transforms (i.e., rank-orders) the profiles before MI estimation. After this step, the MIs are filtered using an appropriate threshold, thus removing many of the indirect interactions using the data processing inequality (DPI). ARACNE eliminates all edges for which the null hypothesis of mutually independent genes cannot be ruled out. TimeDelay-ARACNE [261] tries to extend ARACNE to time-course data. The idea on which TimeDelay-ARACNE is based comes from the consideration that the expression of a gene at a certain time could depend on the expression level of other genes at the previous time point or at very few time points before. TimeDelay-ARACNE is a 3-step algorithm: it first detects, for all genes, the time point of the initial changes in the expression; secondly, there is network construction based on time-delayed Mutual Information calculation; and finally, it performs network pruning using DPI twice. It has shown good

performance in the reconstruction of small biological directed networks from time series data.

2.3.5 Graphical Gaussian Models

Graphical Gaussian Models (GGMs) [117, 194] are a class of graphical models related to correlation networks. They are also known as "covariance selection" or "concentration graph" models. The key idea behind GGMs is to use partial correlations as a measure of independence of any two genes conditioned on all other genes [6]. Edges in GGMs represent high conditional dependency, i.e. direct rather than indirect relationships. In contrast, correlation networks define relationships between genes through standard correlation coefficients. Edges in correlation networks only represent high marginal dependency without telling direct vs indirect relationships. Therefore, GGMs are considered a more accurate model over correlation networks for gene regulatory network reconstruction [195]. However, GGMs assume multivariate normality, which is frequently not the case for real biological systems.

In the following sections, we discuss the formalizations behind the main concepts involved in this thesis: Bayesian networks (BN) and dynamic Bayesian Networks (DBN). Further, we discuss different algorithms that use BN and DBN based modeling techniques for the reconstruction of GRNs.

2.4 Bayesian Network

We start with some definitions. In the reminder of the thesis, we use capital letters, such as X, Y, Z, for variable names and lowercase letters x, y, z to denote specific values taken by those variables. Sets of variables are denoted by boldface capital letters $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$, and assignments of values to the variables in these sets are denoted by boldface lowercase letters $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$. Now, consider a finite set $\boldsymbol{X}$ of n random variables, $\boldsymbol{X} = X_1, \ldots, X_n$. We define the notion of conditional independence as follows:

[6] Note that partial correlations are related to the inverse of the correlation matrix.

Definition 2.4. ***Conditional Independence.*** *Two variables (or set of variables) $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given $\boldsymbol{Z}$ (written $(\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z})$), if*

$$P(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{Z}) = P(\boldsymbol{X}|\boldsymbol{Z}) \tag{2.15}$$

Next, we define the notion of Markov assumption which is the bridge between the graphical representation of a BN and the conditional independence relations that it entails.

Assumption 2.4.1. ***Local Markov Assumption.*** *In a Directed Acyclic Graph (DAG) G, if $Pa(X_i)$ denotes the parents of variable X_i, we say that G encodes the local Markov assumptions over the variable set $\boldsymbol{X}$ of G if, each variable X_i is independent of its non-descendants, given its parents in G. The set of these assumptions are written as $\boldsymbol{Markov}(G)$.*

Based on the above definitions and assumptions, we can now define Bayesian Networks:

Definition 2.5. ***Bayesian Network.*** *A Bayesian network is a representation of a joint probability distribution, consisting of two components. The first component, G, is a directed acyclic graph (DAG) whose vertices correspond to the random variables $\boldsymbol{X} = X_1, X_2, \ldots, X_n$, and whose structure encodes the Markov assumptions **Markov**(G) over $\boldsymbol{X}$. The second component, $\boldsymbol{\theta}$, describes a conditional probability distribution, $P(X_i|Pa(X_i))$ for each variable X_i in $\boldsymbol{X}$ [168].*

The first component (G) of the Bayesian network gives a set of independence conditions between the variables. Formally, G consists of a vertex set, $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$, and $V_i \in \boldsymbol{V}$ corresponds to a random variable X_i; $\boldsymbol{E} = \{e_1, \ldots, e_m\}$ is the edge set and $e_i = (v_x, v_y) \in E$ is a dependence between v_x and v_y. The second component, $\boldsymbol{\theta} = \theta_1, \ldots, \theta_n$ is the parameter set storing the conditional joint probability distribution over $\boldsymbol{X}$, where $\theta_i = \theta(X_i|Pa(X_i))$ is the conditional probability distribution of X_i given all the parents of X_i (denoted by $P(X_i|Pa(X_i))$. These two components specify a *unique* distribution over $X_1, \ldots, X_n$. The uniqueness can be proved [168] using the following result:

Theorem 2.4.1. ***Chain rule for Bayesian Networks.*** *The independence assumptions derived from* ***Markov****(G) are satisfied by a distribution* $P(X_1, \ldots, X_n)$ *if and only if* P *can be written as*

$$P\left(X_1, \ldots, X_n\right) = \prod_{i=1}^{n} P\left(X_i | Pa\left(X_i\right)\right) \tag{2.16}$$

where $Pa(X_i)$ *is the parent set of gene* X_i *in G.*

An example of a BN is shown in Figure 2.16. The joint probability distribution implied by the graph is shown in Equation 2.17.

$$P(A, B, C, D, E) = P(A) * P(B) * P(C|A) * P(D|A, B) * P(E|D) \tag{2.17}$$

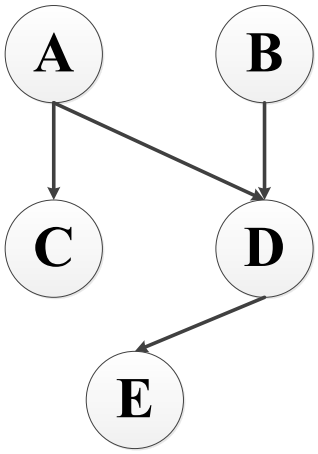

Figure 2.16: A Bayesian Network.

2.4.1 Equivalence Classes of Bayesian Networks

More than one graph can imply exactly the same set of independence relations. For example, consider the graphs $X \leftarrow Y$ and $X \rightarrow Y$. These graphs have the same set of independence relations. This types of graphs are called equivalent graphs. Pearl and Verma [169] showed that we can characterize equivalence classes of graphs using a simple representation. We first define a useful sub-structure that plays a key role in the definition of graph equivalence.

Definition 2.6. ***V Structure.*** *A v-structure is an induced sub-graph of the form $X \leftarrow Y \rightarrow Z$ (fork v-structure) or $X \rightarrow Y \rightarrow Z$ (chain v-structure) or $X \rightarrow Y \leftarrow Z$ (collider v-structure) so that no edge exists between X and Z.*

Based on definition 2.6, we can now define equivalent graphs as follows:

Theorem 2.4.2. ***Graph Equivalence.*** *Two Bayesian network structures are equivalent if and only if they have the same underlying undirected graph (termed skeleton) and the same v-structures.*

Figure 2.17 shows the equivalence classes for a three variable skeleton $A - B - C$.

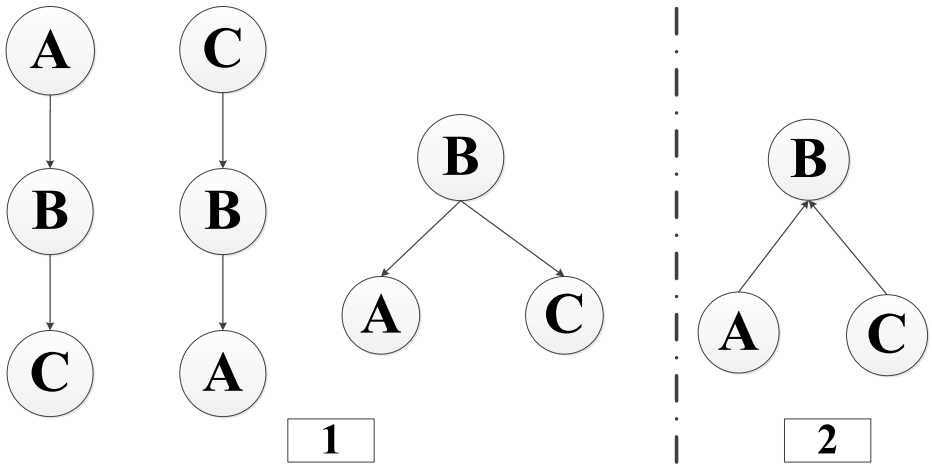

Figure 2.17: The two equivalent classes of the skeleton graph $A - B - C$

The notion of equivalence is crucial, since when we examine observations from a distribution, we cannot distinguish between equivalent graphs, under the common scenario of learning networks. This scenario is violated in two cases: First when we restrict the allowed networks to a certain structural family, for example trees. Second, when we use a type of CPDs that prefers a certain directionality in the connections. In both these cases we might have a preference of one equivalent network over another.

2.5 Dynamic Bayesian Network (DBN)

Considering ***X*** to be a set of attributes changing in a temporal process of T time slices, a DBN represents the joint probability distribution over the variables $\mathbf{X}[0] \bigcup \mathbf{X}[1] \bigcup \cdots \bigcup \mathbf{X}[T-1]$, where random variable $X_i[t]$ denotes the value of node X_i at time slice *t*, and $\mathbf{X}[t]$ denotes the set of variables $\{X_i[t] | 1 \leq i \leq n\}$, for $0 \leq t \leq T-1$ [77,248].

In case of static Bayesian networks, we assume that our data set is composed of independent samples from the generating distribution. Unless the process we are watching is totally memoryless, an observation at time t carries some information on observation at adjacent times. This is called the Markov property.

2.5.1 Markov Property

The (first-order) Markov property says that given the current observation $\boldsymbol{X}[t]$, the next observation $\boldsymbol{X}[t+1]$ is independent of past observations, $\boldsymbol{X}[0], \ldots, \boldsymbol{X}[t-1]$ (or more simply: the future is independent of the past given the present). Formally,

$$P\left(\mathbf{X}[t]|\mathbf{X}[t-1], \ldots, \mathbf{X}[0]\right) = P\left(\mathbf{X}[t]|\mathbf{X}[t-1]\right) \tag{2.18}$$

Similarly, we can define the $d-th$ order Markov Property as follows 2.19

$$P\left(\mathbf{X}[t]|\mathbf{X}[t-1], \ldots, \mathbf{X}[0]\right) = P\left(\mathbf{X}[t]|\mathbf{X}[t-1] \ldots \mathbf{X}[t-d]\right) \tag{2.19}$$

where $d \leq t$.

2.5.2 Stationarity

If the Markov assumption holds, the data likelihood can be decomposed as:

$$P(\boldsymbol{X}[1], \ldots, \boldsymbol{X}[T]) = P(\boldsymbol{X}[1]) \prod_{t=2}^{T} P(\boldsymbol{X}[t]|\boldsymbol{X}[t-1]) \tag{2.20}$$

Using this formulation, we still have to specify M sets of probability distributions, where M might be very large. We therefore usually make another simplifying assumption, that is, the probabilistic model is time invariant. Formally, the stationarity assumption says that $P(\mathbf{X}[t]|\mathbf{X}[t-1])$ is independent of t. In practice, all the DBN based GRN reconstruction techniques makes use of the above two assumptions.

A sample dynamic Bayesian network employing the above assumptions is shown in Figure 2.18. The DBN in the figure is called a third-order Markov DBN.

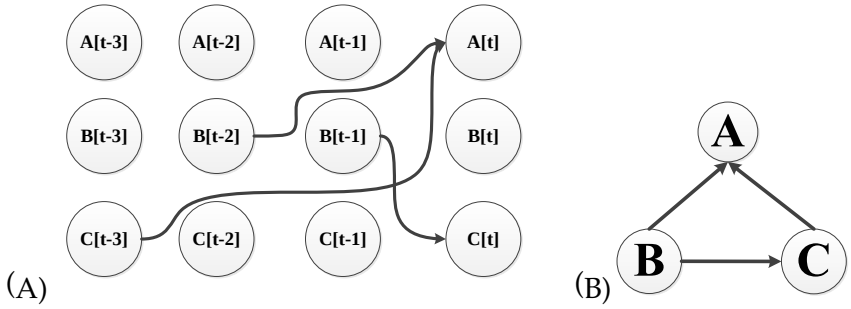

Figure 2.18: A 3rd-order Markov DBN. (A) Unrolled representation. (B) Rolled representation.

2.6 Advantages of Using Dynamic Bayesian Network Based Reconstruction Methods

Some advantages of using dynamic Bayesian network based reconstruction methods are:

1. One of the main advantages of dynamic Bayesian networks and overall Bayesian networks in general is the ability to factorize the graph [168]. That is, the value of each component directly depends on the values of a relatively small number of components. This greatly simplifies computation in various scenarios.

2. Signal transduction, gene expression and its regulation are stochastic processes [170, 180, 185]. Thus it is appropriate to use stochastic models like BNs/DBNs for GRN modeling.

3. Both intrinsic noise (due to stochastic events during gene expression; responsible for differences between identical reporters in the same cell), and extrensic noise [170, 180] (due to cellular heterogeneity; causes differences between identical reporters in different cells) can be better taken care of using DBN/BN based models.

4. Bayesian networks and dynamic Bayesian networks are based on solid statistical foundation, and computational algorithms to learn the structures and parameters of such networks are well understood. As a result, these models have been used successfully in many applications [43, 92, 134].

5. The idea of using BNs for GRN reconstruction can be thought of as a way to simplify the mathematical description of the biological system by replacing the coupled

differential equations by simple conditional probability distributions of a standard form such that the unknown parameters can be integrated out analytically. This results in a marginal likelihood of closed form that depends only on the structure of the regulatory network and avoids the over-fitting problem suffered by differential equation based models.

6. DBNs and BNs allow the combination of highly dissimilar types of data (i.e., numerical and categorical), converting them to a common probabilistic framework, without unnecessary simplification [104].

7. DBNs and BNs readily accommodate missing data, which is a common problem when designing microarray experiment based GRN reconstruction techniques [104, 176]. Also, hidden variables in a network are easier to handle use BN/DBN based techniques [176].

8. BNs and DBNs have the natural capability of weighing each information source according to its reliability [161].

9. BNs and DBNs are readily interpretable, unlike "black-box" predictors, as they represent conditional probability relationships among information sources [104].

10. BNs and DBNs can model causal interactions, which is particularly useful for GRN analysis [7, 19, 88, 168, 206, 235].

11. BNs and DBNs can naturally deal with the stochastic aspects of gene expression and the noisy measurements of DNA microarray and other data sources because of its firm statistical footing [176].

12. BNs and DBNs are able to handle a large number of variables with only a few samples [7, 21, 64, 194]. It is especially useful when learning gene networks, since gene networks are often plagued by the well known curse of dimentionality problem (large number of genes but only a few samples).

13. Bayesian networks are capable of estimating the confidence of different features in networks [76]. The absence of data often leads to a situation where many networks explain the data equally well. The confidence is useful for measuring whether a statistical feature of the network is likely to be true [76, 176].

14. Using information theoretic quantities, dynamic Bayesian networks can capture many types of relationships among genes: linear, non-linear, combinatorial, stochastic and so on. It remains unclear which types of relationships a gene regulatory system may pursue. The ability of Bayesian networks to grasp various types of relationships makes it appropriate for learning gene networks.

Using a good model for the reconstruction completes half of the task only. After a modeling technique is decided upon, we then have to use a learning strategy for the identification of regulatory interactions.

In the next section, we will discuss about learning the structure of Bayesian Networks. Albeit the discussion is mainly based on static Bayesian Networks, extending the reasoning to DBNs is very straightforward. For example, for representing dependencies, any form of CPD we use in a BN can also be used in a DBN. Inference is essentially similar to the BN case, as it is done on the unrolled network model. Parameter and structure learning are typically done on the unrolled representation [77], but other than that they are similar to what is done on static Bayesian networks.

2.7 Learning Bayesian Network Structure

Somewhat generalizing, there are two approaches for finding the structure of GRNs using DBNs. The first approach poses the learning task as a constraint satisfaction problem. In this approach, we try to estimate properties of conditional independence among the variables in the data. Usually this is done using a statistical hypothesis test, such as t-test or χ^2-test. We then build a network that exhibits the observed dependencies and independencies. The second approach poses the learning task as an optimization problem, and these are usually called *score and search* based approaches. We start by defining a statistically motivated score that describes the fitness of each possible structure to the observed data. The learners task is then to find a structure that maximizes the score. In general, this is an NP-hard problem [45,46], and thus we need to resort to heuristic methods. Although the constraint satisfaction approach is efficient, it is sensitive to failures in independence tests. Thus, the common opinion is that the optimization approach is a better tool for learning structure from data.

2.7.1 Constraint Based Learning

Constraint based learning methods usually use conditional independence tests. At first we describe three concepts that are vital for constraint based learning.

Assumption 2.7.1. ***Causal Sufficiency Assumption.*** *The causal sufficiency assumption states that there are no unobserved variables in the domain that might explain the independencies that are observed in the data, or lack thereof. It is a crucial assumption for applications that need to determine the true underlying (causal) structure of the domain.*

Assumption 2.7.2. ***Causal Markov Assumption.*** *It expresses a minimum set of independence relations that exist between every node and its non-descendants, given a BN model. From these, and a set of axioms described in [168,215], one can produce the entire set of independence relations that are implied by that BN model.*

Definition 2.7. ***Faithfulness.*** *A BN graph G and a probability distribution P are faithful to one another if and only if every one and all independence relations valid in P are those entailed by the Markov assumption on G.*

Definition 2.8. ***d-separation.*** *Two sets of nodes $\mathbf{X}$ and $\mathbf{Y}$ are d-separated in Bayesian networks by a third set $\mathbf{Z}$ (excluding $\mathbf{X}$ and $\mathbf{Y}$) if and only if every path between $\mathbf{X}$ and $\mathbf{Y}$ is "blocked", where the term "blocked" means that there is an intermediate variable V (distinct from $\mathbf{X}$ and $\mathbf{Y}$) such that:*

- *The connection through V is "tail-to-tail" or "tail-to-head" and V is instantiated, or*
- *The connection through V is "head-to-head" and neither V nor any of Vs descendants have received evidence.*

The graph patterns of "tail-to-tail", "tail-to-head" and "head-to-head" are shown in Figure 2.19. The minimal set of nodes which d-separates node A from all other nodes is A's Markov blanket (MB). The Markov blanket $MB(X)$ of node X in a Bayesian network is the set of nodes composed of A's parents, its children, and its children's parents.

The SGS (Sprites-Glymour-Scheines) algorithm [215] is one of the most studied algorithms using constraint based tests. In this algorithm, the existence of an edge between two variables, is tested using a number of conditional independence tests. Each of these conditions is a subset of universal subset U. If Faithfulness holds and there exists an edge,

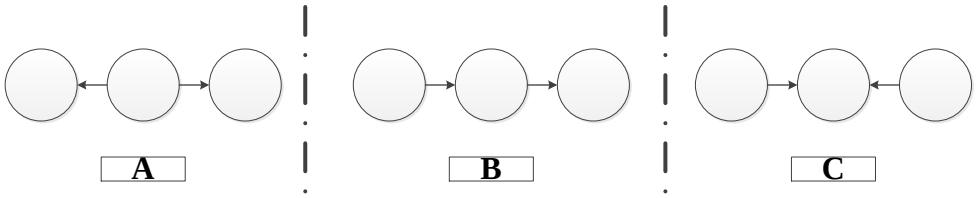

Figure 2.19: The three patterns related to d-separation. (A) tail-to-tail, (B) tail-to-head, (C) head-to-head.

then all these independence tests should be false. If there is no edge, then there must exist a subset d-separating them. Assuming that there is no direct edge between nodes X and Y in the true model, one such subset is the set of parents of one of the nodes. By trying all possible subsets of U, the SGS algorithm can make a conclusion on the existence of an edge between every pair of variables in the domain. After the undirected connectivity is determined, SGS attempts to determine the directionality of these edges. This is done by examining triples of variables X, Y, and Z, such that there is no subset that includes Z, can d-separate X and Y. This is repeated for all such triples, and is followed by verification of acyclic behavior of the graph.

Other algorithms exist in the literature that do not make use of independence tests but take into account d-separation in order to discover structure from data. Cheng *et al.* [43], for example, used mutual information instead of conditional independence tests. The algorithm requires the ordering of the variables to be given in advance. PC [215] is another constrained based algorithm, which orders CI tests to improve efficiency.

2.7.2 The Score and Search Paradigm

The algorithms based on a scoring function attempt to find a graph that maximizes the selected score, which is usually defined as a measure of fitness between the graph and the data. All of them use the scoring function in combination with a search method in order to measure the goodness of each explored structure from the space of feasible solutions. Different learning algorithms are obtained depending on the search procedure used, as well as on the definitions of the scoring function and the search space.

2.7.3 Scoring Techniques

Several scoring functions for learning Bayesian networks have been proposed in the literature. It is common to classify scoring functions into two main categories: Bayesian and information-theoretic. In general, for efficiency purposes, these scores need to decompose over the network structure. The decomposability property allows for efficient learning algorithms based on local search methods. Moreover, when the learning algorithm searches in the space of equivalence classes of network structures, scoring functions must also be score equivalent, that is, equivalent networks must score the same.

Bayesian Scoring Functions

The general idea of Bayesian scoring functions is to compute the posterior probability distribution, starting from a prior probability distribution on the possible networks, conditioned on the data D, that is, $P(G|D)$. The best network is the one that maximizes the posterior probability. Since the term $P(D)$ is the same for all possible networks, in practice, for comparative purposes, computing $P(G, D)$ is sufficient. Moreover, as it is easier to work in the logarithmic space, the scoring functions use the value $log(P(G, D))$ instead of $P(G, D)$.

Before moving on to the formulation of the scoring functions, some notations will be introduced: the number of states of the variable X_i is r_i; the number of possible configurations of the parent set $Pa(X_i)$ of X_i is q_i; obviously, $q_i = \prod_{X_j \in Pa(X_i)} r_j$; w_{ij}, $j = 1, \ldots, q_i$, represents a configuration of $Pa(X_i)$; N_{ijk} is the number of instances in the data set D where the variable X_i takes the value x_{ik} and the set of variables $Pa(X_i)$ take the value w_{ij}; N_{ij} is the number of instances in the data set where the variables in $Pa(X_i)$ take their j-th configuration w_{ij} (i.e., $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$). Similarly, N_{ik} is the number of instances in D where the variable X_i takes its k-th value x_{ik}, and therefore $N_{ik} = \sum_{j=1}^{q_i} N_{ijk}$; the total number of samples in D is N.

Heckerman *et al.* [93] proposed the Bayesian Dirichlet (BD) score by making four assumptions on $P(G, D)$. The first one assumes that data D is exchangeable, that is, if an instance of the data is exchanged with another instance, the exchanged data has the same probability as the original one. The second assumption assumes that parameters θ_{ij}

have a Dirichlet distribution. This hypothesis is convenient because the Dirichlet distribution is closed under multinomial sampling, that is, if the prior distribution is Dirichlet, the posterior distribution, given a multinomial sample, is also Dirichlet. The third hypothesis imposes that the parameters associated with each variable in the network are independent, and, moreover, the parameters associated with each instance of the parents of a variable are also independent. Finally, the fourth assumption states that the density for the parameters θ_{ij} depends only on X_i and its parents, that is, on the local structure of X_i.

Based on the four assumptions, Heckerman *et al.* [93] showed the following result:

Theorem 2.7.1. *Under assumptions 1 to 4, the joint probability P(G, D) can be expressed by the following formula:*

$$P(G,D) = P(G) \times \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})} \right) \tag{2.21}$$

The theorem above induces the Bayesian Dirichlet (BD) score defined as:

$$BD(G,D) = log\left(P(G)\right) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(log\left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})} \right) + \sum_{k=1}^{r_i} log\left(\frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})} \right) \right) \tag{2.22}$$

where the values η_{ijk} are the hyperparameters for the Dirichlet prior distributions of the parameters given the network structure, and $\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk}$, and $\Gamma(\cdot)$ is the Gamma function.

Another Bayesian scoring functions, called K2, was proposed by Cooper and Herskovits [54]. It also relies on several assumptions (multinomiality, lack of missing values, parameter independence, parameter modularity, uniformity of the prior distribution of the parameters given the network structure), and can be expressed as follows:

$$K2(G,D) = log\left(P(G)\right) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left(log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \right) + \sum_{k=1}^{r_i} log\left(N_{ijk}!\right) \right) \tag{2.23}$$

From the K2 score, we can readily see that it is a particular case of BD, where the uninformative assignments are done for the hyperparameters ($\eta_{ijk} = 1$). In fact, as Heckerman *et al.* mentioned, specifying all η_{ijk} for all i, j and k is formidable, making the BD score unusable in practice. However, by considering the additional assumption of likelihood equivalence [93], it is possible to specify the hyperparameters relatively easily. We first define the notion of likelihood equivalence:

Given a Bayesian network G, the data D can be seen as a multinomial sample of the joint space $\boldsymbol{B}$ with parameters $\boldsymbol{\theta_B} = \{\theta_{x_1,\ldots,x_n}\}$, $x_i = 1, \ldots r_i, i = 1 \ldots n$, where $\theta_{x_1,\ldots,x_n} = \prod_{i=1}^{n} \theta_{x_i|Pa(x_i)}$.

Assumption 2.7.3. ***Likelihood Equivalence.*** *Given two directed acyclic graphs, G and G' such that $P(G) > 0$ and $P(G') > 0$, if G and G' are equivalent then $P(\boldsymbol{\theta_B}|G) = P\left(\boldsymbol{\theta_B}|G'\right)$.*

Under the likelihood equivalence assumption, it follows that for equivalent DAGs G and G' we have $P(D|G) = P(D|G')$, that is, the data D does not help to discriminate equivalent DAGs. The result is a scoring function called BDe score, whose expression is identical to the BD one, but the hyperparameters now can be computed using the following formula:

$$\eta_{ijk} = \eta \times P\left(x_{ik}, w_{ij}|G_0\right) \tag{2.24}$$

where $P(\cdot|G_0)$ represents a probability distribution associated with a prior Bayesian network G_0 and η is a parameter representing the equivalent sample size.

A particular case of BDe which is especially interesting appears when $P(x_{ik}, w_{ij}|G_0) = \frac{1}{r_i q_i}$, that is, the prior network assigns a uniform probability to each configuration of $X_i \cup Pa(X_i)$. The resulting score is called BDeu, which was originally proposed by Buntine [32]. This score only depends on one parameter, the equivalent sample size η, and is expressed as follows:

$$BD_{eu}(G, D) = log\left(P(G)\right) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(log\left(\frac{\Gamma\left(\frac{\eta}{q_i}\right)}{\Gamma\left(N_{ij} + \frac{\eta}{q_i}\right)}\right) + \sum_{k=1}^{r_i} log\left(\frac{\Gamma\left(N_{ijk} + \frac{\eta}{r_i q_i}\right)}{\Gamma\left(\frac{\eta}{r_i q_i}\right)}\right)\right) \tag{2.25}$$

Regarding the term $log(P(G))$ which appears in all the previous expressions, it is quite common to assume a uniform distribution (except if we really have information about

the greater desirability of certain structures) so that it becomes a constant and can be removed.

Information Theory Based Scoring Functions

Information theory based scoring functions represent another option for measuring the degree of fitness of a DAG to a data set and are based on codification and information theory concepts. Coding attempts to reduce the number of elements which are necessary to represent a message (depending on its probability). Frequent messages will therefore have shorter codes whereas larger codes will be assigned to the less frequent messages. Shannon's source coding theorem [203] (or noiseless coding theorem) establishes the limits to possible data compression. There are several optimal codes that asymptotically achieve Shannon's limit, such as the Fano-Shannon code and the Huffman code. To construct these codes one requires as input, a probability distribution over the data, which can be derived from a Bayesian network. So, given data D, one can score a Bayesian network graph G by the size of an optimal code, induced by the distribution of G, when encoding D. This value is the information content of D by G and is given by

$$L(D|G) = -\sum_{i=1}^{n}\sum_{j=1}^{q_i} N_{ij} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N_{ij}} log\left(\theta_{ijk}\right) \tag{2.26}$$

Using Gibbs inequality, the value θ_{ijk} for which Equation 2.26 is minimized is given by:

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}} \tag{2.27}$$

This implies that the Bayesian network that induces a code that compresses D the most is precisely the Bayesian network that maximizes the probability of observing D. By applying a logarithm to the likelihood of D given G, we obtain $log(P_G(D)) = L(D|G)$ that is commonly called the log-likelihood of D given G. Observing that maximizing the log-likelihood is equivalent to minimizing the information content of D by G, we can define the log-likelihood (LL) score [28] in the following way:

$$LL(G|D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\left(\frac{N_{ijk}}{N_{ij}}\right) \tag{2.28}$$

The LL score tends to favor complete network structures and it does not provide an useful representation of the independence assumptions of the learned network. This phenomenon of overfitting is usually avoided in two different ways. First, by limiting the number of parents per network variable. Second, by using some systematic penalization factor over the LL score.

The minimum description length (MDL) scoring function [126, 224] takes the second approach to prevent overfitting discussed above, preferring simple Bayesian networks over complex ones. The minimum description length principle (MDL) selects the coding that requires minimum length to represent the messages. Another more general formulation of the same idea establishes that in order to represent a data set with one model from a specific type, the best model is the one that minimizes the sum of the description length of the model and the description length of the data given the model. Complex models usually require greater description lengths but reduce the description length of the data given the model (they are more accurate). On the other hand, simple models require shorter description lengths but the description length of the data given the model increases. The minimum description length principle establishes an appropriate trade-off between complexity and precision. In our case, the data set to be represented is D and the selected class of models are Bayesian networks. Therefore, the description length includes the length required to represent the network plus the length necessary to represent the data given the network [27, 60, 126, 224]. In order to represent the network, we must store its probability values, and this requires a length which is proportional to the number of free parameters of the factorized joint probability distribution. This number, called network complexity and denoted as $C(G)$, is:

$$C\left(G\right) = \sum_{i=1}^{n} \left(r_i - 1\right) q_i \tag{2.29}$$

The usual proportionality factor is $\frac{1}{2}log(N)$ [181]. Therefore, the description length of the network is $\frac{1}{2}C(G)log(N)$. Regarding the description of the data given the model, by using Huffmann codes its length turns out to be the logarithm of the likelihood function of the data with respect to the network, i.e., the negative of the log-likelihood. This value is minimum for a fixed network structure when the network parameters are estimated from the data set itself by using maximum likelihood. Therefore, by changing the signs

to deal with a maximization problem, we get the MDL scoring function as follows:

$$MDL\,(G,D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\left(\frac{N_{ijk}}{N_{ij}}\right) - \frac{1}{2}C(G)log(N) \tag{2.30}$$

Another way of measuring the quality of a Bayesian network is to use measures based on information theory. The basic idea is to select the network structure that best fits the data, penalized by the number of parameters which are necessary to specify the joint distribution. This leads to a generalization of the MDL scoring function:

$$GMDL(G,D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} log\left(\frac{N_{ijk}}{N_{ij}}\right) - C(G)f(N) \tag{2.31}$$

where $f(N)$ is a non-negative penalization function. Based on this, we can define several scoring functions:

- If $f(N) = 1$, we have the Akaike Information Criterion (AIC) scoring function [2].
- If $f(N) = 0$, we have the log likelihood score.
- If $f(N) = \frac{1}{2}log(N)$, we have the Bayesian Information Criterion (BIC), which coincides with the MDL score. However, this quantity is based on the Schwarz Information Criterion [197].

As far as the search is concerned, although the most frequently used are local search methods [32, 60, 61, 93, 154], due to the exponentially large size of the search space, there is a growing interest in other heuristic search methods such as simulated annealing [93, 255], tabu search [1, 28], branch and bound [227], genetic algorithms and evolutionary programming [127,153], Markov chain Monte Carlo [87,119], and ant colony optimization [57, 62].

However, due to the complexity in learning GRNs, simple and straightforward techniques employing the AIC/BIC scores or simple constraint based learning schemes usually do not obtain good results. Therefore, an ensemble of scoring/constraint based learning techniques is often used in the recent literature. Ram *et al.* [176] proposed such an approach that introduces a causal modeling framework using genetic algorithms. The framework requires decomposing a GRN into sub models which are the Markov blanket

graphs of each node of the network. To each of the MBs of the network, following steps are then sequentially applied.

1. Gene Expression Matrix calculation. Using the dataset, obtain a matrix E corresponding to the set of genes that are affected by gene X (i.e. MB of gene X).

2. Finding causal relations. The causal relationships between genes are defined as gene X affecting gene Y either directly or indirectly. This step thus create *n* binary causal relations.

3. Adjacency matrix calculation. The adjacency matrix (of size n X n where n is the number of nodes in the MB) is based directly on the binary relation R. The entries in the matrix get a value of 1, -1 and 0, if the regulation $i \rightarrow j$ is positive, negative and nonexistent, respectively.

4. Skeleton matrix calculation. A skeleton matrix S is developed from the adjacency matrix A to include both the direct and indirect effects observed in the MB. While matrix element $A(i, j)$ represents only the direct relationship, the corresponding element of skeleton matrix S also includes the indirect causal relationship between genes corresponding to X and Y.

5. From skeleton matrix S, the direct and indirect effects are respectively converted as conditional dependence (CD) and conditional independence (CI) constraints. Zero order constraints are obtained from the direct interactions while higher order constraints are obtained from the indirect interactions via a condition set.

6. Reduced Constraint set calculation. Some tests are not necessary to be implemented and can be eliminated from the constraint set. In this step, these reductions in statistical tests is done.

7. Constraints Evaluation. The consistency of the constraints with respect to data is evaluated in this step. Statistical significance test, namely F-test is conducted to check if the correlation coefficients differ significantly from zero value. These tests apply the Bonferroni-corrected p-value threshold to produce satisfactory correlations.

8. Fitness of the MB. The score of the overall putative network is obtained as a weighted linear combination of the consistency scores as shown below:

$$Fitness\_Score = \sum_i (w_1 \times Score_{1_i} + w_2 \times Score_{2_i} + w_3 \times Score_{3_i}) \tag{2.32}$$

Here w_1, w_2, w_3 are weights assigned to each of the three sub model scores such that $w_1 + w_2 + w_3 = 1$. As for $Score_{1_i}$, it scores the correctness of the structure based on partial correlation coefficients. The quantity $Score_{2_i}$ scores directions of causality by employing delay parameters in the partial correlation calculation. Finally, $Score_{3_i}$ scores the sign of regulation (positive/negative regulations).

The above steps are applied on each MB for a particular GRN, and a GA-based search strategy is employed to find possible GRN networks, the objective being finding the GRN that maximizes the fitness score (Equation 2.32). Subsequently, a constraint logic minimization (CLM) algorithm was proposed that improves the efficiency further. In addition to this PCC based scoring method employing CI tests, one can make use of information theoretic quantities to devise techniques that makes use of CI tests [157]. Finally, information theoretic CI tests can also be used in a scoring framework. Examples include [153] and [154].

Due to the vast popularity of BN based modeling techniques, there are many software packages for GRN inference that uses BN/DBN as the modeling technique. We discuss a few of the most popular and publicly available BN/DBN based software packages next.

BANJO

BANJO [255] is a popular gene network inference software that is based on the BN formalism and implements both static and dynamic Bayesian networks. It implements both greedy search and simulated annealing, and a BDe based scoring technique is used. From a high-level point of view, during searching, an initial "current" network (can be the empty network, or some other pre-selected network) is selected, and subsequently, the search process iterates through the following set of steps:

- Propose a new network that is to be considered. Often, the proposed network is dependent on the current network, and it represents a local change to it. This is called the "Proposer" module.
- Check the proposed network for cycles. This is called the "Cycle Checker" module, and it is optional.
- Compute the score of the proposed network using a pre-defined metric. Called the "Evaluator" module.
- Decide, possibly stochastically, whether to accept the proposed network (as the new current network). Called the "Decider" module.

A "Proposer" implements the part of the search algorithm that specifies what possible change or changes are to be considered at a single search iteration step. There are two proposer choices: 'RandomLocalMove', which does addition, deletion, or reversal of an edge in the current network, selected at random, and 'AllLocalMoves', which considers all changes arising from a single addition, deletion, or reversal of an edge in the current network. The task of the cycle checker is to examine (using DFS) whether each proposed network contains a cycle. If it does, the proposed change is discarded, and the search goes back to the "Proposer" to request another possible network change. If the proposed network does not contain a cycle, then the next step in the search is the score computation performed by an "Evaluator", which computes the parameters of the conditional probability density distribution to find an overall network score, using the BDe metric. The "Decider" uses two strategies: a greedy approach, where a network is accepted if and only if its score is better than or equal to that of the current network (in the case of 'AllLocalMoves' option in "Proposer" module, the best local move is considered), and a Metropolis decider, which uses a Metropolis-Hastings stochastic decision mechanism, where any network with a higher score is accepted, and any with a lower score is accepted with a probability based on a system parameter known as the 'temperature'. BANJO outputs a signed directed graph indicating regulation among genes. BANJO can analyse both steady-state and time series data. In the case of steady-state data, BANJO is not able to infer networks involving cycles (e.g., feedback loops).

BNFinder

BNFinder is a software package, which allows Bayesian network reconstruction from experimental data. It supports dynamic Bayesian networks and, if the variables are partially ordered, also static Bayesian networks. The BNFinder program is based on a polynomial-time algorithm for learning an optimal Bayesian network structure [66]. It works under the following four assumptions:

Assumption 2.7.4. ***Acyclicity.*** *There is no need to examine the acyclicity of the graph.*

Assumption 2.7.5. ***Additivity.*** $S(G,D) = \sum_{i=1}^{n} s\left(X_i, Pa(X_i), D\big|_{X_i \cup Pa(X_i)}\right)$ *where* $D\big|_{X_i \cup Pa(X_i)}$ *denotes the restriction of* D *to the values of the members of* $X_i \cup Pa(X_i)$.

To simplify notation, we write $s(Pa(X_i))$ for $s\left(X_i, Pa(X_i), D\big|_{X_i \cup Pa(X_i))}\right)$.

Assumption 2.7.6. ***Splitting.*** $s(Pa(X_i)) = g(Pa(X_i)) + d(Pa(X_i))$ *for some non-negative functions* g, d *satisfying* $Pa(X_i) \subseteq Pa'(X_i) \Rightarrow g(Pa(X_i)) \leq g(Pa'(X_i))$.

Assumption 2.7.7. ***Uniformity.*** $|Pa(X_i)| = |Pa'(X_i)| \Rightarrow g(Pa(X_i)) = g(Pa'(X_i))$.

Assumption 2.7.4 is valid for DBN in general. Assumption 2.7.5 states that the employed scoring function decomposes over the variables. Together with assumption 2.7.4, this assumption allows us to compute the parent set of each variable independently. Assumption 2.7.6 requires the scoring function to decompose into two components: d, evaluating the accuracy of representing the distribution underlying the data by the network, and g, measuring its complexity. Finally, g is required to be a monotonically non-decreasing function in the cardinality of $Pa(X_i)$. The BNFinder algorithm uses the BDe and MDL scoring function, which satisfy the above constraints.

As stated, BNFinder can learn either dynamic Bayesian networks (from time series data) or static BNs (from independent experiment data). In the second case it is necessary to specify constraints on the network's structure, forcing its acyclicity. A special treatment is required for experiments, in which the values of some variables were perturbed (e.g. knockout experiments). Since perturbations change the structure of interactions, learning procedures have to use data selectively. In perturbation experiment datasets, for scoring sets of parents of a variable v, it takes into account only the experiments where v was not perturbed. Finally, a prior distribution on the network structure may be specified through

assigning weights to potential variable interactions. However, in this case the size of the regulator sets of each variable may be bounded to a given number and the spaces of possible conditional probability distributions of selected variables may be restricted to noisy-and or noisy-or distributions.

Bayes Net Toolbox

The Bayes Net Toolbox (BNT) [159] is an open source Matlab package for directed graphical models. BNT supports many kinds of nodes (probability distributions), exact and approximate inference, parameter and structure learning and static as well as dynamic Bayesian models. It can also handle missing data and hidden variables (partial observability). In BNT, a Bayes net is represented as a structure, which contains the graph, as well as the CPDs (conditional probability distributions). The graph is represented as an adjacency matrix, whereas the CPDs are represented as a list of objects. BNT supports both structure and parameter learning. The parameter estimation routines in BNT supports computing both a full Bayesian posterior over the parameters, and point estimates (maximum likelihood or maximum a posteriori). The structure learning routine in BNT can similarly be classified into four categories, depending on full/partial observability and calculation of point or Bayes estimates. For fully observed data and point estimates, it supports K2 and IC/PC algorithms. For Bayesian estimates it uses MCMC techniques.

2.8 Summary

In this chapter, we briefly discussed the biological preliminaries, describing the cell and its relation to gene regulatory networks. This was followed by elaborating mathematical quantities used for the reconstruction of these networks. Next, we reviewed various reconstruction techniques, ranging from simple clustering based approaches to parameter-heavy differential equation based networks, and also Bayesian networks. Further, the two different paradigms of learning the structure of Bayesian networks were also discussed. From the literature reviewed, it is observed that compared to various simple network reconstruction techniques, Bayesian networks are much more robust, and at the same time achieve very good performance. Moreover, BN models work with much less parameters compared with the nonlinear differential equation based models. In the next

chapter, we employ MI/CMI based CI tests for developing an algorithm that leverages the benefits of DBNs and also has the ability of detecting statistically significant interactions among genes.

Chapter 3

Information Theoretic Bayesian Approach for Genetic Network Reconstruction

3.1 Introduction

In the previous chapter, we reviewed different methods used for the reconstruction of gene regulatory networks, and also discussed the associated challenges. Amongst the various methods proposed to overcome the difficulties associated with GRN modeling and inference mechanism, it was observed that the Bayesian network (BN) framework is widely used [66, 255, 258]. As discussed, the firm statistical basis of Bayesian networks (BNs) allows them to deal with the stochastic aspects of gene expression and the noisy measurements of microarray data in a natural way [76, 175]. Moreover, hidden variables in a network and missing values in the gene expression data are easy to handle using this formalism.

Although effective in dealing with noise, incompleteness and stochastic aspects of gene regulation, BNs fail to consider the temporal dynamic aspects that are an important part of gene regulatory network modeling [37]. Dynamic Bayesian networks (DBN) [76, 168], an extension of Bayesian networks, can effectively deal with the temporal aspects of such regulatory networks. Along with other advantages, the BN part of a DBN helps, to some extent, to handle the problem arising due to the curse of dimensionality.

By incorporating network dynamics, we can efficiently handle the temporal aspects of GRN, and also allow occurrence of feedback loops in the network, which are an integral part of regulatory networks [20, 122, 247]. Alongside, the temporal regulation relationship implies that the directions of regulations between the genes can be readily inferred. However, for basic Bayesian networks, this computation can be very time consuming as large number of conditional independence tests are required. Thus, dynamic Bayesian networks can be seen as a promising trade-off between over-simplicity and loss of computational tractability.

In learning genetic networks, partial correlation coefficients (PCC) have been commonly used to assess genetic interactions [13, 68, 176, 177, 256]. However, these coefficients quantify linear dependencies. Since the genetic interactions are usually nonlinear, PCCs are thus unsuitable in reconstructing GRNs. In this research, we therefore propose to use MI and CMI instead, because MI and CMI are both nonlinear measures and more suitable to represent the nonlinear genetic interactions. Further, an important advantage of using MI and CMI is that they are zero if and only if the measurements on any two systems A and B are statistically independent. This puts them in an advantageous situation compared to the other commonly used measures, such as partial correlations. A vanishing MI or CMI does imply that two variables are independent, while for the Pearson correlation this does not hold true. Thus, the mutual information can be interpreted as a generalized measure of correlation, analogous to partial correlation but sensitive to any functional relationship, not just linear dependencies.

Literature [37, 75, 76, 171, 248] reveals a limited application of MI and CMI based learning techniques using Bayesian networks based GRN modeling. To date, in relevance networks (not BN/DBN), MI and CMI have been used essentially as threshold values without involving any modeling. When applied to BN and DBN, they have merely been used either for learning three node substructures (in constraint based learning methods [15, 42]) or for pairwise association measures (using threshold based techniques [248]). As we show in this chapter, MI and CMI can be applied in a novel manner as powerful regulatory association measures within DBN frameworks, for performing statistical significance tests exploiting the decomposition property of MI. They are also suitable for identification of the strength of regulatory interactions to allow removal of spurious or indirect network arcs.

In this chapter, with a view to using the solid mathematical underpinnings of Bayesian networks, and also for taking advantage of the temporal dynamics embedded within the time series microarray datasets, we propose a novel information theory based DBN design for gene regulatory network reconstruction. The approach employs mutual information based conditional independence tests, to systematically search for potential gene regulators, and builds a network connecting genes to its regulators. With in-built statistical significance testing, it can capture regulatory relations with good accuracy. The method is flexible, computationally fast and allows a-priori incorporation of biological domain knowledge. Due to the use of MI/CMI based CI tests, it can assess nonlinear regulatory interactions among genes. The proposed approach is validated by carrying out experiments using both synthetic and real-life data, and comparison with other methods shows the effectiveness of our approach.

The rest of the chapter is organized as follows. Section 3.2 explains the proposed methodology and its formalization as an algorithm. Section 3.3 discusses the synthetic and real-life networks used for validating our approach and also its comparison with two other techniques. Finally, Section 3.6 concludes with some observations and remarks, and the motivations behind the developments in the next chapter.

3.2 BITGRN: Bayesian Information Theoretic Gene Regulatory Network Reconstruction Algorithm

Regulatory relationship can be assessed by using Mutual Information (MI) based CI tests.

Let us consider the mutual information between a gene X_i and its parents, $Pa(X_i)$ in a Directed Acyclic Graph (DAG) $\boldsymbol{G}$. According to the decomposition property of MI (equation 3.1), which asserts that in a DBN, if $Pa(X_i)$ is the parent set of a node X_i ($X_{ik} \in Pa(X_i), k = 1, \ldots s_i$), and the cardinality of the set is s_i, the following identity holds [60]:

$$MI(X_i, Pa\,(X_i)) = MI\,(X_i, X_{i1}) + \sum_{j=2}^{s_i} MI\left(X_i, X_{ij} | \left\{X_{i1}, \cdots, X_{i(j-1))}\right\}\right) \tag{3.1}$$

where X_{ij} denotes the j-th parent of gene X_i. The elements in the decomposition on the right side can be interpreted as follows. We find the best parent for gene X_i (first term in

the right side of equation 3.1) by calculating its MI with all the other potential parent candidates and select the gene X_j for which $MI(X_i, X_j)$ is maximum. This approach helps us in discarding potential indirect regulators (parents) of gene X_i. This is because of the Data Processing Inequality (DPI) [55] which states that, if nodes X and Z are connected through an intermediate node Y (see Figure 3.1), then,

$$MI(X, Z) = \min\{MI(Y, X), MI(Z, Y)\} \tag{3.2}$$

i.e., the lowest MI value corresponds to either the indirect relationship or another weak

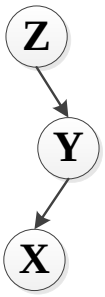

Figure 3.1: Illustration of the Data Processing Inequality

regulatory relationship. Hence, given two candidate parents Y and Z for X, say, the MI between X and Y is higher; then Y is considered the better candidate to be the parent of X. Assessing the MI between a node X_i and the candidate parents in this way helps to assess how much additional information we get about X_i by adding a candidate as a parent of this gene, enabling us to identify the best candidate parent.

After the first parent is added and if X_i has more than one parent, there will still be (entropic) uncertainties about this gene. So we add a second parent. While adding the second parent, rather than computing the pairwise MI between X_i and all the second parent candidates, we calculate the mutual information between X_i and a candidate parent conditional on the current parent set of X_i, i.e., we calculate $MI(X_i[t+1], X_{CP_k(X_i)}[t]|Pa_c(X_i[t]))$, where $CP_k(X_i)$ represents an element in the current candidate parent set of the node X_i and $Pa_c(X_i)$ represents the current parent set of X_i. We continue the insertion of arcs in this manner until the last parent has been included. We stop the insertion of any additional arcs if each of the remaining variables in the current candidate parent set, $CP_k(X_i)$, does not contribute an appreciable amount of additional information about X_i. The question as to how to determine whether the value of the CMI

represents some statistically significant amount of information is done by applying the following Theorem of Kullback 3.2.1 [123]:

Theorem 3.2.1. *Given a data set D with N samples, if the hypothesis that X and Y are conditionally independent given Z is true, then the statistic $2N \times MI(X, Y|Z)$ approximates to a chi-square distribution with $df = (r_X - 1)(r_Y - 1)r_Z$ degrees of freedom, where r_X, r_Y and r_Z represent the number of configurations (possible values) for the sets of variables X, Y and Z respectively. If $Z = \varnothing$, the statistic $2N \times MI(X, Y)$ approximates to a chi-square distribution with $df = (r_X - 1)(r_Y - 1)$ degrees of freedom.*

Let us fix a confidence level α and determine the value $\chi(\alpha, df_{ik})$ such that $P(\chi^2(df_{ik}) \leq \chi(\alpha, df_{ik})) = \alpha$. This evaluation actually represents a statistical test of conditional independence [60]. The test would assert that these two variables are dependent, if

$$2N \times MI(X_i[t+1], X_{CP_k(X_i)}[t]|Pa_c(X_i[t])) \gg \chi(\alpha, df_{ik}) \tag{3.3}$$

The more dependent they are, the larger will be the difference.

On the other hand, if

$$2N \times MI(X_i[t+1], X_{CP_k(X_i)}[t]|Pa_c(X_i[t])) < \chi(\alpha, df_{ik}) \tag{3.4}$$

it would mean that they are independent. Hence, when the maximum CMI value conditioned on the current parent set fails this test, we stop adding parents to this gene.

This approach is summarized in Table 3.1, as the Bayesian Information Theory based Gene Regulatory Network (BITGRN) reconstruction algorithm. Along with incorporating the aforementioned concepts, the approach also addresses another important issue: handling weak regulations. It is well known that gene regulations are not equal in strength and some of the regulations can be quite weak. While adding parents using the above procedure, if we have added some parents to a gene, and if the remaining uncertainty is due to some weakly regulating parent, the CMI (between the gene and the weakly regulating parent, given the already added parents) might not remain statistically significant. It is to be noted that this can also happen due to noise; however, since the expression profiles are discretized, the effect of noise would be appreciably low. Hence we can safely consider that if the CMI value is greater than (β x significance threshold), the

1. **for** each gene $X_i \in X_{1,\ldots,n}$ **do**
2. $Pa_c(X_i) \leftarrow \varnothing$
▶ Initial empty parent set
3. $maxMI_{ik} \leftarrow 0$
4. $attemptFailed \leftarrow false$
5. **while** $((|Pa_c(X_i)| \leq maxParents$ & $(attemptFailed = false))$ **do**
▶ *maxParents* is the maximum number of parents that is allowed per gene
6. $CP_k(X_i) \leftarrow findPromisingParentSet(X_i, Pa_c(X_i))$
7. **for** each $X_k \in CP_k(X_i)$ **do**
8. $mi_{ik} \leftarrow 2N \times MI(X_k[t], X_i[t+1]|Pa_c(X_i[t]))$
9. **if** $(mi_{ik} \geq maxMI_{ik})$ **then**
10. $maxMI_{ik} \leftarrow mi_{ik}$
11. $maxK \leftarrow X_k$
12. **end if**
13. **end for**
14. $j \leftarrow |Pa_c(X_i)| + 1$
15. $l_{ik} \leftarrow \chi(\alpha, df_{ik})$
▶ α is the confidence level
where $df_{ik} = \begin{cases} (r_i - 1)(r_{ik} - 1) \prod_{m=1}^{j-1} r_{im}, & j \geq 2 \\ (r_i - 1)(r_{ik} - 1), & j = 1 \end{cases}$
and $r_{im} = config(X_m)$, such that $X_m \in Pa_c(X_i)$
▶ $config(X_m)$ is the number of possible states/values that gene X_m can take
16. **if** $(((Pa_c(X_i) \neq \varnothing)$ & $(maxMI_{ik} \geq \beta * l_{ik}))$ | $(maxMI_{ik} \geq l_{ik}))$ **then**
17. $Pa_c(X_i) \leftarrow Pa_c(X_i) \cup maxK$
18. **end if**
19. **if** $(maxMI_{ik} < l_{ik})$ **then**
20. $attemptFailed \leftarrow true$
21. **end if**
22. **end while**
23. **end for**

Table 3.1: Algorithm BITGRN

candidate gene can be treated as a parent. In essence, the β value qualifies the amount of noise present in the data, and typically has a value in the range $[0.2, 0.4]$.

Referring to Table 3.1, the algorithm iterates for each gene (lines 1-23), trying to add parents for the current gene under consideration. Lines 2-4 initialize variables, and lines 5-22 iteratively add parents to the current gene under consideration. In line 6, probable parents for the current gene are selected using the *findPromisingParentSet* routine. This routine adds the capability of incorporating any additional information that we have regarding that gene. For example, if we have a-priori biological information about a particular gene violating specified preconditions for becoming a regulator of another gene, this can be included in the routine to exclude that gene as a potential parent (e.g., most yeast networks do not have auto regulations[1], although they may contain feedback loops).

[1] a gene regulating itself

Lines 7-13 find the parent which provides the maximum amount of reduction of uncertainty about the gene under consideration. Lines 14-15 initialize related variables used for testing conditional independence relations, and finally, lines 16-21 conduct the conditional independence testing, setting required flags to indicate whether iterative addition of parents to this gene should continue further or not.

3.3 Salient Features of Experimental Setup

DNA microarray datasets are one of the primary tools which are used for the reconstruction of GRNs. DNA microarray experiments can be divided into two main types: static and time series. In static expression experiments, a snapshot of the expression of genes from a number of samples at a given instant in time is measured. For example, in a type of static microarray experiment where study of the mechanism of a particular disease is done, researchers measure and compare gene expression levels in tissue samples taken from individuals with and without the disease. In time series experiments, expression levels are measured in a single sample at a number of points in time. Compared to static experiments, time series microarray data has greater scope and application for gene regulatory network study, since gene expression itself is a temporal process. Most of the applications of time series microarray experiments can be classified into one of four broad categories [18]. The first among these is uncovering the dynamics behind various biological systems, e.g., the cell cycle or the circadian clock in mouse and humans [18, 167, 218], and also various aspects of the biological systems in yeast [49, 214]. Secondly, researchers can use time series experiments to determine genetic responses to various conditions of interest, such as gene knockouts [260], stress conditions [81], and drug administration [121, 219]. The third category is development related, where time series expression experiments can be used to study development at the molecular level, and to identify genes that control, or play a key role in different stages of development. Study of the nervous system development and stem cell differentiation [11, 102] are some important examples of processes that have been studied in this way. Finally, temporal microarray experiments can be used to shed light on disease progression by revealing the genetic changes underlying observable symptoms [11, 18, 121]. Thus, temporal expression data

clearly has the potential to generate a great deal of biological knowledge. Hence, in all the experiments that we report in this thesis, we will use time series microarray data.

Since our method assumes that data is discrete, in the simulation results described here, we use the Persist [150] algorithm to discretize the data into 3 levels (corresponding to normal condition, up regulation and down regulation). The algorithm is based on the Kullback-Leibler divergence between the marginal and the self-transition probability distributions of the discretization symbols[2]. Based on the argument that one discretization is better than another if the resulting states show more persisting behavior, the method uses the Kullback-Leibler divergence of the marginal and self-transition probability distributions of the symbols as the quality measure used for persistence. Persist achieves significantly higher accuracy than various existing methods (e.g., equal width and equal frequency histogram methods, k-means and HMM based methods) and is robust against noise [150].

We used a value of 0.90 for the confidence level (α). Based on the justifications given later at the end of this section, for the comparison of BITGRN with various other methods, we have used the beta value of 0.2 for biological and synthetic networks with noisy data, and a beta value of 0.4 for other networks (synthetic networks having a large number of time points). The *maxParents* parameter (i.e., the maximum number of parents a gene can have) was set to 4. For all the experiments related to synthetic network, we used 3 different datasets for each experiment and combined these 3 datasets using the procedure described in [222, 250].

We will briefly summarize the idea of dataset combination using Figure 3.2 as an illustration. In the figure, there are M datasets, and we are trying to asses whether gene A is a parent of gene B, given that we have already added gene C as a parent. In this case, since A and C are parent/parent candidate, they act as the antecedent, and thus the alignment needs to be done in a way so that data at a time point for the parents/parent candidates gets aligned with data for the child at the next time point. This process is applicable for all the M datasets (see the figure). The shaded boxes indicate the data points which will be used for calculating mutual information and conditional mutual information.

[2]The concept of symbolic time series analysis is built upon phase-space partitioning for encoding nonlinear system dynamics from observed time series data, followed by construction of a finite state space model from a symbol sequence [48].

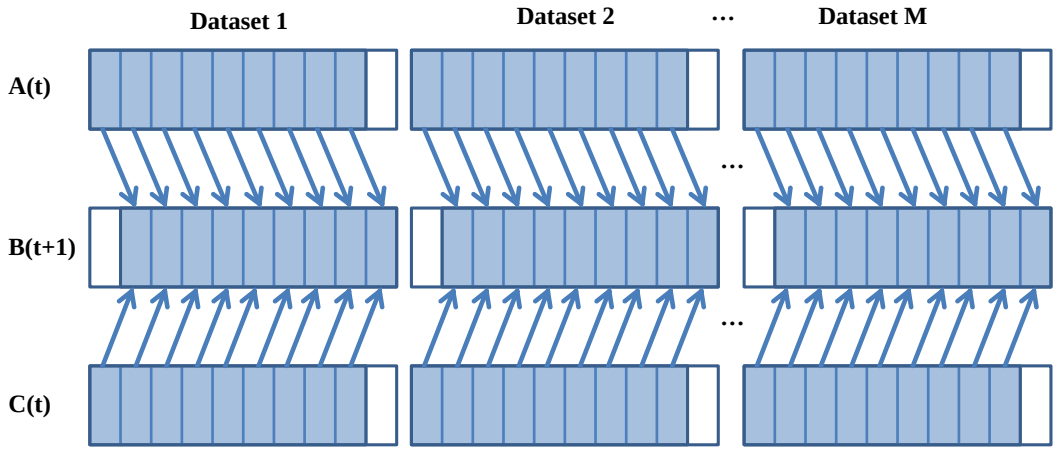

Figure 3.2: Illustration of how multiple sources of data can be combined with an emphasis on correct alignment

For the synthetic networks used in this study, we do not presume any prior knowledge; hence, the *findPromisingParentSet* routine allows self loops (i.e., auto-regulation) for these networks. However, for real-life networks auto regulations are not common. Hence, for the real-life networks that we consider in this chapter, when generating candidate parents for a gene under consideration, the *findPromisingParentSet* routine excludes the same gene, so that it can not be considered as a potential parent.

The evaluation of the proposed technique is carried out by both synthetic networks and real-life biological networks of *Saccharomyces cerevisiae* (yeast), and *E. coli*. The overall accuracy of the inference method and correctness of the modeling approach is evaluated by the four well known performance measures, namely *Se*, *Sp*, *Pr* and *F*, defined next. The terms, TP, FP, TN and FN, used in the following expressions respectively mean the number of true positives, number of false positives, number of true negatives and number of false negatives.

1. ***Sensitivity(Se):*** It measures the proportion of true connections which are correctly inferred by the algorithm. It is defined as follows.

$$Se = \frac{TP}{TP + FN} \tag{3.5}$$

2. ***Specificity (Sp):*** Specificity is defined as follows.

$$Sp = \frac{TN}{TN + FP} \tag{3.6}$$

3. ***Precision (Pr):*** Precision is proportional to the inferred connections which are correct. It is defined as follows:

$$Pr = \frac{TP}{TP + FP} \tag{3.7}$$

4. ***F-score (F):*** Biologically, a good reconstruction algorithm should infer as many correct arcs as possible, in addition to the criteria that most of the inferred arcs should be correct. The F-score measure is the harmonic mean of *Se* and *Pr* [261] and represents a compromise between these two objectives:

$$F = \frac{2\ Pr\ Se}{Pr + Se} \tag{3.8}$$

3.4 Experimental Results Using Synthetic Network

3.4.1 5-Gene Synthetic Network

As a first step towards evaluating the performance of our method, we consider the 5-gene target network given in Figure 3.3 [187,190]. For this network, its initial conditions and S-system model parameters (g and h) are available in the literature [187,222]. We use the Runge-Kutta integration method to obtain the 3 sets of time series data, each having 30 time points. We use 10 such different combined datasets in our simulations and calculate the above four performance measures using our BITGRN technique and compare the performance with five different DBN techniques reported earlier, namely, DBN with nonparametric regression, DBN (NPR) [114], BANJO [255], dynamic programming based DBN method, DBN(DP) [70] and BNFinder (BDe and MDL) [66]. The results of this comparative study are shown in Table 3.2. Referring to Table 3.2, the results for the network having the best performance measure are given in row 1, whereas the average and standard deviation of the measures (corresponding to the 5 different combined datasets) is given in row 2. Row 3 provides the values for the dynamic programming based DBN method. Rows 4, 5 and 6 give the values for the nonparametric regression based DBN method, BANJO, and BNFinder methods respectively. We observe that the average values of sensitivity and F-score of our method are higher than all the other algorithms. In terms of specificity and precision, BANJO performs better compared to BITGRN. However, BANJO infers a far fewer correct interactions. Also, with regard to the balance

between sensitivity and precision, BITGRN performs much better compared to BANJO, indicating a better overall performance by BITGRN.

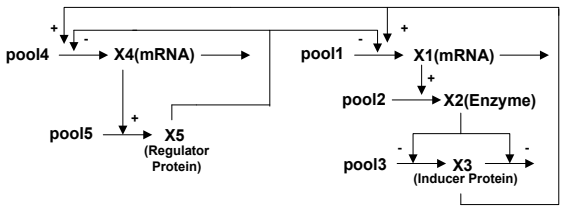

Figure 3.3: 5-gene target network. Source: [190].

	Se	***Sp***	***Pr***	***F***
BITGRN(Best)	0.75	0.92	0.9	0.82
BITGRN	$\mathbf{0.67 \pm 0.09}$	0.86 ± 0.08	0.82 ± 0.11	$\mathbf{0.74 \pm 0.091}$
DBN(DP)	0.5	0.89	0.82	0.62
DBN(NPR)	0.67	0.77	0.73	0.70
BANJO	0.47 ± 0.11	$\mathbf{0.97 \pm 0.04}$	$\mathbf{0.93 \pm 0.09}$	0.62 ± 0.1
BNFinder+BDe	0.27 ± 0.09	0.69 ± 0.12	0.46 ± 0.14	0.33 ± 0.097
BNFinder+MDL	0.35 ± 0.12	0.58 ± 0.09	0.43 ± 0.09	0.38 ± 0.011

Table 3.2: Performance comparison of BITGRN with DBN(DP), DBN(NPR), BNFinder and BANJO on 5-gene S-system based synthetic network

Effect of beta: To study the effect of beta, we ran the algorithm with different beta values. The results are shown in Figure 3.4. We observe from the results that increasing the value of beta has a positive effect on the specificity and precision. This is expected since an increasing value of beta means a stricter test of conditional independence. However, as we observe from the other two measures, the number of inferred arcs in the case of high beta values is low, thereby reducing the value of sensitivity and F-score. We observe that from the beta value 0.4 onwards, there is a 9 percent increase in specificity and 13 percent increase in precision. On the other hand, in the same region, sensitivity reduces by 22 percent, and F-score reduces by around 15 percent. The beta value of 0.4 can thus be considered a suitable cut-off point for a balance of Se, Sp, Pr and F.

3.4.2 20-Gene Synthetic Network

To study the effect of the number of samples and noise, we use a larger network, containing 20 genes, as shown in Figure 3.5 [163]. The network should also enable us to study the

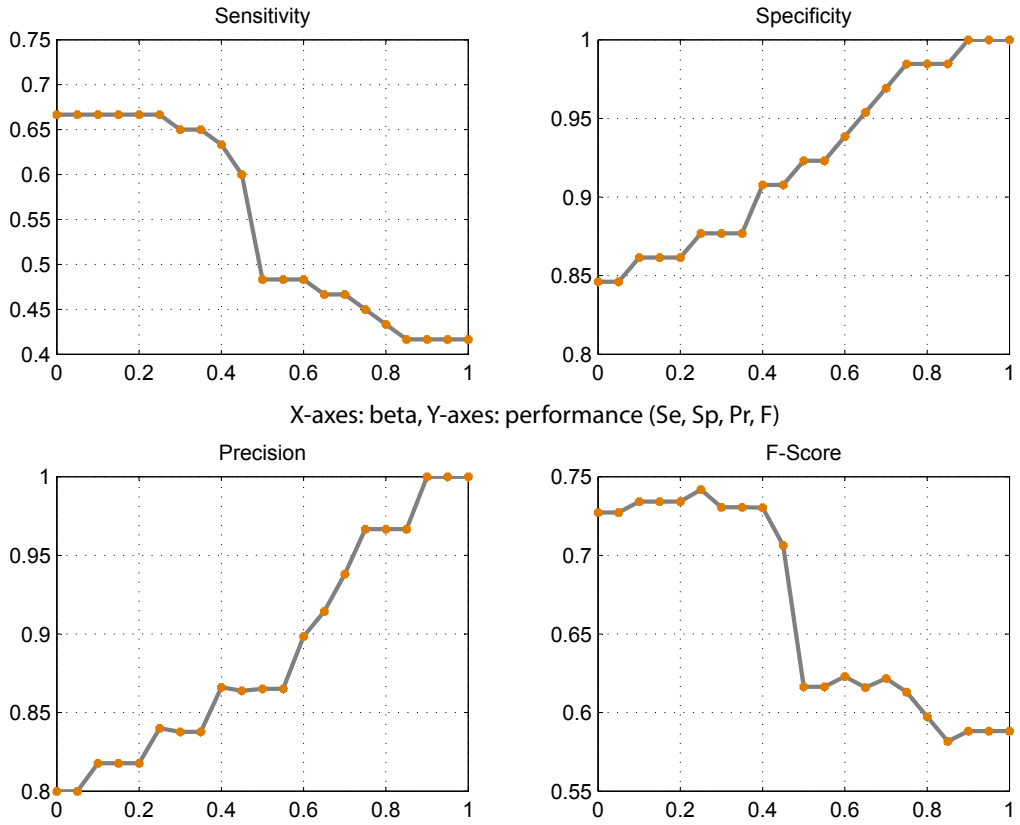

Figure 3.4: Effect of changing beta on the 5-gene S-system based network. X-axes: Beta values; Y-axes: Performance measures.

effect of beta on a larger network. We used the same parameters as described in Noman *et al.* [163] (shown in Table 3.3) for data generation. This artificial network models several types of regulations (for example, cyclic regulation and feedback regulation) commonly found in biochemical networks, which makes it a standard simulation model [163]. For our experiments, the number of data points was varied to observe the effect of data points (20 and 30 data points for each dataset). To study the effect of noise, we added 6 different levels of noise (random Gaussian noise with zero mean and variance, $\sigma^2 = 0.05, 0.1, 0.2$). The results are summarized in Figures 3.6 and 3.7. From the figures, we observe that increasing the number of data points improves the performance of the method, especially with regard to handling noisy data. As an example, we see from the 60-samples data (3×20) 3.6 that due to the increase of noise from 0.05 to 0.20, the sensitivity falls from 0.4 to 0.33 (for noise with variance 0.1) and then to 0.23 (variance 0.2). However, for the 90-samples data, the decrease is much less (0.4 to 0.35 and then to 0.345). The same is true for precision, and since F-score is a harmonic mean of these two, it also shows a similar trend. The specificity values are high for both the datasets, and they remain a

bit insensitive to noise levels (albeit the same trend is visible also here). With regard to the beta values, we observe that for the 60-samples data, there is a slightly downward trend of sensitivity for increasing values of beta. However, precision takes a slightly upward path, and the overall F-score remains largely less sensitive to the beta changes with a slightly upward trend with increased values of beta. Specificity increases with higher values of beta. This trend of higher values for the higher beta values is expected since as mentioned, increasing the value of beta means a stricter conditional independence test. However, for the higher number of samples, the trend of an increase in performance with an increase in beta values is much less pronounced; we see a slightly downward trend of F-score for these datasets, especially for higher noise levels, and an expected downward trend of sensitivity with increasing beta values (corresponding to stricter conditional independence tests). This supports our hypothesis that when the noise levels are high, a very strict conditional independence may not help much. The number of false predictions decreases with stricter conditional independence tests for the 30 samples per dataset case also, although this time the slope is less, and for higher noise levels, the increase is even less pronounced.

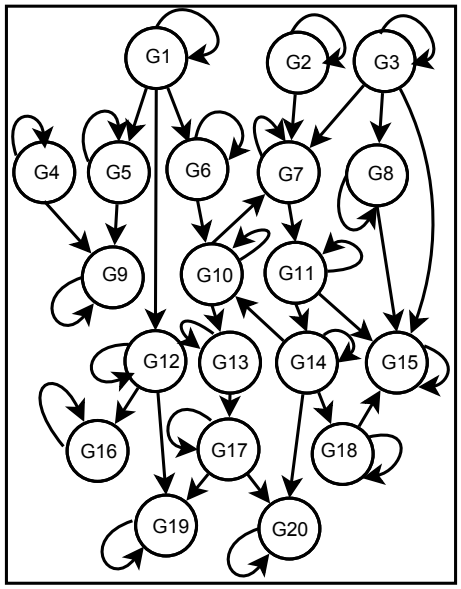

Figure 3.5: 20-node target network

α_i, β_i	10.0					
$g_{i,j}$	$g_{3,15} = -0.7$,	$g_{5,1} = 1.0$,	$g_{6,1} = 2.0$,	$g_{7,2} = 1.2$,	$g_{7,3} = -0.8$,	$g_{7,10} = 1.6$,
	$g_{8,3} = -0.6$,	$g_{9,4} = 0.5$,	$g_{9,5} = 0.7$,	$g_{10,6} = -0.3$,	$g_{10,14} = 0.9$,	$g_{11,7} = 0.5$,
	$g_{12,1} = 1.0$,	$g_{13,10} = -0.4$,	$g_{13,17} = 1.3$,	$g_{14,11} = -0.4$,	$g_{15,8} = 0.5$,	$g_{15,11} = -1.0$,
	$g_{15,18} = -0.9$,	$g_{16,12} = 2.0$,	$g_{17,13} = -0.5$,	$g_{18,14} = 1.2$,	$g_{19,12} = 1.4$,	$g_{19,17} = 0.6$,
	$g_{20,14} = 1.0$,	$g_{20,17} = 1.5$,	$g_{i,j} = 0.0$ otherwise			
$h_{i,j}$	1.0 if $(i = j)$, 0.0 otherwise					

Table 3.3: Parameter values used for generating data for 20-node S-system network

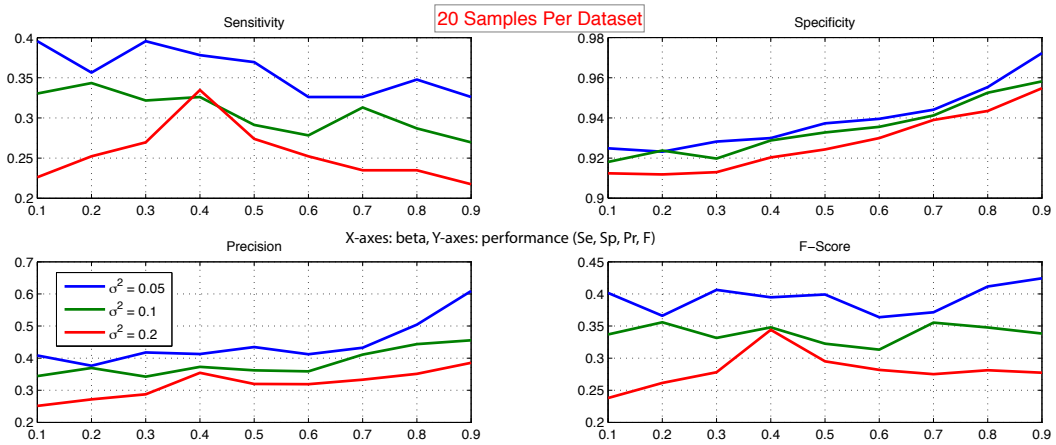

Figure 3.6: Effect of beta and noise on the performance of BITGRN using 60 samples (20 samples per dataset) on the 20-node S-system network. X-axes represent the beta values (0.1 to 0.9). Y-axes represent the corresponding performance measures. The three noise levels in three different colors are indicated in the legend.

3.5 Experimental Results Using Real-Life Biological Data

3.5.1 IRMA Network

Existing literature has frequently used a dataset from Spellman *et al.* [214], which was published in 1998. One of the main problems with this dataset is that it is highly noisy, and the effect of endogenous genes is quite prominent. Hence, if one takes a small number of genes for testing, combined with the fact that there is no benchmark network topology for the overall gene network, it becomes very difficult to quantitatively assess the effectiveness of the method. However, it should also be mentioned here that the dataset of Spellman *et al.* is still frequently used as a test dataset. The point being emphasized is that for quantitative assessment, it is better to use datasets which have a better-known network topology and a relatively less-noisy dataset. As a result, to validate our BITGRN method with a real-life biological gene regulatory network, we investigate a recent network reported in Cantone *et al.* [35]. In that significant work, the authors built a network,

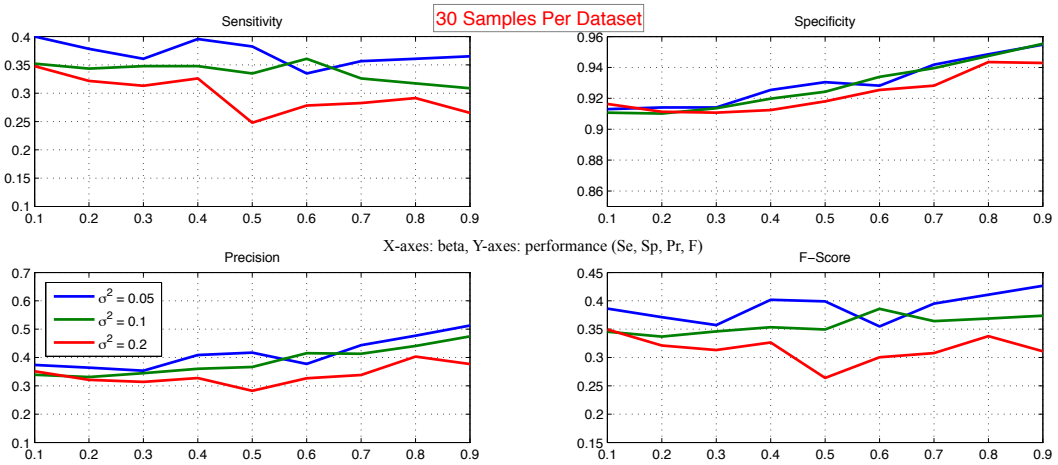

Figure 3.7: Effect of beta and noise on the performance of BITGRN using 90 samples (30 samples per dataset) on the 20-node network. X-axes represent the beta values (0.1 to 0.9). Y-axes represent the corresponding performance measures. The effect of the three noise levels in three different colors are indicated in the legend.

called IRMA, of the yeast *Saccharomyces cerevisiae*, for in vivo benchmarking of reverse-engineering and modeling approaches. They tested the transcription of network genes by culturing the cells in the presence of galactose and glucose. The network is composed of five genes regulating each other; it is also negligibly affected by endogenous genes. The time series and steady-state expression data are both measured after introducing different perturbations to the network. This is one of the first attempts at building a reference dataset having an accurately known target network [261]. There are two sets of gene profiles called Switch ON and Switch OFF for this network, each containing 16 and 21 time series data points, respectively. The former corresponds to the shifting of the growing cells from glucose to galactose medium, the latter to the reverse phase. Some edges in the original network actually represent protein level interactions and since they are not directly gene-gene regulation, a simplified network is also reported. We compare our reconstruction method with 4 other methods, namely, TDARACNE [261], BANJO [255], ARACNE [139], and BNFinder (both BDe and MDL). These methods have previously been successfully used for reconstructing the networks under consideration.

IRMA ON Dataset

The results for the ON state data are shown in Figure 3.8. Of the total 8 arcs for the original network, our method correctly identifies 5 arcs. One arc has a wrong direction. For

the simplified network, the method recovers 4 edges, all of which are correct. The performance comparison amongst various methods is shown in Table 3.4. From the table, we can see that BITGRN clearly outperforms all the other algorithms. In terms of sensitivity, BITGRN and TDARACNE perform the best. However, in terms of all the other performance measures, BITGRN outperforms the other algorithms, both for the original network and also the simplified network. The effect of beta values on the performance of the algorithm is shown in Figure 3.9. From the graph, we get observations similar to the performance on the 20-node synthetic network data with higher noise levels. Both sensitivity and precision continue to decrease with stricter conditional independence tests, and the F-score thus follows this trend. Sensitivity remains indifferent to the changes in beta.

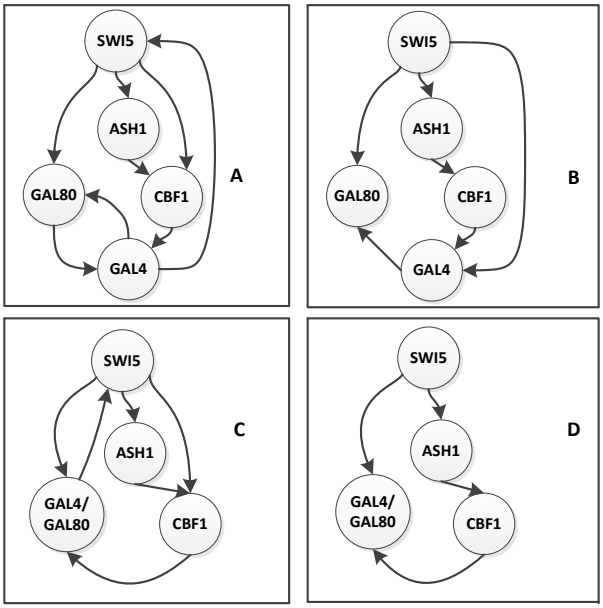

Figure 3.8: Yeast network (ON dataset) and BITGRN inferred network. (A) Target network. (B) Inferred network by BITGRN. (C) Target network, simplified. (D) Inferred network, simplified.

IRMA OFF Dataset

As has been indicated in Zoppoli *et al.* [261], the OFF dataset lacks the presence of a great stimulus, making it difficult to reconstruct the exact network. Even with this limitation of the OFF dataset, our proposed method performs better. The comparison is shown in Table 3.5. As we can see, for the original network, the F-score and the sensitivity are pretty high compared to the other methods. BANJO performs better in terms

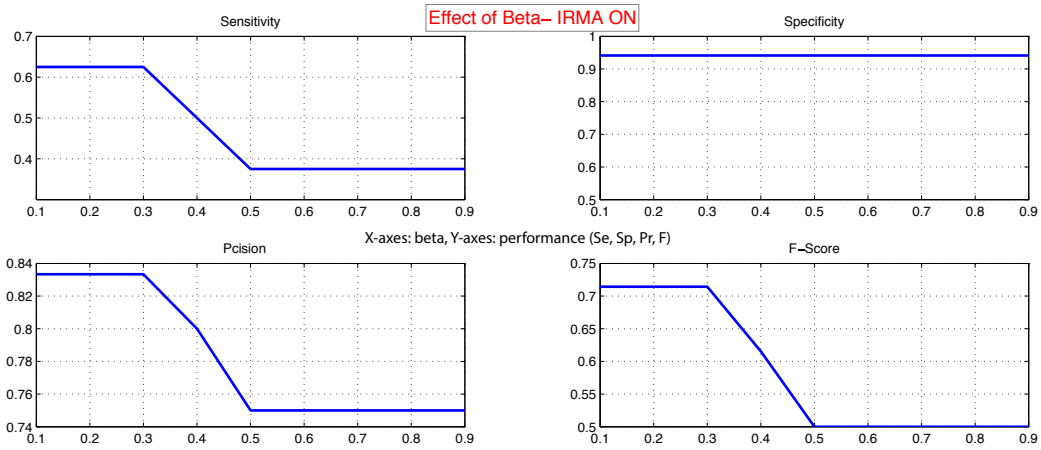

Figure 3.9: Effect of Beta on the performance of BITGRN, using IRMA ON dataset

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
BITGRN	**0.63**	**0.94**	**0.83**	**0.71**	**0.67**	**1.0**	**1.0**	**0.80**
TDARACNE	**0.63**	0.88	0.71	0.67	**0.67**	0.90	0.80	0.73
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50
ARACNE	0.60	-	0.50	0.54	0.50	-	0.50	0.50
BNFinder+BDe	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BNFinder+MDL	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22

Table 3.4: Performance comparison of BITGRN based on IRMA ON dataset

of specificity and precision, albeit the F-score and especially sensitivity are much lower compared to BITGRN. For the simplified network, BANJO only performs better in terms of the number of false predictions (specificity), and BITGRN performs best in terms of all the other performance measures. Hence, it can be considered that BITGRN is the better overall performer.

The effect of beta values on the performance of the algorithm is shown in Figure 3.10. We get observations similar to the ON dataset, and see that sensitivity continues to fall overall with increased values of beta. Although precision shows an initial upward trend with increasing beta, after a certain beta value (in this case 0.2) the performance degrades with stricter conditional independence tests. Following this trend, F-score initially has an upward trend, albeit, after a beta value of 0.2, a gradual performance degradation occurs. Specificity increases with increasing values of beta, rising from 0.71 to 0.82 while the beta values change from 0.1 to 0.9.

	Original Network				Simplified Network			
	Se	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*
BITGRN	**0.63**	0.76	0.56	**0.59**	**0.83**	0.89	**0.71**	**0.77**
TDARACNE	0.60	-	0.37	0.46	0.75	-	0.50	0.60
BANJO	0.38	**0.88**	**0.60**	0.46	0.33	**0.90**	0.67	0.44
ARACNE	0.33	-	0.25	0.28	0.60	-	0.50	0.54
BNFinder+BDe	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BNFinder+MDL	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40

Table 3.5: Performance comparison of BITGRN based on IRMA OFF dataset

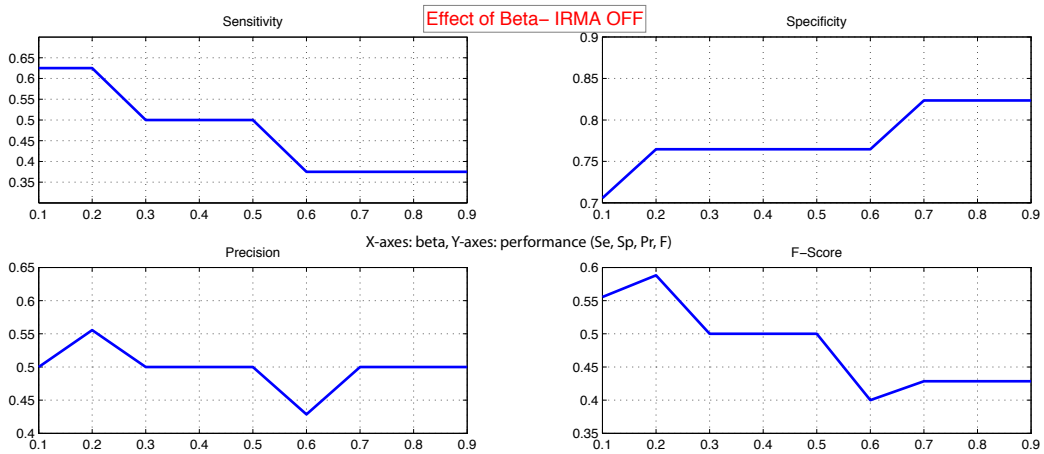

Figure 3.10: Effect of Beta on the performance of BITGRN, using IRMA OFF dataset

3.5.2 SOS DNA Repair Network of *E. coli*

We analyze the well known SOS DNA Repair Network in *Escherichia coli* as shown in Figure 3.11(A). This GRN is well known for its responsibility of repairing the DNA if it gets damaged. It is the largest, most complex, and best understood DNA damage-inducible network to be characterized to date.

The expression of the genes in the SOS regulatory network is controlled by a complex circuitry which involves the RecA and LexA proteins [149]. Normally, LexA acts as the master repressor of more than 20 genes, including lexA and recA genes. This repression is done by its binding to the interaction sites in the promoter regions of these genes. When DNA damage occurs, one of the SOS proteins, RecA, acts as a sensor. By binding to single-stranded DNA, it becomes activated, senses the damage and mediates LexA autocleavage [149]. The drop in LexA levels in turn stops the repression of the SOS genes and activates them. When the damage has been repaired, the level of activated RecA

drops and it stops mediating LexA autocleavage. LexA level in turn increases, starting repression of the SOS genes, and the cell then returns to its normal state.

The expression datasets of the SOS DNA repair system were obtained from Uri Alon Lab [8]. These data are expression kinetics of 8 genes, namely uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB. Four experiments were carried out with various UV light intensities (Exp. 1 and 2:$5Jm^{-2}$, Exp. 3 and 4:$20Jm^{-2}$). In each experiment, the above 8 genes were monitored (along with other genes) at 50 instants which are evenly spaced by 6 minutes intervals.

The results corresponding to Experiment 1 are presented in Figure 3.11(B). Along with our result, we include the results from BANJO, TDARACNE and BNFinder in Figure 3.11(C)-(F) and the target network in 3.11(A). From the results in Figure 3.11, we observe that our method correctly identifies lexA as one of the 'hub' genes for this network. Again, the exact ground truth for this network is not precisely known, and hence it has not been possible to calculate the well known performance measures. Instead, using the known interactions obtained from the literature [116, 163], an analysis of correct and incorrect predictions by our method was carried out and it is shown in Table 3.6. We observe that four inferred interactions by our proposed method are correct. It successfully infers lexA as the regulator of uvrA, ruvA and umuD. Also, considering the indirect regulation of RecA through LexA, one more interaction, namely recA$\rightarrow$polB can also be considered correct. In contrast, 3 of the 5 identified interactions by TDARACNE [261] are correct. Both BNFinder+BDe and BNFinder+MDL successfully identify regulation of ruvA, polB and uvrA by lexA. In addition, the regulation of umuD by recA can also be considered correct (indirect regulation). Although BNFinder (both BDe and MDL) finds the same number of correct interactions as BITGRN, they also infer a lot of incorrect arcs (seven and six incorrect arcs, respectively), when compared to BITGRN. Finally, BANJO infers the highest number of correct arcs (5); however, in terms of incorrect arcs, it is also the highest (eight incorrect arcs). Thus, overall we see that the performance of BITGRN is the most balanced, with respect to the number of correct and incorrect predictions.

Further, considering the results corresponding to Experiments 2, 3 and 4 (see Figure 3.12 for results corresponding to these experiments), we see that for both Experiments

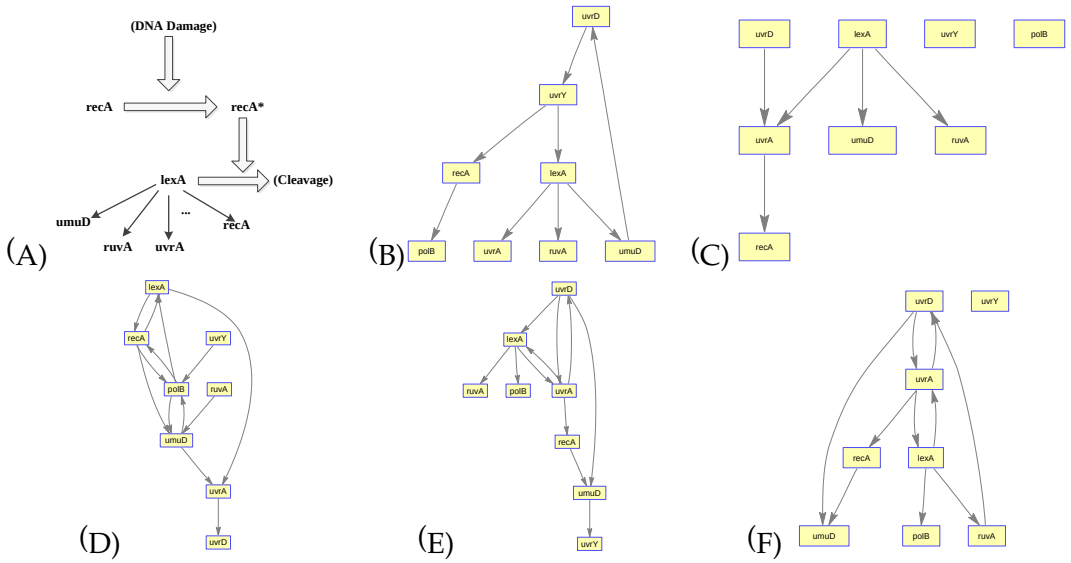

Figure 3.11: Reconstruction of SOS DNA Repair Network (Experiment 1). (A) Target Network. (B) Network Inferred by BITGRN. (C) Network Inferred by TDARACNE. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

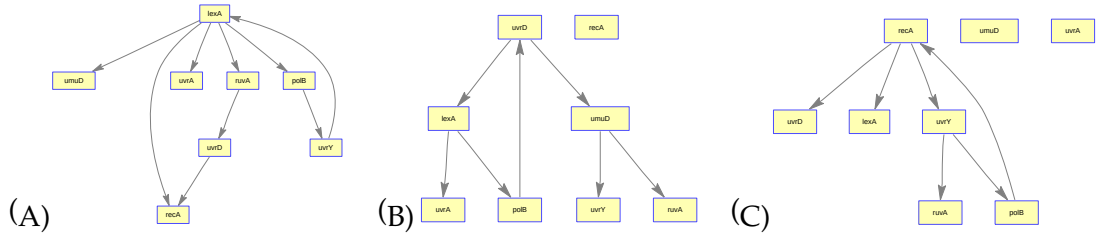

Figure 3.12: Reconstruction of SOS DNA Repair Network (Experiments 2, 3, 4). (A) Experiment 2. (B) Experiment 3. (C) Experiment 4.

2 and 3, BITGRN correctly infers lexA as one of the main hubs. BITGRN infers five correct arcs for Experiment 2, and two from the Experiment 3 dataset. For Experiment 4, it infers three correct and three incorrect arcs.

To find out a 'consensus' value for beta, we consider Figures 3.4, 3.6, 3.7, 3.9 and 3.10. The results come from both synthetic and real-life biological data, and contain both noise-free (the 5-gene synthetic network results) and noisy datasets (20-gene synthetic network and the IRMA datasets), and also, large datasets (the synthetic datasets) and small datasets (IRMA datasets). We observe from the graphs that for noiseless data usually the higher the beta value, the better is the performance. However, as we see from Figure 3.4, increasing beta value has negative effects on the true arcs discovered, and thus

Regulator	Target	correct/ incorrect
lexA	uvrA	correct
	umuD	correct
	ruvA	correct
recA	polB	correct[a]
uvrD	uvrY	incorrect
uvrY	recA	incorrect
uvrY	lexA	incorrect
umuD	uvrD	incorrect

[a] correct considering indirect regulation of RecA through LexA

Table 3.6: Analysis of individual interactions inferred by BITGRN - SOS DNA Repair Network

there is a need for a balance between true arcs versus correct predictions. For the noisy and real-life datasets (which inherently contain some noise), we observe that the need of a balance is even more evident. Considering Figures 3.6, 3.7, 3.9 and 3.10, and taking a disjunctive approach, we observe that when the number of samples is low or the data is noisy, starting from beta value 0.2, there is usually an overall downward trend in the performance measures, thus suggesting a beta value of 0.2. Similar consideration for noise-free datasets suggest a value of 0.4 for beta. It may be noted here that the specificity value will always increase with increasing values of beta; however, usually for BITGRN, the specificity values are consistently higher compared to other approaches due to rigorous statistical significance based conditional independence tests, so the effect of higher specificity for higher values of beta is of comparatively less significance.

3.6 Summary

In this chapter, we proposed BITGRN, a novel MI/CMI based dynamic Bayesian network learning framework for reconstructing gene regulatory networks. The method employs CI tests pertaining to information theoretic quantities to find statistically significant genetic interactions, thereby reconstructing gene regulatory networks with high accuracy.

Studies on both synthetic and real biological datasets showed good performance. Even in cases where other algorithms perform poorly due to a lack of stimulus, the BITGRN algorithm performed much better compared to other related algorithms. The high accuracy of the proposed approach is clearly demonstrated with the aid of simulation experiments. The method is flexible in the sense that it can allow the incorporation of additional biological knowledge, and a 'tunable' parameter makes it capable of handling weak regulations which is particularly true for gene regulatory networks. We also prescribed the values that should be used in different scenarios based on simulation studies.

Nevertheless, currently the proposed approach can only model time-delayed interactions among genes. However, in biological systems, genes may interact with other genes either almost instantaneously, or with time delay. In the next chapter, we extend the concepts and methods developed in this chapter to incorporate both instantaneous and time-delayed interactions for a more accurate and realistic GRN modeling.

Chapter 4

Realistic Modeling of Genetic Interactions

4.1 Introduction

In the previous chapter, we proposed a novel algorithm which learns the structure of a GRN employing MI/CMI based CI tests. A standard DBN based modeling technique was used where it was assumed that all gene-gene interactions in the GRN are time-delayed. However, in any biological system, various genetic interactions occur concurrently amongst different genes with some interactions being time-delayed and some occuring almost instantaneously. To the best of our knowledge, there is no systematic study reported on modeling these two types of genetic interactions together. In this chapter, we propose a modeling framework which, unlike standard DBN based GRN reconstruction techniques, provides more accurate and realistic representation of the biological processes - by modeling both instantaneous and time-delayed interactions.

When considered from a biological perspective, instantaneous regulations in GRNs represent the effect when a change in the expression level of a regulator gene is carried on to the regulated gene (almost) instantaneously[1]. In these cases, the effect will be reflected almost immediately in the regulated gene's expression level [158]. On the other hand, in cases where regulatory interactions are time-delayed in nature, the effect may be

[1]The time it takes to carry the effect will always be greater than zero. However, if the delay is small enough so that the regulated gene is affected before the next data sample is taken, it can be considered an instantaneous interaction.

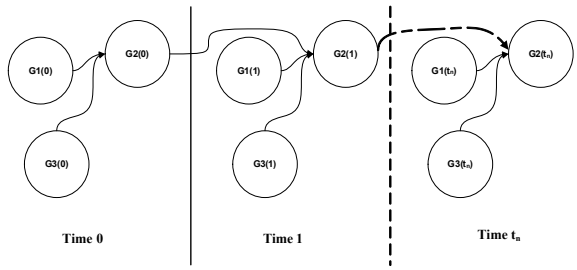

Figure 4.1: Proposed network structure for the BN based modeling. Arcs between genes across time slices (time-delayed interactions) are accompanied by arcs within time slices (instantaneous interactions).

seen on the regulated gene after some time. To the best of our knowledge, the currently existing techniques that use time series data make simplifications and assume that the interactions can be either of these but not both, i.e., they assume that either the effect is instantaneous or it maintains a d-th order Markov relation with its regulator (i.e., regulations occur between two time slices, which may be d time steps apart, $d = 1, 2, \dots$). For example, all BN based GRN reconstruction methods consider the regulatory effects to be instantaneous and cannot model time-delayed interactions directly. In contrast, developing dynamic probabilistic networks [77] requires defining an initial network and a transition network. This type of representation also does not capture both types of interactions. DBN based methods assume that regulations occur with a certain amount of time delay. Thus, they cannot model instantaneous interactions. In this chapter, to model realistic genetic interactions, we unify the approaches of BN and DBN in a systematic manner and propose a framework shown in Figure 4.1 for capturing both types of interactions.

In this chapter, to achieve the objective of capturing both instantaneous and time-delayed interactions, we first describe a modeling framework that can represent both these interactions. However, to harness the power of the modeling framework, one needs a learning strategy. The learning for this modeling framework, in general, can be accomplished in two ways: by utilising the information theory based concepts developed in the previous chapter, or employing a score and search strategy. We investigate both these approaches. For learning based on the proposed framework, we first present a two-phase GRN inference algorithm that can *sequentially* learn these two types of interactions. Next, we employ a *score+search* based approach, using a decomposable scoring metric and a genetic algorithm based evolutionary strategy for searching the large space of possible

solutions. We assess the performance of both the algorithms, using synthetic networks as well as real-life biological networks, with regard to the four well known performance measures used in the previous chapter. The results from these sequential learning approaches prove that both these types of interactions are necessary and essential for accurate and realistic modeling.

The rest of the chapter is organized as follows. In Section 4.2, we present the framework that we propose for representing both instantaneous and time-delayed interactions. Section 4.3 explains the proposed methodology and its formalization using information theoretic quantities. Section 4.4 discusses the synthetic and real-life networks used for assessing our approach and also its comparison with other techniques. Section 4.5 proposes a GA based strategy using the proposed framework, and its evaluation is shown in Section 4.6. Section 4.7 concludes with some observations and remarks.

4.2 The Modeling Framework

To employ information theoretic quantities to the problem of reconstructing a GRN from data that can capture both instantaneous and time-delayed interactions, let us model a GRN containing n genes (denoted by X_1, X_2, X_n), with a corresponding microarray dataset having t_n time points. A DBN based method would try to find associations between genes X_i and X_j by taking into consideration the data $x_{i1}, \ldots, x_{i(t_n-1)}$ and $x_{j2}, \ldots, x_{jt_n}$ or vice versa (small case letters mean data values in the microarray). This will effectively enable it to capture single-step (corresponding to a first-order Markov assumption) time-delayed interactions. On the other hand, a BN based strategy would use the whole t_n time points and it would capture regulations that are *deemed* to be effective instantaneously.

Now, let us double the number of nodes in the way shown in Figure 4.2. The first n nodes of this new network model will correspond to the data $x_{k1}, \ldots, x_{k(t_n-1)}$ whereas the second half will contain $x_{k2}, \ldots, x_{kt_n}$, $k = 1, 2, \ldots, 2n$. Hence, from this data, if we use the BN formalism to construct a final network where we see, for example, edge $X_1 \rightarrow X_2^{'}$, we conclude that the time-delayed interaction between X_1 and X_2 is recovered. Similarly, if we find that $X_2 \rightarrow X_5$, we say that the instantaneous interaction between X_2 and X_5 is recovered. In this way, we can capture both types of interactions.

It should be noted that, along with this formalism, the following four assumptions are also mandatory to ensure consistency:

Assumption 4.2.1. ***DAG Constraint.*** *The intra-slice portion of the network must be a DAG. In other words, since the leftmost n-columns are representative of the Bayesian network (instantaneous interactions), this portion must correspond to a network that does not contain any cycle.*

The second assumption ensures that information flows in the correct direction.

Assumption 4.2.2. ***Information flow constraint.*** *The inter-slice arcs must not contain any backward arcs. All arcs from genes in a particular time point must go to genes which are in a time point later than the current gene under consideration.*

Further, we also make the following first-order Markov assumption to account for single-step time-delayed interactions:

Assumption 4.2.3. ***First-Order Markov Assumption.*** *The first-order Markov assumption says that given the current observation $\boldsymbol{X}[t]$, the next observation $\boldsymbol{X}[t+1]$ is independent of past observations, $\boldsymbol{X}[0], \ldots, \boldsymbol{X}[t-1]$ (or put more simply: the future is independent of the past given the present). Formally,*

$$P\left(\boldsymbol{X}[t]|\boldsymbol{X}[t-1], \ldots, \boldsymbol{X}[0]\right) = P\left(\boldsymbol{X}[t]|\boldsymbol{X}[t-1]\right) \tag{4.1}$$

Next, we need to make the stationarity assumption, which is necessary for the BN/DBN based reconstruction techniques:

Assumption 4.2.4. ***Stationarity.*** *The transition probability $P(\mathbf{X}[t]|\mathbf{X}[t-1])$ is independent of t. In other words, interactions remain in existence irrespective of time.*

	X_1	X_2	$\ldots$	X_n	X_1'	X_2'	$\ldots$	X_n'
X_1	0	1	$\ldots$	0	1	0	$\ldots$	1
X_2	0	0	$\ldots$	1	1	0	$\ldots$	0
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
X_n	0	0	$\ldots$	0	0	1	$\ldots$	1

Figure 4.2: The adjacency matrix based approach for the representation

Under Assumptions 4.2.1- 4.2.4, this DBN based model encodes the joint probability distribution over the random variables (corresponding to the genes $X_1, \ldots, X_n$) $\boldsymbol{X}[0] \bigcup \boldsymbol{X}[1] \bigcup \ldots \boldsymbol{X}[T]$, which is defined by Equation 4.2.

$$P(\boldsymbol{x}[0], \ldots, \boldsymbol{x}[T]) = \prod_{t=1}^{T} P(\boldsymbol{x}[t]|\boldsymbol{x}[t-1]) \tag{4.2}$$

Finally, the transition probability defined by this network model is defined as follows.

$$P(\boldsymbol{x}[t]|\boldsymbol{x}[t-1]) = \prod_{i=1}^{n} P(x_i[t]|Pa(X_i[t])) \tag{4.3}$$

Here, as we already defined in the previous chapter, the uppercase letters denote genes (i.e., random variables corresponding to the gene's expression value), lowercase letters denote specific values taken by these variables, and boldface letters denote sets of variables. $X_i[t]$ denotes the expression value of gene X_i at time t.

We will demonstrate the application of this modeling framework in two ways. First, for smaller scale networks and for the purpose of demonstrating the concept of the detection of two types of interactions, we apply a two-phase hill-climbing search based approach using the information theoretic properties introduced in the previous chapter. Next, we propose a more sophisticated *score and search* based algorithm, using a suitable evolutionary search technique, that should cope better with more complex networks. We explore both these approaches in the following sections.

4.3 GRN Reconstruction with Contemporaneous Arcs Using Information Theory

4.3.1 The Search Strategy for Time-Delayed Interaction Detection

We employ the same search strategy for the inter-slice (time-delayed) arcs as we proposed in the previous chapter: using MI/CMI based conditional independence tests. As we explained in the previous chapter, regulatory relationships can be assessed by using Mutual Information (MI) based CI tests. A low value of MI between two genes implies the genes are conditionally independent while on the other hand, a high MI value, means a higher

likelihood of a relation between the genes. This interaction may be a direct or indirect relation. This poses a problem which, however, can be overcome by using the Data Processing Inequality (DPI) [60], which states that if nodes A and C are connected through an intermediate node B, then $MI(A, C) = \min\{MI(B, A), MI(C, B)\}$, i.e., the lowest MI value is always for the indirect relationships. Hence, while adding the first parent, we calculate pair-wise MI between the gene under consideration, and all the parent candidates. The parent candidate which has the highest MI with the gene under consideration is ultimately selected as the parent.

Next, while adding subsequent parents, we calculate how much additional information we get about X_i by adding a candidate parent (X_j) as a parent of this gene, using $MI(X_i, X_{CP_k(X_i)}|Pa_c(X_i))$, where $X_{CP_k(X_i)}$ represents genes that are in the current candidate parent set of the gene X_i and $Pa_c(X_i)$ represents the current parent set of X_i. The candidate gene which can best explain the unexplained uncertainty of X_i relative to the current parent set of this gene is added as the parent of X_i.

However, merely getting a high MI value does not suffice to make it statistically significant. To assess whether the gain in information is statistically significant, we use the theorem of Kullback [123]. According to the theorem, for a particular confidence level α, determining the value of $\chi(\alpha, df_{ik})$ such that

$$p(\chi^2(df_{ik}) \leq \chi(\alpha, df_{ik})) = \alpha \tag{4.4}$$

represents a statistical test of conditional independence [60]. Here df_{ik} represents the degrees of freedom defined by the following equation:

$$df_{ik} = \begin{cases} (r_i - 1)(r_{ik} - 1) \prod_{m=1}^{j-1} r_{im}, & j \geq 2 \\ (r_i - 1)(r_{ik} - 1), & j = 1 \end{cases} \tag{4.5}$$

where r_{im} is defined by:

$$\begin{gathered} r_{im} = config(X_m), \\ X_m \in Pa_c(X_i) \end{gathered} \tag{4.6}$$

here $config(X_m)$ is the number of possible states/values that gene X_m can take. Based on the theorem, we can say that the test for statistical significance would assert that the genes are dependent if, in a data set containing N elements,

$$2N.MI(X_i, X_{CP_k(X_i)}|Pa_c(X_i)) \gg \chi(\alpha, df_{ik}) \tag{4.7}$$

Conversely, the genes are conditionally independent if

$$2N.MI(X_i, X_{CP_k(X_i)}|Pa_c(X_i)) < \chi(\alpha, df_{ik}) \tag{4.8}$$

Thus, if the maximum CMI value for the current candidate parent set fails this test, we stop adding parents to gene X_i.

4.3.2 Finding the Directions of Instantaneous Arcs

The arcs corresponding to time-delayed interactions in the network can be deduced uniquely since for this part, we are effectively calculating $MI(X_k[t], X_i[t+1]|Pa_c(X_i[t]))$. However, this is not the case with instantaneous (intra-slice) arc additions. Since MI is symmetric, the directions of the instantaneous arcs cannot be uniquely determined. To determine the direction of the instantaneous arcs, we use the directionality index, DI_{XY}.

The Directionality Index [131] between genes *X* and *Y* is defined as:

$$DI_{XY} = \frac{MI_{X\to Y} - MI_{Y\to X}}{MI_{X\to Y} + MI_{Y\to X}} \tag{4.9}$$

where the quantities $MI_{X\to Y}$ and $MI_{Y\to X}$ are defined by the following equations:

$$MI_{X\to Y} = \frac{1}{N}\sum_{\delta=1}^{N} MI^{\delta}_{X\to Y} \tag{4.10}$$

$$MI_{Y\to X} = \frac{1}{N}\sum_{\delta=1}^{N} MI^{\delta}_{Y\to X} \tag{4.11}$$

Here, the quantities in the left side of the equation 4.10 and 4.11 quantify the information that is gained from the gene *X* (or *Y*) about the gene *Y* (or *X*) at some later point in time and *N* is the maximal later point.

If we assume that the quantity X_δ (or Y_δ) is an observable derived from the state of the gene X (or Y) δ steps in the future, i.e., $X_\delta : x_{t+\delta} = x_t$ (or $Y_\delta : y_{t+\delta} = y_t$), $MI^\delta_{X\to Y}$ and $MI^\delta_{Y\to X}$ can be defined in terms of Conditional Mutual Information (CMI) by the following equations:

$$MI^\delta_{X\to Y} = MI\left(X, Y_\delta | Y\right) \tag{4.12}$$

$$MI^\delta_{Y\to X} = MI\left(Y, X_\delta | X\right) \tag{4.13}$$

The value of DI_{XY} ranges from -1 to +1. A positive value means that the direction of regulation between X and Y is from X to Y, whereas a negative value implies the inverse direction.

Although Directionality Indices can be used for deducing the direction of regulation, due to the finite size of the data, it may be erroneous. As a result, while applying the direction suggested by the directionality index, if any of the conditions listed in part 1 of this Section is violated (e.g., the direction violates the DAG property), we reverse the direction suggested by the directionality index and, if it does not violate the properties, we apply that direction to the corresponding edge.

The approaches described in the previous paragraphs are summarized in Table 4.1 as a 2-phase algorithm called GRNCIT (learning GRNs with Contemporaneous arcs using Information Theory). In the first phase, the inter-slice (time-delayed interactions) portion of the network is built. The second phase builds the intra-slice (instantaneous interactions) portion and the directionality index is applied to each instantaneous arc to determine the direction of interactions. The two networks are then combined to give a final gene regulatory network. The $graphRemainValid(\cdot)$ subroutine checks that any resulting configuration obtained by various operations satisfy the assumptions listed in Section 4.2. Finally, the $findPromisingParentSet(\cdot)$ subroutine, as described in the previous chapter, adds the capability of incorporating any additional information that we have regarding a particular gene. For example, if we have a-priori biological information about a particular gene violating specified preconditions for becoming a regulator of another gene, this can be included in the routine to exclude that gene as a potential parent. In addition, unlike various synthetic networks which are based on S-system based dynamics, real-life biological networks mostly do not contain auto regulation, and this

information can be used by this routine. Finally, for the instantaneous arc additions, it additionally bars duplicate testing of arc additions (e.g., if a gene has already been added as a parent of another gene with time-delayed interaction, then there is no need to consider it as a potential parent for the instantaneous interactions).

Phase 1:

for each gene $X_i \in X_{n+1,\ldots,2n}$ **do**
  $CP_k(X_i) \leftarrow findParentCandidates(X_i, Pa_c(X_i))$
  find $X_k \in CP_k(X_i)$ for which $MI(X_k[t], X_i[t+1]|Pa_c(X_i[t]))$ is maximum
  if $((maximumMI \geq \chi(\alpha, df_{ik}))$ **and** $graphRemainValid(X_k, X_i))$ **then**
    $Pa_c(X_i) \leftarrow Pa_c(X_i) \cup X_k$
  end if
  continue inclusion **until** the above test fails
end for

Phase 2:

for each gene $X_i \in X_{1,\ldots,n}$ **do**
  $CP_k(X_i) \leftarrow findParentCandidates(X_i, Pa_c(X_i))$
  find gene $X_k \in CP_k(X_i)$ for which $MI(X_k, X_i|Pa_c(X_i))$ is maximum
  if (maximum $MI \geq \chi(\alpha, df_{ik})$)) **then**
    if ($DI_{X_k X_i} > 0$ **and** $graphRemainValid(X_k, X_i)$) **then**
      $Pa_c(X_i) \leftarrow Pa_c(X_i) \cup X_k$
    else if ($graphRemainValid(X_k, X_i)$) **then**
      $Pa_c(X_k) \leftarrow Pa_c(X_k) \cup X_i$
    end if
  end if
  continue inclusion **until** the above test fails
end for

combine the two networks and get final network, ***G***

Table 4.1: Algorithm GRNCIT

4.4 Experimental Results for GRNCIT

We evaluate GRNCIT by both synthetic networks and real-life biological networks of *Saccharomyces cerevisiae* (yeast) and *E. coli*. We applied the four performance measures used previously, namely sensitivity (*Se*), specificity (*Sp*), precision (*Pr*) and F-score (*F*) for assessing the overall performance of the algorithm.

We used the Gaussian Kernel estimator to calculate MI [139, 261] from continuous data. The maximum value of the lag-parameter (δ) for the directionality index calculations was set to 5 [131]. As our method uses discrete data for the statistical significance tests, we used the Persist [150] algorithm to discretize the data into 3 levels. The value of confidence level (α) used was 0.9. For all the experiments related to synthetic network,

we used 3 different datasets for each experiment and combined these 3 datasets using the procedure described in Chapter 3, Section 3.3 [222,250].

4.4.1 Synthetic Network

5-Gene Synthetic Network

As a first step towards evaluating the performance of our method, we consider the 5-gene target network given in Section 3.3, Chapter 3 (see Figure 3.3) . We use the R-K integration method to obtain 3 sets of time series data, each having 30 time points. We use 5 such different 'combined' datasets in our simulations and calculate the above four performance measures using our technique and compare the performance with five other DBN based techniques, namely, BITGRN [157], DBN(DP) [70], DBN(NPR) [114], BANJO [255], and BNFinder (BDe and MDL) [66]. The results are shown in Table 4.2, where we observe that the values of *Se* and F-score of our method are higher than the corresponding values of the other methods. The *Sp* and *Pr* values are also quite high. Overall, we observe that although for the *Sp* and *Pr* values BANJO performs best, the other two performance measures for this method are much lower. As a result, GRNCIT achieves the overall best performance.

	Se	***Sp***	***Pr***	***F***
GRNCIT(Best)	0.83	0.85	0.83	0.83
GRNCIT(Average)	$\mathbf{0.8 \pm 0.09}$	0.82 ± 0.04	0.80 ± 0.02	$\mathbf{0.80 \pm 0.05}$
BITGRN	0.67	0.86	0.82	0.74
DBN(DP)	0.5	0.89	0.82	0.62
DBN(NPR)	0.67	0.77	0.73	0.70
BANJO	0.47 ± 0.11	$\mathbf{0.97 \pm 0.04}$	$\mathbf{0.93 \pm 0.09}$	0.62 ± 0.1
BNFinder+BDe	0.27 ± 0.09	0.69 ± 0.12	0.46 ± 0.14	0.33 ± 0.097
BNFinder+MDL	0.35 ± 0.12	0.58 ± 0.09	0.43 ± 0.09	0.38 ± 0.011

Table 4.2: Performance comparison of GRNCIT with BITGRN, DBN(DP), DBN(NPR), BANJO and BNFinder on the 5-gene synthetic network

Effect of the Number of Data Points and Noise

To study the effect of the number of data points and noise, we use the larger network shown in Chapter 3, Section 3.3, Figure 3.5. The network is composed of 20 nodes. We used the same parameters (see Table 3.3) as described by Noman *et al.* [163] for data generation. The number of data points was varied to observe the effect of data points (20 and 30 data points for each dataset). To study the effect of noise, we added 6 different levels of noise (random Gaussian noise with zero mean and variance, $\sigma^2 = 0, 0.01, 0.02, 0.05, 0.1, 0.2$). Each experiment was repeated using 5 different datasets and the averages of these results are shown in Figure 4.3. Rectangles are used in the figure for the results from the 20 data points experiment whereas triangles represent results from the 30 data points experiment. Vertical lines denote standard deviation. From the figure, we observe that increasing the number of samples increases both the accuracy of the method and the noise performance. For higher levels of noise, the more the number of data points, the better is the performance. Moreover, for low values of noise, the performance measures are similar for both the datasets, indicating that the method performs well with small data samples in these cases.

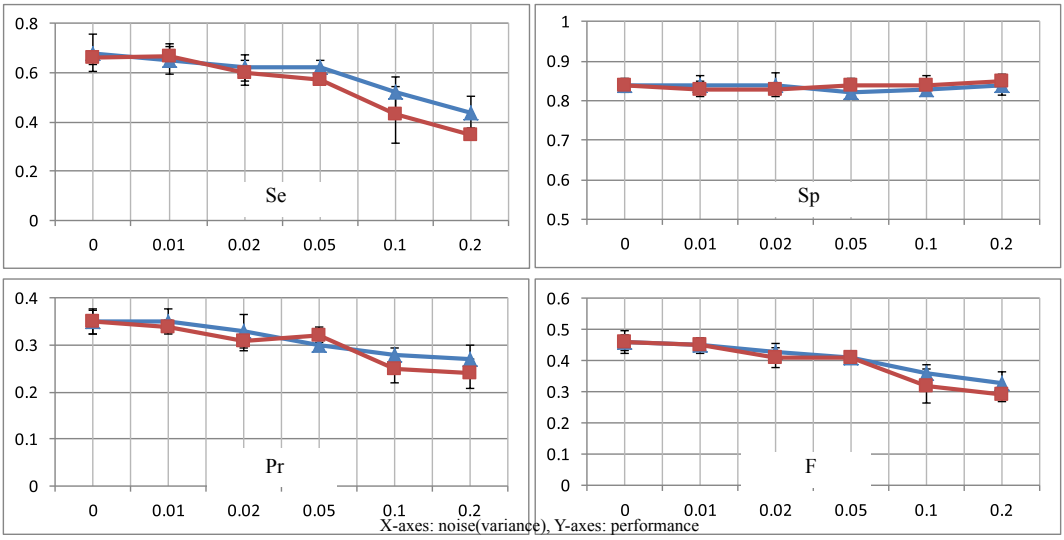

Figure 4.3: Effect of noise and data points on the performance of GRNCIT applied to the 20-node synthetic network. X-axes represent the variance values of the 6 noise levels used. Y-axes represent the corresponding performance measures. Rectangles - 20 data points (per dataset) experiment, triangles - 30 data points (per dataset) experiment. Vertical lines denote standard deviations.

4.4.2 Real-life Biological Data

IRMA Network

We use the 5-gene IRMA network presented in Figure 3.8(A), Section 3.5.1 for the initial evaluation of GRNCIT on real-life data. As described already, the network is composed of five genes regulating each other, and there are two sets of gene profiles called Switch ON and Switch OFF for this network, each containing 16 and 21 time series data points, respectively. A 'simplified' network, ignoring selected protein level interactions, is also reported in Figure 3.8(C).

IRMA ON Dataset

There are a total of 8 arcs in the original IRMA network. Using the ON dataset, our method correctly identified 6 arcs, corresponding to a sensitivity, precision and F-score of 0.75. For the simplified network, the method correctly recovered 4 arcs. The performance comparison amongst various methods is shown in Table 4.3. From the table, we can clearly see that the overall performance of our method is satisfactory. It infers the highest number of correct arcs, and the F-score is also very high. However, the number of false inferences also increases, thereby a decrease in the specificity and precision measures compared to BITGRN is observed.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
GRNCIT	**0.75**	0.88	0.75	**0.75**	**0.67**	0.89	0.67	0.67
BITGRN	0.63	**0.94**	**0.83**	0.71	**0.67**	**1**	**1**	**0.80**
TDARACNE	0.63	0.88	0.71	0.67	**0.67**	0.90	0.80	0.73
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50
ARACNE	0.60	-	0.50	0.54	0.50	-	0.50	0.50
BNFinder+BDe	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BNFinder+MDL	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22

Table 4.3: Performance comparison of GRNCIT based on IRMA ON dataset

IRMA OFF Dataset

The overall performance of the algorithms using the OFF dataset is shown in Table 4.4. From the table, we observe that the four performance measures of our method are either higher than, or comparable to the other methods. In terms of specificity and precision BANJO performs better; however, one of the reasons behind this is that BANJO infers only a few arcs (evidenced by a low sensitivity value). When compared to BITGRN, for the simplified network the performances are comparable, and for the original network structure, it performs either equal to, or better than BITGRN.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
GRNCIT	**0.63**	0.82	0.56	**0.59**	**0.83**	0.84	0.63	0.71
BITGRN	**0.63**	0.76	0.56	**0.59**	0.83	0.89	**0.71**	**0.77**
TDARACNE	0.60	-	0.37	0.46	0.75	-	0.50	0.60
BANJO	0.38	**0.88**	**0.60**	0.46	0.33	**0.90**	0.67	0.44
ARACNE	0.33	-	0.25	0.28	0.60	-	0.50	0.54
BNFinder+BDe	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BNFinder+MDL	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40

Table 4.4: Performance comparison of GRNCIT based on IRMA OFF dataset

4.4.3 SOS DNA Repair Network of *E. coli*

Next, we analyze the well known SOS DNA Repair Network in *Escherichia coli* as presented in Section 3.5.2; the network is controlled by a complex circuitry which involves the RecA and LexA proteins [149] as master repressors and master sensors, respectively. When DNA damage occurs, RecA acts as a sensor, and becomes activated after sensing the damage and mediates LexA autocleavage [149], which in turn stops the repression of the SOS genes and activates them. When the damage has been repaired, the level of activated RecA drops and it stops mediating LexA autocleavage. LexA level thus increases afterwards, starting repression of the SOS genes, and the cell then returns to its normal state. We used the same four datasets used in the previous chapter for the reconstruction of this network (obtained from the website of Uri Alon Lab [8]). The eight genes,

namely uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB, were investigated for the experiment.

The results corresponding to Experiment 1 are presented in Figure 4.4(B). Along with our result, we include the results from BANJO, TDARACNE and BNFinder in Figure 4.4(C)-(F) and the target network in 4.4(A). From the results, we observe that our method correctly identifies lexA as the 'hub' gene for this network. Again, since the exact ground truth for this network is not precisely known, it is not possible to calculate the well known performance measures. Instead, using the known interactions obtained from the literature [116,163], an analysis of correct and incorrect predictions by our method is obtained and shown in Table 4.5. From the experiment 1 dataset, GRNCIT successfully infers lexA as the regulator of uvrD, umuD, ruvA and recA. Further, considering the indirect regulation of RecA through LexA, one more interaction, namely recA→uvrY, can also be considered correct. In contrast, 3 of the 5 interactions identified by TDARACNE [261] are correct. Both BNFinder+BDe and BNFinder+MDL successfully identify regulation of ruvA, polB and uvrA by lexA. In addition, the regulation of umuD by recA can also be considered correct. BITGRN, in comparison to GRNCIT, infers one less correct arc (3 direct and 1 indirect arcs). Overall, compared to these methods, GRNCIT infers the highest number of correct predictions (5 arcs). Although BANJO also infers 5 arcs, 3 of these correct inferences are direct regulations (via lexA) and 2 are indirect (via recA) whereas 4 of the 5 correct inferences by GRNCIT are direct regulations, the remaining correct arc being an indirect regulation.

The results corresponding to Experiments 2, 3 and 4 are shown in Figure 4.5. From the figure, we observe that for Experiments 2 and 3, GRNCIT infers 6 correct and 7 incorrect arcs (compared to 5 correct and 4 incorrect arcs for Experiment 2, and 2 correct and 5 incorrect arcs for Experiment 3, by BITGRN). For Experiment 4 GRNCIT infers 5 correct and 8 incorrect arcs whereas BITGRN infers 3 correct and 3 incorrect arcs. From the results, we see that for all the datasets, GRNCIT increases the number of correct predictions. However, at the same time, the number of incorrect predictions also increases.

In this chapter, so far, we have proposed and assessed an information theory based hill-climbing algorithm that captures both instantaneous and time-delayed interactions. We have observed from the results that compared to the *only* single-step time-delayed

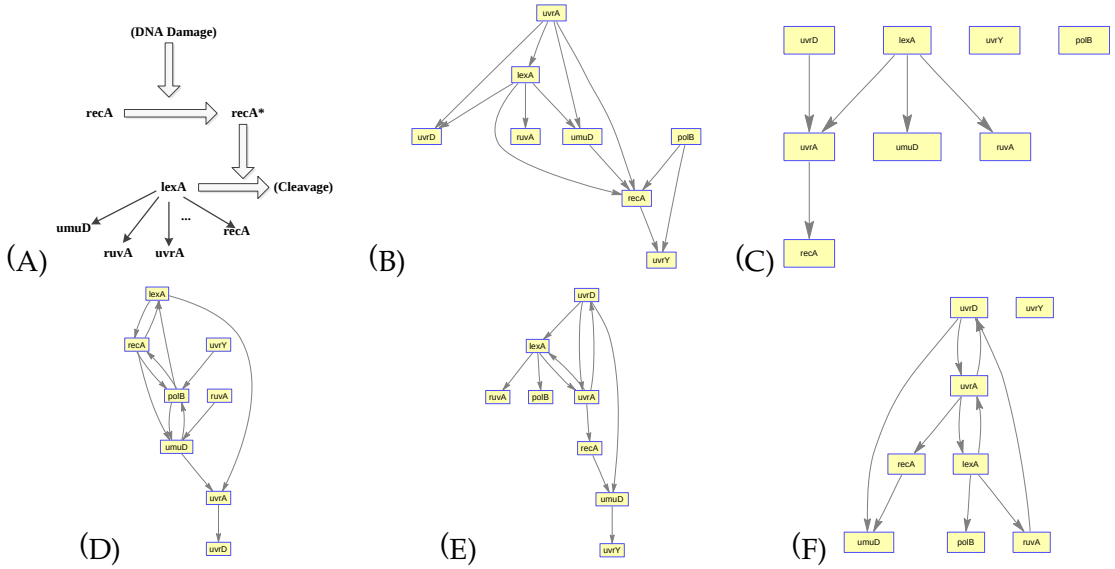

Figure 4.4: Reconstruction of SOS DNA Repair Network (Experiment 1) by GRNCIT. (A) Target Network. (B) Network Inferred by GRNCIT. (C) Network Inferred by TDARACNE. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

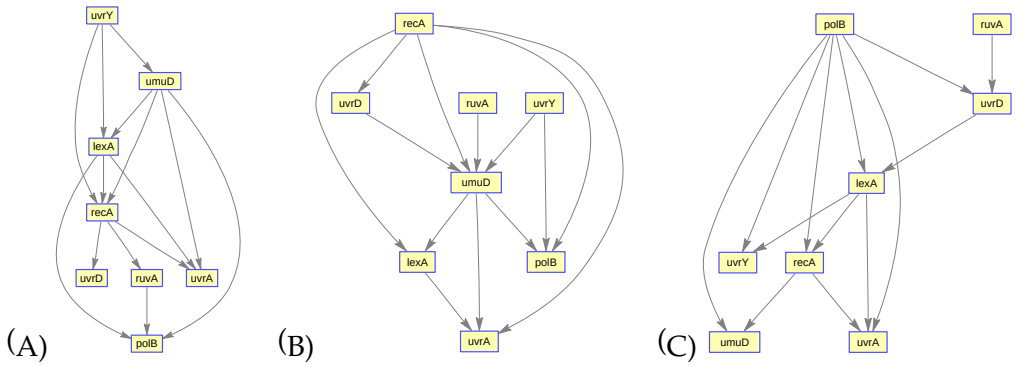

Figure 4.5: Reconstruction of SOS DNA Repair Network by GRNCIT (Experiments 2, 3, 4). (A) Experiment 2. (B) Experiment 3. (C) Experiment 4.

Regulator	Target	correct/ incorrect
lexA	uvrD	correct
	recA	correct
	ruvA	correct
	umuD	correct
recA	uvrY	correct[a]
uvrA	uvrD	incorrect
uvrA	lexA	incorrect
uvrA	umuD	incorrect
uvrA	recA	incorrect
umuD	recA	incorrect
polB	recA	incorrect
polB	uvrY	incorrect

[a] correct considering indirect regulation of RecA through LexA

Table 4.5: Analysis of individual interactions inferred by GRNCIT - SOS DNA Repair Network

regulation approach by BITGRN, it infers a larger number of true positives. However, we see that along with increasing the number of true positives, the number of false positives also increases. Hence, in the following sections of this chapter, we build a *score+search* based algorithm for reconstructing GRNs. We first show how a well known decomposable scoring metric can be adapted to the problem of modeling and capturing both instantaneous and time-delayed interactions. A genetic algorithm (GA) based search strategy using the scoring metric is then proposed that systematically searches for regulators of a gene that may influence it instantaneously or in a time-delayed fashion and scores them, eventually creating a GRN that more realistically represents the underlying biological regulations. The proposed approach is validated by carrying out experiments using both synthetic and real-life data. The comparison with other methods shows the superiority of our proposed approach in discovering meaningful regulatory relationships.

4.5 Genetic Algorithm Based Search

4.5.1 The Scoring Technique

In contrast to greedy/hill-climbing approaches [54, 157], which usually achieve good results for smaller search spaces but usually fail to perform better for larger networks, a *score+search* approach can traverse a larger search space and can provide better results in case of larger networks. There is obviously a trade-off in such a decision. The computational time and complexity increases, both due to repeated scoring and searching. Designing a strategy that can efficiently search through the space thus becomes important. The scoring function should also have some desirable properties, for example, decomposability. We use a modified form of the MIT (Mutual Information Tests) score proposed by de Campos [60], which can work with the representational framework described in the previous section. The MIT score, which is a decomposable scoring metric, relies on the decomposition property of MI (Equation 3.1) and a theorem of Kullback [123]. It is similar to those metrics which are based on maximizing a penalized version of the log-likelihood, such as BIC/MDL. However, the penalty component in this case is specific rather than global for each variable and its parents, and takes into account not only the complexity of the structure but also its reliability. Also, though the *score+search* strategy based on MIT has similarities with learning algorithms based on independence tests, it has an additional strength in that the tests are not only used to decide whether the variables under consideration are independent or not, but they also quantify the extent to which they are. The experimental results in [60] show that MIT systematically outperforms other scores such as the Bayesian scores and that it should be the score of reference within those based on information theory.

Formally, the MIT score is defined as follows:

$$\begin{aligned} g_{MIT}(G:D) &= \sum_{\substack{i=1 \\ Pa(X_i)\neq\phi}}^{n} 2N.MI(X_i, Pa(X_i)) \\ &- \max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha, l_i \sigma_i(j)} \end{aligned} \tag{4.14}$$

where $Pa(X_i)$ refers to the parent of gene X_i in graph G, N is the number of data points and s_i denotes the number of parents of gene X_i. $\sigma_i = (\sigma_i(1), \ldots, \sigma_i(s_i))$ denote any

permutation of the index set $(1, \ldots, s_i)$ of the variables $Pa(X_i) = X_{i1}, \ldots, X_{is_i}$ and $l_{i\sigma_i(j)}$ is defined as follows:

$$l_{i\sigma_i(j)} = \begin{cases} (r_i - 1)(r_{i\sigma_i(j)} - 1) \prod_{k=1}^{j-1} r_{i\sigma_i(k)}, j = 2, \ldots, s_i \\ (r_i - 1)(r_{i\sigma_i(1)} - 1), j = 1 \end{cases} \tag{4.15}$$

The computation of each penalty component (the second term on RHS of Equation 4.14) can be done using the following identity,

$$\underset{\sigma_i}{max} \sum_{j=1}^{s_i} \chi_{\alpha, l_i \sigma_i(j)} = \sum_{j=1}^{s_i} \chi_{\alpha, l_i \sigma_i^*(j)} \tag{4.16}$$

where σ_i^* is any permutation of $Pa(X_i)$ satisfying $r_{i\sigma_i^*(1)} \geq r_{i\sigma_i^*(2)} \geq \cdots \geq r_{i\sigma_i^*(s_i)}$. Equation 4.16 says that the desired permutation of the parents is the one where the first parent has the highest number of states, the second parent has the second highest number of states, and so on. Clearly, this simplifies the computation.

Referring to the assumptions in Section 4.2, it should be made clear here that the scoring should not add up the scores for the same regulations multiple times (which is the usual behavior of a scoring metric). This requirement helps in bringing diversity in the networks. Consider for example, a GRN where gene X has two parents, A and B and A has a stronger regulatory effect on X. Now, consider two candidate solutions. In the first solution, X has both an intra-slice (instantaneous) and an inter-slice (time-delayed) arc from A, with the intra-slice arc carrying stronger regulation. In the second solution, X has an inter-slice arc from A and an inter-slice arc from B (lower in strength than regulation A). Now, if we just add up individual scores, it is obvious that the first network will achieve a better score (the penalty being the same in both cases). This behavior is not desirable; hence we decide that in the case of multiple arcs for the same interaction, only the arc with the highest MI will contribute to the overall score.

4.5.2 The Search Strategy

A genetic algorithm (GA), applied to explore this structure space, begins with a sample population of randomly selected network structures and their fitness calculated. Iteratively, crossovers and mutations of networks within a population are performed and the best fitting individuals of the population are kept for future generations.

During crossover, two random edges are chosen and swapped between them. Mutation is applied on an individual edge of a network. For our study, we incorporate the following three types of mutations:

1. Deleting a random edge from the network.
2. Creating a random edge in the network.
3. Changing direction of a randomly selected edge.

As both the crossover and mutation operations directly affect the structure of the network, the conditions listed in Section 4.2 need to be satisfied whenever an edge is created or manipulated.

In the search process, if for five consecutive generations the best score is not increasing then we aggregate the best 5 networks by taking a majority voting scheme (three out of five). The overall algorithm, called GRNCGA (learning GRNs with Contemporaneous arcs using Genetic Algorithm), that includes the modeling of the GRN and the stochastic search of the network space using GA is shown in Table 4.6.

4.6 Experimental Results for GRNCGA

We evaluate our proposed method by both synthetic network and real-life biological network of *Saccharomyces cerevisiae* (yeast). However, due to the lack of availability of biological information regarding which interactions are instantaneous and which are time-delayed, it is not possible to investigate the performance of the approach on the basis of whether an individual arc is instantaneous or not. As a result, the overall accuracy of the inference method and correctness of the modeling approach is evaluated by the conventional performance measures, i.e., using *Se*, *Sp*, *Pr* and *F*. Similar to the experimental

1. Create initial population of network structures. For each individual, genes and set of parent genes are selected based on a Poisson distribution and edges are created such that the resulting network complies with the conditions listed in Section 4.2.
2. Evaluate each network and sort the chromosomes based on the fitness score.
   (a) Generate new population by applying crossover and mutation on the previous population. Ensure that assumptions listed in Section 4.2 are not violated.
   (b) Sort each individual using the fitness function.
   (c) If the best individual score has not increased for five consecutive generations, aggregate the 5 best individuals using a majority voting scheme. Ensure that assumptions listed in Section 4.2 are not violated.
   (d) Take best individuals from the two populations based on fitness score and create the population of elite individuals for next generation.
3. Repeat steps a) - d) until the stopping criteria (800 generations in our case) is reached.

   When the GA stops, take the best chromosome and reconstruct the final genetic network. ◁

Table 4.6: Genetic algorithm (GRNCGA)

setting we have used throughout, we used the Persist [150] algorithm to discretize continuous data into 3 levels. The confidence level (α) was set to 0.9. The number of individuals in the population was set to 50 for all the experiments. For all the experiments related to synthetic network, we used 3 different datasets for each experiment and combined these 3 datasets using a known procedure described in [222].

4.6.1 Synthetic Network

As a first step towards evaluating the performance of GRNCGA, we consider the 5-gene target network given in Chapter 3, Section 3.3, Figure 3.3. For this network, its initial conditions and the S-system model parameters (g and h) are available in literature [187], [222]. We use the R-K integration method, to obtain the 3 sets of time series data, each having 30 time points. We compare the performance of our method with five other techniques reported earlier, namely, BITGRN [157], DBN(DP) [70], DBN(NPR) [114], BANJO [255] and BNFinder (BDe and MDL) [66]. Referring to Table 4.7, the results for the network having the best performance measure are given in row 1 whereas the average and standard deviation of the measures (corresponding to the 5 different GA runs) is given in row 2. The values corresponding to BITGRN are given in row 3. Rows

4-8 correspond to results obtained from DBN (DP), DBN(NPR), BANJO, BNFinder (BDe) and BNFinder (MDL), respectively.

	Se	***Sp***	***Pr***	***F***
GRNCGA(Best)	0.92	0.77	0.79	0.85
GRNCGA	$\mathbf{0.80}\pm$	$0.79\pm$	$0.78\pm$	$\mathbf{0.79}\pm$
(Average$\pm$ Std Dev)	**0.094**	0.036	0.019	**0.047**
BITGRN	0.67	0.86	0.82	0.74
DBN(DP)	0.5	0.89	0.82	0.62
DBN(NPR)	0.67	0.77	0.73	0.70
BANJO	0.47 ± 0.11	$\mathbf{0.97 \pm 0.04}$	$\mathbf{0.93 \pm 0.09}$	0.62 ± 0.1
BNFinder+BDe	0.27 ± 0.09	0.69 ± 0.12	0.46 ± 0.14	0.33 ± 0.097
BNFinder+MDL	0.35 ± 0.12	0.58 ± 0.09	0.43 ± 0.09	0.38 ± 0.011

Table 4.7: Performance comparison of GRNCGA with BITGRN, DBN (DP), DBN (NPR), BANJO and BNFinder, on the 5-gene Synthetic Network

From the table, we observe that the average values of the sensitivity and F-score of our method are much higher compared to the corresponding values of the other methods. Again, we see that BANJO performs better in terms of specificity and precision, but the other two measures of BANJO are very low. Also, the standard deviations of the performance measures are lower for GRNCGA compared to BANJO, indicating a more consistent performance of the proposed method.

4.6.2 Effect of Number of Samples and Noise

To study the effect of the number of samples, we use the larger 20-node network [163] presented in Chapter 3, Section 3.3, Figure 3.5. We use the same parameters as described in [163] for data generation. The number of samples was varied to observe the effect of the number of samples (20 and 30 samples for each dataset). To study the effect of noise, we added 6 different levels of noise (random Gaussian noise with zero mean and variance, $\sigma^2 = 0, 0.01, 0.02, 0.05, 0.1, 0.2$). Each experiment was repeated 5 times and the averages of these results are shown in Figure 4.6.

From the figure, we can observe that due to the increase in the size of the network (at the same time using the same or fewer samples as in the 5-gene network in the previous subsection), the performance suffers, but not significantly. Also, for a given sample size,

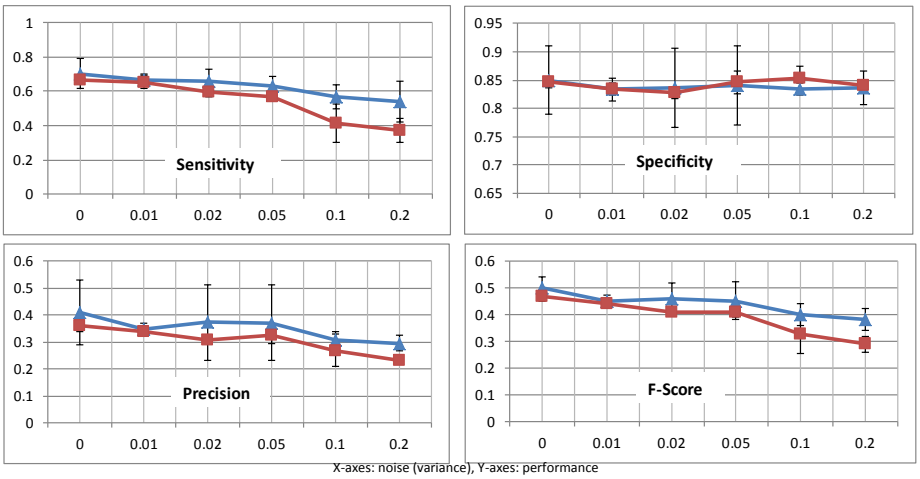

Figure 4.6: Effect of noise and number of samples on the performance of GRNCGA using the 20-node synthetic network. X-axes represent the 6 noise levels (corresponding variance values) used. Y-axes represent the corresponding performance measures. Rectangles - 20 data points (per dataset) experiment, triangles - 30 data points (per dataset) experiment. Vertical lines denote standard deviations.

an increase in noise level does not significantly deteriorate the performance of network reconstruction. We also observe that increasing the number of samples significantly improves the accuracy of the method in network reconstruction. Further, the increase in sample size also aids in better handling of noise. For higher levels of noise, the performance is better with a higher sample size. Moreover, for lower noise levels, the sensitivity, specificity and F-score remain almost unaffected for different sample sizes, indicating the robustness of the method at low noise levels.

4.6.3 Real-Life Biological Data

The first real-life biological network that we consider for investigation is the well known IRMA network already described in previous chapters. In short, for this network there are two sets of gene profiles called Switch ON and Switch OFF, each containing 16 and 21 time series data points, respectively. The former corresponds to the shifting of the growing cells from glucose to galactose medium, the latter to the reverse phase. Some edges in the original network actually represent protein level interactions and are not directly contributing to gene-gene regulation. A 'simplified' network, ignoring these interactions, is also reported in [35]. To compare our reconstruction method, we consider 5 recent methods, namely, BITGRN [157], TDARACNE [261], BANJO [255], ARACNE [139], and

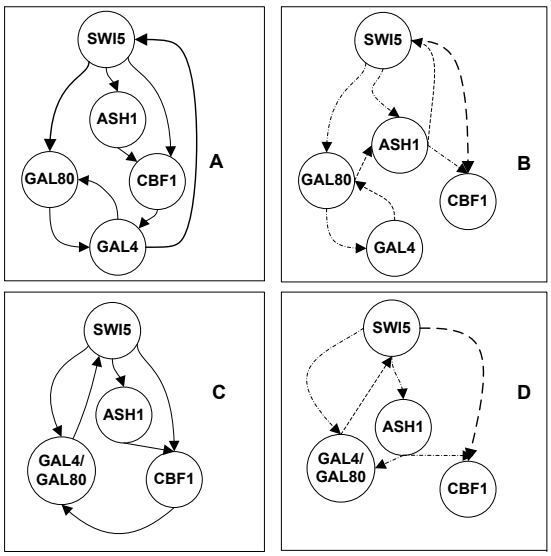

Figure 4.7: Yeast network and inferred network by GRNCGA. (A) Target network. (B) Inferred network by our method (best case). (C) Target network (simplified). (D) Inferred network (simplified), best case. Dashed (–) arcs mean intra-slice (instantaneous) arcs, dash-dotted (-.-) arcs mean inter-slice (time-delayed) arcs.

BNFinder [246], which have previously been used for reconstructing the networks under consideration.

IRMA ON dataset

The results for the ON dataset are shown in Figure 4.7. In the figure, dashed lines mean intra-slice arcs and dash-dotted lines mean inter-slice arcs. Of the total eight arcs in the original network, our method correctly identified six arcs (the best performing network), corresponding to a sensitivity, precision and F-score of 0.75. Four of these were inferred by our method as time-delayed regulations and two were inferred as instantaneously active regulations. The best performing network corresponding to the simplified representation has five correct arcs (out of six), two of them were inferred as instantaneous and the remaining three as time-delayed. The overall performance comparison amongst various methods is shown in Table 4.8. The average and standard deviation correspond to the 5 different runs of the GA. We observe that the overall performance of GRNCGA is satisfactory, with the highest number of true predictions, and high values of the other performance measures.

	Original Network				Simplified Network			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
GRNCGA(Best)	0.75	0.88	0.75	0.75	0.83	0.95	0.83	0.83
GRNCGA	**0.73**±	0.83±	0.68±	0.70±	**0.80**±	0.90±	0.73±	0.76±
(Average± Std Dev)	**0.054**	0.027	0.044	0.044	**0.072**	0.027	0.06	0.058
BITGRN	0.63	**0.94**	**0.83**	**0.71**	0.67	**1**	**1**	**0.80**
TDARACNE	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
ARACNE	0.60	-	0.50	0.54	0.50	-	0.50	0.50
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50
BNFinder+BDe	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BNFinder+MDL	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22

Table 4.8: Performance comparison of GRNCGA based on IRMA ON dataset

IRMA OFF dataset

As already stated, the overall performances of all the algorithms suffer to some extent for the OFF dataset, due to the lack of 'stimulus'. The comparison amongst various methods is shown in Table 4.9. As we can see, for the simplified network, BITGRN and BANJO perform comparatively better. However, for the original network, the F-score, sensitivity and precision are higher compared to the other methods, thereby outperforming them. Specificity value, although not the best in all cases, is also quite satisfactory, implying that the inference of false positives is not high.

4.6.4 SOS DNA Repair Network of *E. coli*

We also analyze the SOS DNA Repair Network in *Escherichia coli*, presented in Section 3.5.2. As discussed, the network is controlled by a complex circuitry involving the RecA and LexA proteins [149] as master repressors and sensors, respectively, which act as the 'hub' genes for the network. We used the same four datasets used in the previous chapter for the reconstruction of this network (from Uri Alon Lab [8]). As usual, the eight genes, namely uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB, were investigated for the experiment.

	Original Network				Simplified Network			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
GRNCGA(Best)	0.75	0.82	0.67	0.71	0.83	0.89	0.71	0.77
GRNCGA	**0.65**±	0.78±	**0.60**±	**0.62**±	0.77±	0.87±	0.66±	0.71±
(Average± Std Dev)	**0.054**	0.033	**0.051**	**0.052**	0.088	0.027	0.059	0.065
BITGRN	0.63	0.76	0.56	0.59	**0.83**	0.89	**0.71**	**0.77**
TDARACNE	0.60	-	0.37	0.46	0.75	-	0.50	0.60
ARACNE	0.33	-	0.25	0.28	0.60	-	0.50	0.54
BANJO	0.38	**0.88**	**0.60**	0.46	0.33	**0.90**	0.67	0.44
BNFinder+BDe	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BNFinder+MDL	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40

Table 4.9: Performance comparison of GRNCGA based on IRMA OFF dataset

The results corresponding to Experiment 1 are presented in Figure 4.8(B). Along with our result, we include the results from BANJO, TDARACNE and BNFinder in Figure 4.8(C)-(F) and the target network in 4.8(A). The results corresponding to the other experiments are shown in Figure 4.9. As already stated, the exact ground truth for this network is not precisely known, and hence it is not possible to calculate the well known performance measures. Instead, using the known interactions obtained from the literature [116, 163], an analysis of correct and incorrect predictions by our method is obtained and shown in Table 4.10.

Considering results corresponding to Experiment 1, we observe that our method correctly identifies lexA and recA as the 'hub' genes for this network. Also, we observe that five interactions inferred by GRNCGA are correct. It successfully infers lexA as the regulator of uvrA, uvrD and umuD. Moreover, considering the indirect regulation of RecA through LexA, two more interactions, namely recA→uvrY and recA→polB can also be considered correct. In contrast, 3 of the 5 interactions identified by TDARACNE [261] are correct. Most of the interactions identified by BNFINDER+BDe and BNFinder+MDL are incorrect (4 correct versus 6 incorrect for the MDL approach, and 4 correct versus 7 incorrect for the BDe based approach). Both these approaches successfully identify regulation of ruvA, polB and uvrA by lexA. In addition, the regulation of umuD by recA can also be considered correct. However, compared to these methods, GRNCGA infers the highest

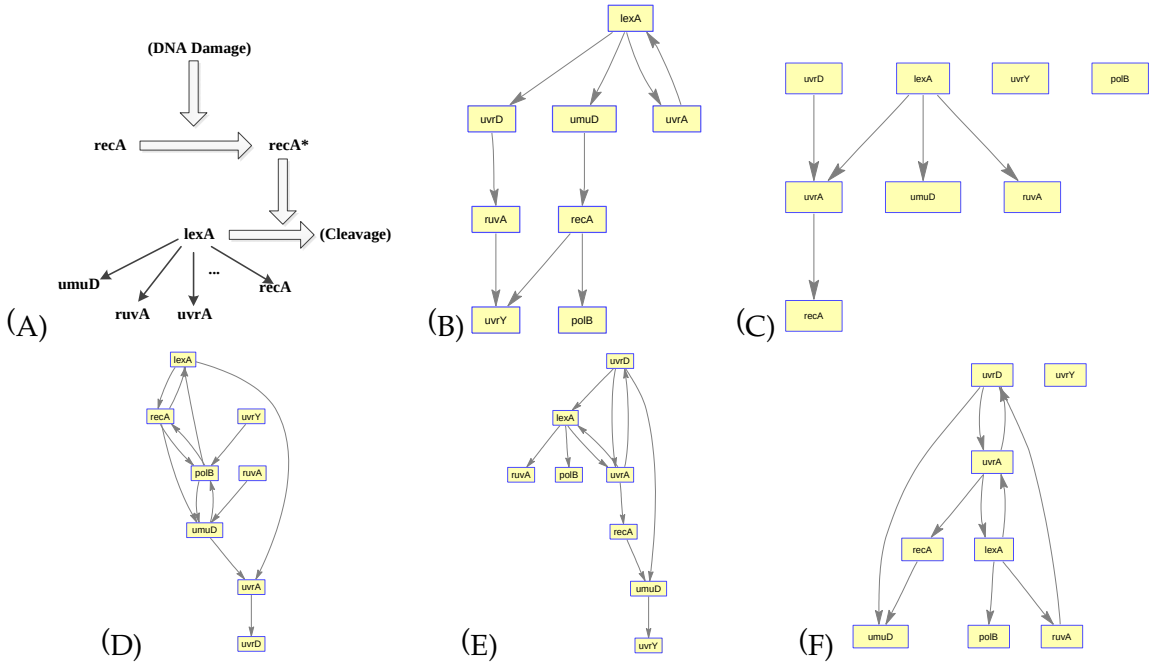

Figure 4.8: Reconstruction of SOS DNA Repair Network (Experiment 1). (A) Target Network. (B) Network Inferred by GRNCGA. (C) Network Inferred by TDARACNE. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

number of correct predictions (same as BANJO, although BANJO infers 8 incorrect arcs). The number of incorrect predictions is also low (4) for our method.

Next, considering the results corresponding to the other experiments (Figure 4.9), we observe that for Experiment 2, GRNCGA infers 5 correct arcs and 6 incorrect arcs. As a comparison, GRNCIT infers 6 correct arcs and 7 incorrect arcs for this dataset, and BITGRN infers 5 correct and 4 incorrect arcs. For Experiment 3, GRNCGA infers 6 correct and 4 incorrect arcs (compared to 6 correct and 7 incorrect for GRNCIT, and 2 correct and 5 incorrect for BITGRN). Finally, GRNCGA infers 6 correct and 8 incorrect arcs whereas GRNCIT infers 5 correct and 8 incorrect arcs, and BITGRN infers 3 correct and 3 incorrect arcs. When we compare these results with those from BITGRN and GRNCIT, we observe a trend that with the incorporation of instantaneous interactions, the number of correct interactions increases (as seen from GRNCIT). However, at the same time, the number of incorrect predictions also increases in GRNCIT. The *score+search* strategy employed by GRNCGA checks the incorrect predictions by GRNCIT to some extent.

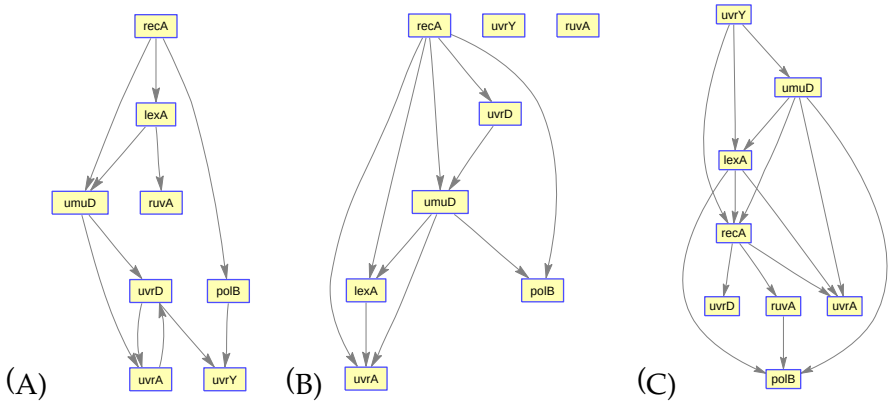

Figure 4.9: Reconstruction of SOS DNA Repair Network by GRNCGA (Experiments 2, 3, 4). (A) Experiment 2. (B) Experiment 3. (C) Experiment 4.

Regulator	Target	correct/ incorrect
lexA	uvrD	correct
	umuD	correct
	uvrA	correct
recA	polB	correct[a]
	uvrY	correct[a]
uvrA	lexA	incorrect
uvrD	ruvA	incorrect
umuD	recA	incorrect
ruvA	uvrY	incorrect

[a] correct considering indirect regulation of RecA through LexA

Table 4.10: Analysis of individual interactions inferred by GRNCGA - SOS DNA Repair Network

4.7 Summary

In this chapter, we proposed a DBN based modeling framework that can represent both instantaneous and single-step time-delayed regulations among genes. We used the framework with two different learning approaches: a greedy hill-climbing based approach using information theoretic quantities, and also a *score+search* based GRN reconstruction algorithm, such that both these approaches can model both instantaneous and time-delayed interactions among genes. The final reconstructed networks using our proposed modeling scheme show better performance compared to other methods. This indicates that our method is a more biologically relevant approach compared to both, methods that do (*i*) only instantaneous and (*ii*) only time-delayed interaction modeling. Also, to improve confidence on the ability of the methods to reconstruct meaningful GRNs, we have analyzed different networks using both synthetic data and real-life biological data of yeast and *E. coli*. It was observed that increase in the number of samples helped to maintain accuracy of network reconstruction. By applying the technique to networks containing noisy data, we showed the robustness of the methods in the presence of noise.

Although both instantaneous and time-delayed interactions have been considered in this chapter, the time-delays are considered to be single-step delays. Further, the learning has been carried out in a sequential manner. The next chapter is devoted to exploring these issues.

Chapter 5

Joint Learning of Instantaneous and Multi-Step Time-Delayed Interactions

5.1 Introduction

Realistic and accurate reconstruction of gene regulatory networks is very crucial for correct understanding and interpretation of genetic interactions. Usually, modeling of gene regulations using the BN or DBN formalism has taken the view that genes interact either instantaneously or with a certain time delay. In the previous chapter, we elaborated the point that in biological systems both instantaneous and time-delayed interactions occur, and thus it is vital to model both these types of interactions in a single modeling framework. We also showed the effectiveness of the modeling framework by using two different learning strategies which learn these interactions - albeit sequentially. However, since biological regulations, both instantaneous and time-delayed, occur simultaneously in various living organisms, we need to develop a learning framework that is able to learn both these types of interactions *simultaneously*. This would result in a more accurate representation of gene regulatory networks.

In this chapter, we primarily focus on improving the learning strategy for GRN reconstruction. In this process, we also extend the modeling framework proposed in the previous chapter to model multi-step time-delayed interactions. To develop a learning

strategy that evaluates both instantaneous and time-delayed interactions between genes *jointly*, we propose a novel scoring metric having firm mathematical underpinnings that, unlike other recent methods, can score both types of interactions concurrently and takes into account the reality that multiple regulators can regulate a gene jointly, rather than in an isolated pair-wise manner. Due to the nature of the scoring metric, while reconstructing the network using time series data, the learning framework that we propose needs special alignment mechanism, which is also elaborated in this chapter. Further, a gene regulatory network (GRN) inference method employing a local search that makes use of the modeling framework and the scoring metric is also presented.

Due to the extension of the approach to multi-step time-delayed interactions, the already large search space becomes even larger and multi-modal. Exploring the search space using simple hill climbing or basic genetic algorithm based approaches runs the risk of not being able to achieve the globally optimal target. For this reason, subsequently, we also propose a novel two-stage genetic algorithm that uses frequently occurring subgraphs (called network motifs) in the first stage, to detect the common patterns evident in different optimal solutions, and then combines these patterns in the second stage to obtain an optimal solution. Although evolutionary algorithms, in general, cannot guarantee global optima, our design allows us to explore a much larger search space compared to the basic GA based approaches, thereby having a much higher probability of obtaining a better solution. Using the scoring function proposed for joint learning, we apply the algorithm to both synthetic networks and real-life networks of *E. coli* to show the effectiveness of the algorithm.

The rest of the chapter is organized as follows. In Section 5.2, we provide the reasoning behind extending the modeling framework to multi-step time-delayed interactions, and its differences from other methods (these methods are related to our approach, but not directly comparable to our objective or what we achieve). Section 5.3 shows the modified representational framework, explains the scenarios that may complicate the use of the framework, and provides examples to clarify the usage. Section 5.4 proposes the scoring metric called CCIT (Combined Conditional Independence Tests) which takes full advantage of the proposed representational framework. Experimental evaluation and discussion of the proposed approach are presented in Section 5.5. Finally, from Section 5.8 through to Section 5.10, we present and evaluate the mDBN approach which uses the

concept of network motifs that can explore a much larger search space compared to naive GA search, thereby encouraging convergence to the near-optimal solution. Section 5.11 concludes the chapter.

5.2 Modeling Multi-Step Time-Delayed Interactions

In the previous chapter, we considered only single-step time-delayed interactions (along with instantaneous interactions) for modeling GRNs. The algorithms proposed to capture these interactions showed good performance, indicating that there are indeed various type of interactions occurring in biological systems. However, in terms of biology, modeling only single-step time-delayed interactions (along with instantaneous interactions) means that all genes which regulate their target genes with a single-step time-delay take almost the same amount of time for their regulatory relation to take effect. This assumption is restrictive, because different genes may take different amounts of time for the effect of regulation to take place. In fact, as has been pointed out in several articles [58, 248, 262], the timing of regulations can vary widely among different regulators, and thus it is appropriate to allow different regulations to have different time delays (see Figure 5.1).

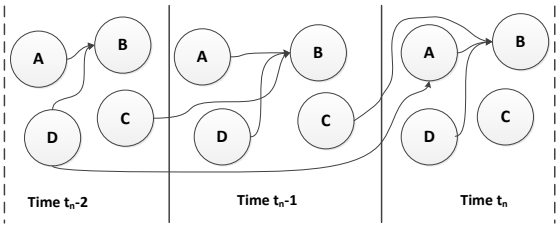

Figure 5.1: Network structure with both instantaneous and multi-step time-delayed interactions

To the best of our knowledge, prior works on inter-slice (equivalent to time-delayed interactions in GRNs) and intra-slice (instantaneous interactions) connections in the dynamic probabilistic network formalism [59, 77] have modelled a DBN using an initial network and a transition network employing the first-order Markov assumption, where the initial network exists only during the initial period of time and subsequently the dynamics is expressed using only the transition network. Realising that a d-th order DBN

has variables replicated d times [70], a first-order DBN for this task[1] is therefore usually limited to around 10 variables. If a second-order DBN model is chosen, it can mostly deal with 6-7 variables [70]. Since our proposed approach does not replicate variables, we can study any complex network configuration without limitations on the number of nodes, unlike the 'replicate layers for each order' approach of Eaton *et al.* [70]. Zou *et al.* [262], while highlighting the existence of both instantaneous and time-delayed interactions among genes while considering the parent-child relationships of a particular order, did not account for the regulatory effects of other parents (having a different order of regulation than the current one) on that particular child. This is in violation of the biological reality that parents with various orders of regulation can jointly regulate a child. Our proposed learning method supports multiple parents to regulate a child simultaneously, with different orders of regulation. Moreover, the limitation of detecting genetic interactions such as $A \leftrightarrow B$, which are prevalent in genetic networks [37], is also overcome in the proposed method. We present the modified representation (modeling) framework in the next section. Furthermore, with this extension, the interpretation and the use of the framework becomes more complex, and these are also elaborated in the next section.

5.3 The Modified Representational Framework

Let us model a gene network containing n genes (denoted by $X_1, X_2 \ldots, X_n$) with a corresponding microarray dataset having N time points. A basic DBN based GRN reconstruction method would try to find associations between genes X_i and X_j by taking into consideration the data $x_{i1}, \ldots, x_{i(N-\delta)}$ and $x_{j(1+\delta)}, \ldots, x_{jN}$ or vice versa (small case letters mean data values in the microarray), where $1 \leq \delta \leq d$. That is, it will take into consideration the d-th order Markov rule, for a gene having a maximum order of regulation d with its parents. This will effectively enable this model to capture at most d-step time-delayed interactions. Conversely, a basic BN based strategy would use the entire set of N time points and it will capture regulations that are effective instantaneously.

Now, to represent both instantaneous and multiple step time-delayed interactions, we consider an adjacency matrix based structure as shown in Figure 5.2, which is an extended version of the representation framework proposed in Section 4.2. The zero

[1] A tutorial can be found at http://www.cs.ubc.ca/~murphyk/Software/BDAGL/dbnDemo_hus.htm.

entries in the figure denote no regulation. For the first n columns, the entries marked by 1 correspond to instantaneous regulations whereas for the last n columns non-zero entries denote the order of regulation. As an example, the entry 1 in the cell (X_1, X_2) means X_1 has (almost) instantaneous regulatory effect on X_2. Similarly, the entry d in the cell $(X_n, X_2^{'})$ means X_n regulates X_2 with a d-step time delay. Using this representation, we do not need to replicate layers of interactions for each increment in the order of regulations, making it efficient and particularly suitable for representing GRNs, where higher-order regulations are quite common.

	X_1	X_2	...	X_n	$X_1^{'}$	$X_2^{'}$	...	$X_n^{'}$
X_1	0	1	...	0	2	0	...	1
X_2	0	0	...	1	d	0	...	0
...	...	...	...	...	...	...	...	...
X_n	0	0	...	0	0	d	...	1

Figure 5.2: The updated adjacency matrix for the representation of instantaneous and multiple-step time-delayed interactions

Complications in the alignment of data samples can arise if the parents have different orders of regulation with the child node. To clarify, we describe an example where we have already assessed the degree of interest in adding two parents (say genes B and C, having third and first order regulations, respectively) to the gene under consideration, X. Now, we want to assess the degree of interest in adding gene A as a parent of X with a second order regulatory relationship; that is, we want to compute[2] $MI(X, A^2|\{B^3, C^1\})$, where superscripts on the parent variables denote their order of regulation with the child node.

There are two possibilities to consider. The first corresponds to a scenario where the time series data is not periodic. In this case, we cannot use all the N samples for MI computation, rather we are restricted to using $(N - \delta)$ samples where δ is the maximum order of regulation that the gene under consideration has, with its parent nodes (3 in this case). Figure 5.3 shows how the alignment of the samples can be done for the current example. In the figure, we have N samples and since $\delta = 3$, we can effectively use $(N-3)$ samples. The $\surd$ symbol inside a cell denotes that this data sample will be used for MI

[2]Throughout this thesis, we use Mutual Information (MI)/log-likelihood based Conditional Independence tests for analysis of regulatory interactions.

	1	2	3	4	...	N-3	N-2	N-1	N
A		√	√	√	...	√	√		
X				√	...	√	√	√	√
B	√	√	√	√	...	√			
C			√	√	...	√	√	√	

Figure 5.3: Sample points used for the calculation of the Mutual Information (MI)

computation, whereas empty cells denote that these data samples will not be considered for computing the MI. Similar alignments will need to be done for the other case, where the data is considered to be periodic (e.g., datasets of yeast compiled by Cho *et al.* [49] show such cyclic behavior [248]). However, we can use all the N data samples in this case, where the data is shifted in a circular manner.

The interpretation of the results obtained from an algorithm that uses this framework can be done in a straightforward manner. Using this framework and the aligned data samples, if we construct a network where we observe, for example, arc $X_1 \rightarrow X_n^{'}$ having order δ, we conclude that the time-delayed interaction between X_1 and X_n is inferred and X_1 regulates X_n with a δ-step time delay. Similarly, if we find an arc $X_2 \rightarrow X_n$, we say that the instantaneous interaction between X_2 and X_n is inferred and a change in the expression level of X_2 will almost immediately effect the expression level of X_n. Finally, to ensure consistency in the resulting Bayesian networks, the following four assumptions must also be followed by any resulting DBN structure:

Assumption 5.3.1. ***DAG Constraint.*** *The intra-slice portion of the network must be a DAG (directed acyclic graph). In other words, since the left most n-columns are representative of the Bayesian network (instantaneous interactions), they must correspond to a network with no cycles.*

The second assumption ensures that information flow goes in the correct direction:

Assumption 5.3.2. ***Information flow constraint.*** *The inter-slice arcs must not contain any backward directed arcs. All arcs from genes in a particular time point must must be directed towards genes in a later time point with reference to the current gene under consideration.*

The stationarity assumption is also necessary for the BN based reconstruction techniques:

Assumption 5.3.3. ***Stationarity.*** *The transition probability $P(\boldsymbol{X}[t]|\boldsymbol{X}[t-1])$ is independent of t. That is, interactions remain existent irrespective of time.*

Finally, we need to make the d-th order Markov assumption to work with d-step time-delayed interactions:

Assumption 5.3.4. ***d-th Order Markov Assumption.*** *The d-th order Markov assumption states that given the current observation $\boldsymbol{X}[t]$, the next observation $\boldsymbol{X}[t+1]$ is independent of all the past observations until the most recent d observations, $\boldsymbol{X}[t-1], \ldots, \boldsymbol{X}[t-d]$. Formally, the d-th order Markov Property can be defined as follows:*

$$P\left(\boldsymbol{X}[t]|\boldsymbol{X}[t-1], \ldots, \boldsymbol{X}[0]\right) = P\left(\boldsymbol{X}[t]|\boldsymbol{X}[t-1] \ldots \boldsymbol{X}[t-d]\right) \tag{5.1}$$

where $d \leq t$.

The joint probability distribution and the transition probabilities of this model are defined in a similar manner as discussed in Section 4.2 of the previous chapter.

5.4 The Proposed Scoring Metric, CCIT

We share the same idea with MIT (Mutual Information Tests) [60] and MDL (the Minimum Description Length principle) for developing a scoring metric that can score both instantaneous and time-delayed interactions simultaneously: to use the MI/log-likelihood measure between a node X, and its parents, $Pa(X)$, for measuring the degree of association between them, and penalizing the structural complexity. The first part aims at minimizing the Kullback-Leibler (KL) divergence between the joint distribution corresponding to the original network (p_D) and the graph under consideration (p_G), according to the following equation:

$$\underset{G \in G_n}{\operatorname{argmin}}\ KL(p_D, p_G) = \underset{G \in G_n}{\operatorname{argmax}} \sum_{\substack{i=1 \\ Pa_G(X_i) \neq \phi}}^{n} MI(X_i, Pa_G(X_i)) \tag{5.2}$$

which is equivalent to maximizing the log-likelihood (i.e., the higher the MI/log-likelihood score, the better the network). In our approach, calculation of the MI/log-likelihood score

is done in a manner which is similar to the approaches in MIT/MDL, with a major difference: calculation of score (using MI/log-likelihood) in the presence of joint regulation. To make the notion clear consider Figure 5.1. Using MIT, the MI part for scoring for gene B is[3] $MI(B, \{A^0, D^0\}) + MI(B, C^1)$ (similar calculations of log-likelihood are used for MDL). As we can see, the calculation of MI/log-likelihood for the zero-order interactions do not take into account the parents who regulate it with time delay. Unlike the approach in basic MIT and other approaches where zero and higher-order interactions are scored separately and then combined, in our approach, we also condition (during computation) on those parents which have different orders of regulation with the target gene. The marginal probability for each node of this model thus becomes:

$$P(\mathbf{X}[t]|\mathbf{X}[t-1], \ldots, \mathbf{X}[t-d]) = \prod_{i=1}^{n} P(X_i[t]|Pa(X_i[t])) \tag{5.3}$$

The term $Pa(X_i[t])$ in the above equation represents the parents of gene X_i at time t, which can be in the same time-slice or in one of the d previous time-slices (d is the maximum order of regulation) of gene X_i at time t. Thus, using our approach, the scoring function for B will calculate $MI(B, \{A^0, D^0\} \cup \{C^1\})$. Scoring in this manner enables us to score both instantaneous and time-delayed interactions simultaneously, rather than considering these two types of interactions in an isolated manner, making it especially suitable for problems like reconstructing GRNs, where occurrence of joint regulation is a common phenomenon.

The idea of penalizing complex structures is ubiquitous, finding its place in most of the scores like BIC, MIT and MDL. The penalization component for BIC and MDL are global, whereas for MIT it is specific for each variable and its parents. Being local in nature, the MIT scheme usually outperforms the other two [60]. In this scheme, the localised penalty is based on a theorem of Kullback [123], which says that for a particular confidence level α, the quantity $2N.MI(X_i, X_j|Pa(X_i)) - \chi_{\alpha,l_{ij}}$ represents a statistical test of conditional independence, where l_{ij} is the degrees of freedom of a chi-squared distribution, and $\chi_{\alpha,l_{ij}}$ is the statistical significance threshold. The more positive the value is, the more likely is that X_i and X_k are related (given the current parent set, $Pa(X_i)$) and vice

[3]It should be noted here that MIT/MDL are basic scoring metric for BNs, which can be extended to score both Static and Dynamic BNs separately. Here, we are discussing MIT/MDL applied to a network having both zero and higher-order interactions

versa. Thus, adding up the MI quantities for all the genes (multiplied by 2*number of samples) and subtracting the corresponding local penalization measures effectively constitute a series of conditional independence (CI) tests, and this scheme is used for scoring using MIT.

However, porting this idea of local penalization directly to a gene regulatory network which suffers with dimensionality problem, has the problem of over-penalization. This can be exemplified using Figure 5.1. The penalty component for gene B according to MIT, will be: $\chi_{\alpha,4} + \chi_{\alpha,12} + \chi_{\alpha,36}$, assuming the special case where we have 3 levels of discrete data (the details of how these penalization components can be computed will be shown later). For a Bayesian network design having thousands of samples available, this penalization is not a problem. However, for GRN reconstruction with samples ranging between 20-50, this penalization is too high. To remedy this situation, we propose to apply the penalization only on a per-order of regulation basis. Using this modified scheme, the penalization will be $2\chi_{\alpha,4} + \chi_{\alpha,12}$, which constitutes considerable savings, thereby obtaining a better prediction ratio (in terms of sensitivity and specificity).

The approaches described above are summarised as a scoring metric, named CCIT (Combined Conditional Independence Tests) in Equation 5.4. The score, when applied to a graph G containing n genes (denoted by $X_1, X_2 \ldots, X_n$), with a corresponding microarray dataset D, can be expressed as:

$$S_{CCIT}\left(G:D\right) = \sum_{\substack{i=1 \\ Pa(X_i)\neq\phi}}^{n} \left\{ 2N_{\delta_i}.MI(X_i, Pa(X_i)) - \sum_{k=0}^{\delta_i} (\max_{\sigma_i^k} \sum_{j=1}^{s_i^k} \chi_{\alpha, l_i \sigma_i^k(j)}) \right\} \tag{5.4}$$

Here s_i^k denotes the number of parents of gene X_i having a k step time-delayed regulation and δ_i is the maximum time delay that gene X_i has with its parents. The parent set of gene X_i, $Pa(X_i)$ is the union of the parent sets of X_i having zero time-delay (denoted by $Pa^0(X_i)$), single-step time-delay (denoted by $Pa^1(X_i)$) and up to parents having the maximum time-delay (δ_i). This is defined as follows:

$$Pa(X_i) = Pa^0\left(X_i\right) \cup Pa^1\left(X_i\right) \cdots \cup Pa^{\delta_i}\left(X_i\right) \tag{5.5}$$

The number of *effective* data points, N_{δ_i}, depends on whether the data can be considered to be showing periodic behavior or not (e.g., datasets from Cho *et al.* [49] can be considered as showing periodic behavior [248]). In the case of aperiodicity, N_{δ_i} is determined by subtracting, from the total length of the time profile (N), the maximum order of the time-delay that the gene under consideration has with its parents (δ_i).

$$N_{\delta_i} = \begin{cases} N & \text{if data is periodic} \\ N - \delta_i & \text{otherwise} \end{cases} \tag{5.6}$$

Finally, $\sigma_i^k = (\sigma_i^k(1), \ldots, \sigma_i^k(s_i^k))$ denote any permutation of the index set $(1, \ldots, s_i^k)$ of the variables $Pa^k(X_i)$ and $l_{i\sigma_i^k(j)}$, the degrees of freedom, is defined as follows:

$$l_{i\sigma_i^k(j)} = \begin{cases} (r_i - 1)(r_{\sigma_i^k(j)} - 1) \prod_{m=1}^{j-1} r_{\sigma_i^k(m)}, \text{ for } 2 \leq j \leq s_i^k \\ (r_i - 1)(r_{\sigma_i^k(1)} - 1), \text{ for } j = 1 \end{cases} \tag{5.7}$$

where r_p denotes the number of possible values that gene X_p can take (after discretization, if the data is continuous). If the number of possible values that the genes can take is not the same for all the genes, the quantity σ_i^k denotes the permutation of the parent set $Pa^k(X_i)$ where the first parent gene has the highest number of possible values, the second gene has the second highest number of possible values and so on.

5.4.1 Some Properties of CCIT Score

In this section, we present several useful properties of the proposed scoring metric. The first among these is the decomposability property, which is especially useful for local search algorithms:

Proposition 5.4.1. *CCIT is a decomposable scoring metric.*

Proof. This proposition is evident as the scoring function is, by definition, a sum of local scores. □

Next, we show in Theorem 5.4.1 below that CCIT takes joint regulation into account while scoring and it is different from three related approaches, namely MIT [60] applied

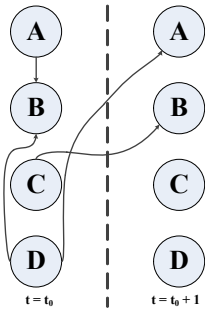

Figure 5.4: Network used for theorem 5.4.1

to: a Bayesian Network (which we call MIT_0); a dynamic Bayesian Network (called MIT_1); and also a naive combination of these two, where the intra-slice and inter-slice networks are scored independently (called MIT_{0+1}). For this, we make use of the decomposition property of MI, defined next:

Property 5.4.1. *(Decomposition Property of MI) In a BN, if $Pa(X_i)$ is the parent set of a node X_i, and the cardinality of the set is s_i, the following identity holds [60]:*

$$MI(X_i, Pa\,(X_i)) = MI\,(X_i, X_{i1}) + \sum_{j=2}^{s_i} MI\left(X_i, X_{ij} | \left\{X_{i1}, \ldots, X_{i(j-1)}\right\}\right) \tag{5.8}$$

Theorem 5.4.1. *CCIT scores intra-slice and inter-slice arcs concurrently, and is different from MIT_0, MIT_1 and MIT_{0+1}.*

Proof. We prove by showing a counter example, using the network in Figure 5.4. We apply our metric along with the three other techniques on the network, describe the working procedure in all these cases to show that the proposed metric indeed scores them concurrently, and finally show the difference from the other three approaches. The network in Figure 5.4 has 4 interactions, 2 of these are instantaneous and 2 are time-delayed (with $\delta = 1$). We assume a non-trivial case where the data is supposed to be periodic (the proof is trivial otherwise). Also, we assume that all the gene expressions were discretized to 3 quantization levels.

1. Application of MIT in a BN based framework:

$$s_{MIT_0} = 2N.MI(B, \{A^0, D^0\}) - (\chi_{\alpha,4} + \chi_{\alpha,12}) \tag{5.9}$$

2. Application of MIT in a DBN based framework:

$$s_{MIT_1} = 2N\{MI(B, C^1) + MI(A, D^1)\} - 2\chi_{\alpha,4} \tag{5.10}$$

3. A naive application of MIT in a combined BN and DBN based framework:

$$s_{MIT_{0+1}} = 2N\{MI(B, \{A^0, D^0\}) + MI(B, C^1) + MI(A, D^1)\} - (3\chi_{\alpha,4} + \chi_{\alpha,12}) \tag{5.11}$$

4. Our proposed scoring metric:

$$s_{CCIT} = 2N\{MI(B, \{A^0, D^0\} \cup \{C^1\}) + MI(A, D^1)\} - (3\chi_{\alpha,4} + \chi_{\alpha,12}) \tag{5.12}$$

The concurrent scoring behavior of CCIT is evident from the first term in RHS of 5.12. Also, the inclusion of C in the parent set in the first term of the RHS of the equation exhibits the manner by which it achieves the objective of taking into account the biological fact that multiple regulators may regulate a gene jointly.

Considering 5.9 and 5.10, it is also obvious that CCIT is different from both MIT_0 and MIT_1. To show that CCIT is different from MIT_{0+1}, we consider 5.11 and 5.12. It suffices to consider whether $MI(B, \{A^0, D^0\}) + MI(B, C^1)$ is different from $MI(B, \{A^0, D^0\} \cup \{C^1\})$. Using 5.8, this becomes equivalent to considering whether $MI(B, \{A^0, D^0\}|C^1)$ is the same as $MI(B, \{A^0, D^0\})$, which are clearly inequal. This completes the proof. □

5.5 Experimental Results Using the CCIT Metric

We evaluate our proposed method by studying both synthetic networks and real-life biological networks of *Saccharomyces cerevisiae* (yeast), *E. coli* and cyanobacteria. The overall accuracy of the inference method and correctness of the modeling approach is evaluated by the four widely used performance measures introduced in Chapter 3, namely, *Se*, *Sp*, *Pr* and *F*. Since our method uses discrete data for the statistical significance tests embedded in the scoring function, we applied the Persist [150] algorithm to discretize the data into 3 levels. The confidence level (α) is set to 0.9. We will use a local search in the DAG space with the classical operators of arc addition, arc deletion and arc reversal. The

starting point of the search is always an empty graph. The parameters for all the other methods that are used for comparison are set to their default values given in their user manuals.

5.5.1 Synthetic Network

Synthetic network using differential equation based models

For performing studies using synthetic networks, we generated 3 random networks of size 10, 25 and 50 using the GeneNetWeaver (GNW) tool [196]. This tool has been used to generate in silico benchmarks in the DREAM (both DREAM3 [172] and DREAM4 [173]) challenge initiative. The tool is able to obtain biologically plausible network topologies (and also biologically plausible network dynamics) of a given size by extracting random sub-networks of *Saccharomyces cerevisiae* and *E. coli* [138, 196]. We used the tool to generate time series data as in the DREAM4 challenge with ten different perturbations for each experiment. Initial and final timestamps for the simulations were 0 and 1000, respectively, and the time step was 50. One of the objectives of this experiment was to test the usefulness of the proposed approach in the presence of noise in mRNA expression levels. We experimented under various noise levels that are likely to be present in the expression data. To mimic a real-life noisy environment, as in Noman *et al.* and Kimura *et al.* [116, 163], we added 5 different noise levels to the data samples (random Gaussian noise with zero mean and variance, $\sigma^2 = 0.0, 0.01, 0.02, 0.05, 0.10$). The performance, measured by the four performance measures, corresponding to the three different sized networks is reported in Figure 5.5. Figure 5.5(A) shows the performance variation as a function of network size and noise level. The X-axes represent the noise levels while the Y-axes represent the corresponding performance measures (*Se, Sp, Pr, F*). In Figure 5.5(B)-(D), we compare our approach with three other methods, namely TDARACNE, BANJO and BNFinder (BDe and MDL) using the F-score (results corresponding to other measures are provided in Appendix B). It is evident from the results that there is no clear winner in all the cases. Some methods perform well in some cases, while others outperform it in other cases. However, it is clear that our proposed approach, albeit not always the best, it is always among the top performers and has consistently superior performance.

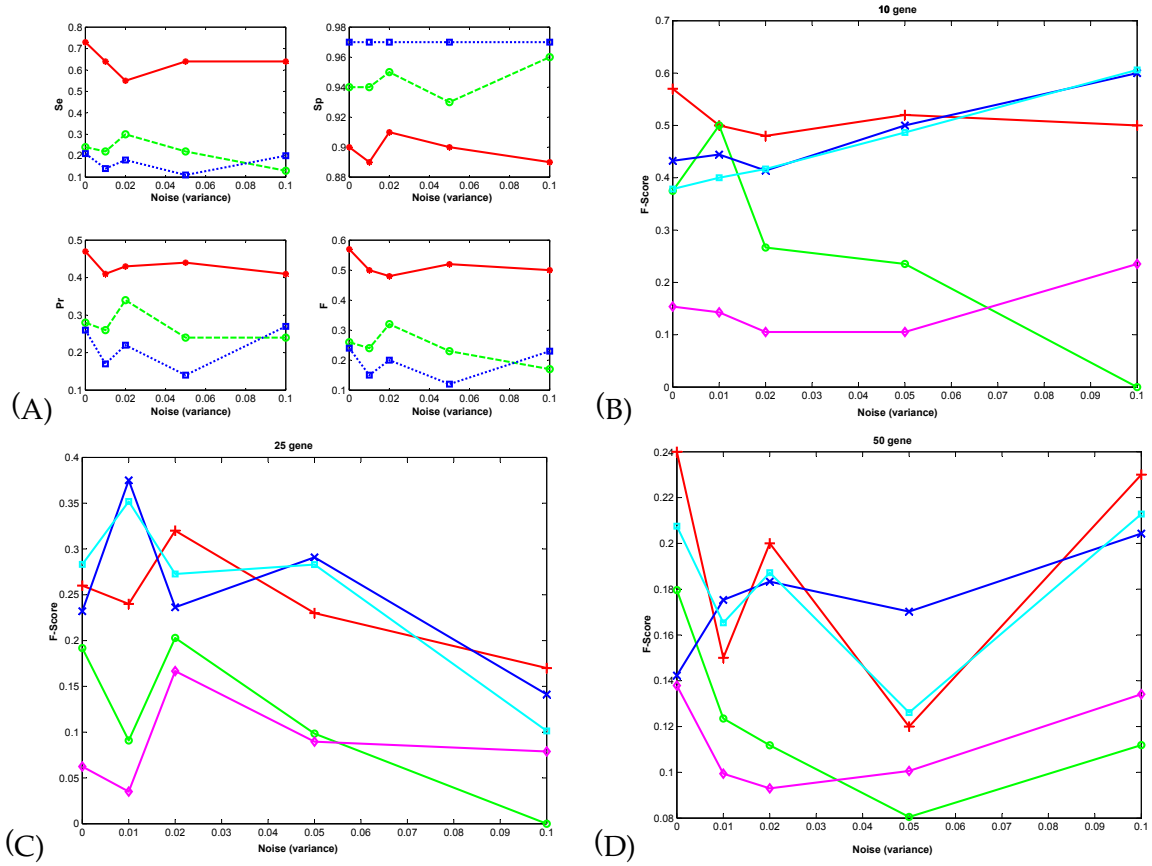

Figure 5.5: Reconstruction of synthetic networks generated using the GeneNetWeaver tool [196]. (A) How performance of our method varies with network size and noise (Red(*) - 10 gene; Green(o) - 25 gene; Blue(square) - 50 gene). The X-axes represent the 5 levels of noise used, whereas the Y-axes represent the corresponding performance measures (see text). (B)-(D) Comparison of performance with 3 other methods for the 10, 25 and 50-gene network. Red(+) - CCIT, Green(o) - BANJO, Blue(x) - BNFinder+BDe, Cyan(square) - BNFinder+MDL, Magenta(diamond) - TDARACNE. X-axes - noise levels, Y-axes - F-score. See text for details.

Probabilistic Network of Yeast

We use a sub-network from the yeast cell cycle, shown in Figure 5.6, taken from Husmeier *et al.* [98]. The network consists of 12 genes and 11 interactions. For each interaction, we randomly assigned a regulation order of 0, 1, 2 or 3. We used two different conditional probabilities for the interactions between the genes, namely, the noisy regulation according to a binomial distribution and the noisy XOR-style co-regulation. For the binomial distribution dependent noisy regulation, the parameters were set as follows: excitation: P(on|on) = 0.9, P(on|off) = 0.1; inhibition: P(on|on) = 0.1, P(on|off) = 0.9. For the noisy XOR-style co-regulation the parameters were set as: P(on|on, on) = P(on|off, off) = 0.1, P(on|on, off) = P(on|off, on) = 0.9 [98]. Eight confounder nodes were also added, resulting in the total number of nodes to be 20.

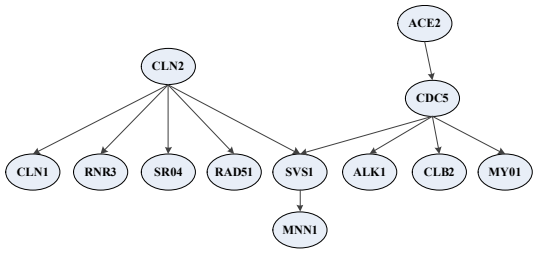

Figure 5.6: Yeast cell cycle sub-network [98]

We used 30, 50 and 100 samples, generated 5 datasets in each case and compared our approach with two other DBN based methods, namely BANJO [255] and BNFinder [246]. Since these methods detect only regulations of order 1, while calculating performance measures for these methods, we ignored the exact orders for the time-delayed interactions in the target network. We could not apply TDARACNE [261] to this network since the generated data has two levels of discrete values and TDARACNE returns error when applied to such datasets. We show the results for this network in Table 5.1, where we observe that our method, coupled with a high precision, outperforms the other two in terms of both sensitivity and specificity. The F-score is also the best in all the cases. This points to the strength of our method in discovering complex interaction scenarios where multiple regulators may jointly regulate target genes with varying time delays.

	N=30				N=50				N=100			
	Se	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*
Proposed Method	**0.62**± **0.12**	0.992± 0.0045	0.57± 0.11	0.59± 0.11	**0.80**± **0.04**	**1.0**± **0.0**	**0.79**± **0.07**	**0.79**± **0.05**	**0.82**± **0.06**	**1.0**± **0.0**	**0.76**± **0.03**	**0.79**± **0.04**
BNFinder +BDe	0.53± 0.04	**0.996**± **0.0006**	**0.68**± **0.02**	**0.59**± **0.02**	0.62± 0.04	0.997± 0.0019	0.74± 0.13	0.67± 0.06	0.69± 0.08	0.997± 0.0007	0.74± 0.06	0.72± 0.07
BNFinder +MDL	0.51± 0.08	**0.996**± **0.0006**	0.63± 0.07	0.56± 0.08	0.60± 0.05	0.996± 0.0022	0.68± 0.15	0.63± 0.09	0.65± 0.0	0.996± 0.0	0.69± 0.04	0.67± 0.02
BANJO	0.51± 0.08	0.987± 0.01	0.49± 0.2	0.46± 0.15	0.55± 0.09	0.993± 0.0049	0.57± 0.23	0.55± 0.16	0.60± 0.08	0.995± 0.0014	0.61± 0.09	0.61± 0.08

Table 5.1: Comparison of CCIT-based method with BANJO and BNFinder on the yeast sub-network

Synthetic Network of Glucose Homeostasis

In higher eukaryotes, glucose homeostasis is maintained via a complex system involving many organs and signaling mechanisms. The liver plays a crucial role in this system by storing glucose as glycogen when blood glucose levels are high, and releasing glucose into the bloodstream when blood glucose levels are low. To accomplish its task, the liver responds to circulating levels of hormones, mainly insulin, epinephrine, glucagon, and glucocorticoids [128].

Le *et al.* [128] conducted an extensive review of the literature regarding the biological components affecting perinatal glucose metabolism. Based on the study, a Bayesian Network model of glucose homeostasis containing 35 nodes and 52 interactions (shown in Figure 5.7) was constructed. We used the model for generating datasets of varying size (50, 75 and 100 samples), having first and second-order regulations using the Bayes Net Toolbox [159]. The random multinomial CPDs used by this approach of data generation were obtained by sampling from a Dirichlet distribution with hyperparameters chosen by the method[4] described in [47] with a corresponding Equivalent Sample Size

[4]The method works as follows: for a variable X_i with k states, a basis vector is constructed for $P(X_i|Pa(X_i))$ by normalizing the vector $\left(\frac{1}{1}, \frac{1}{2}, \cdots, \frac{1}{k}\right)$. For the j-th instantiation $pa(X_i)$ of $Pa(X_i)$, samples are obtained for the probability corresponding to this instantiation by using $\theta_{ij} \sim Dirichlet(s\alpha_{ij})$ where s is the equivalent sample size and the α_{ij}'s are obtained by shifting the basis vector to the right j places where j modulo k is not one.

	N=50				N=75				N=100			
	Se	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*
Proposed Method	0.50	**0.9812**	**0.54**	**0.52**	0.46	**0.9914**	**0.71**	**0.56**	0.54	**0.9906**	**0.72**	**0.62**
BNFinder +BDe	0.48	0.9488	0.29	0.37	0.52	0.9506	0.32	0.39	**0.56**	0.9557	0.36	0.44
BNFinder +MDL	**0.54**	0.948	0.31	0.40	**0.56**	0.9395	0.29	0.38	0.54	0.9369	0.27	0.37
BANJO	0.52	0.97	0.44	0.47	0.48	0.9838	0.57	0.52	0.54	0.9881	0.67	0.60

Table 5.2: Comparison of CCIT-based method with BANJO and BNFinder on the glucose homeostasis network

(ESS) value of 10. The choice of this prior distribution for the conditional parameters ensures a reasonable level of dependence between d-connected variables in the generative structure [47].

We compare our method with the three other methods that were used previously for comparison, namely BANJO [255] and BNFinder [246] (using BDe and MDL). While calculating performance measures for these methods, we ignored the exact orders for the time-delayed interactions in the target network. Similar to the probabilistic network of yeast, we could not apply TDARACNE for this network due to error occurring because TDARACNE is unable to cope with the discrete data. The results are shown in Table 5.2. We observe that, both in terms of specificity and precision, our method outperforms others. The F-score is the highest in all the cases, indicating a good balance between sensitivity and precision.

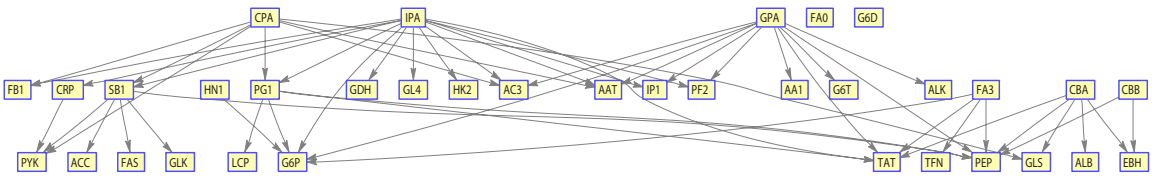

Figure 5.7: Synthetic network of glucose homoeostasis

5.5.2 Real-Life Biological Data of *saccharomyces cerevisiae* (IRMA)

To validate our method with a real-life biological gene regulatory network, we investigate the IRMA network reported in Cantone *et al.* [35]. As stated in previous chapters, there are two sets of gene profiles called Switch ON and Switch OFF for this network, each containing 16 and 21 time series data points, respectively. A 'simplified' network, ignoring some internal protein level interactions, is also reported in [35]. To compare our reconstruction method, we consider 3 other methods, namely, TDARACNE [261], BANJO [255] and BNFinder [246] (both BDe and MDL).

IRMA ON Dataset

The performance comparison amongst various method based on the ON dataset is shown in Table 5.3. We observe that our method clearly outperforms the others. There are no false predictions and precision is the highest. The sensitivity and F-score measures are also very high. One important observation that we make is that the CCIT based approach outperforms the competing algorithms in terms of specificity, precision and F-score, whereas for other algorithms we saw that it lost ground to these methods (e.g., BANJO) for both specificity and precision. This implies that the positive effects of a rigorous statistical significance test is evidenced more clearly when the CCIT-based scoring is used, which were less pronounced when we had been using other algorithms.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
Proposed Method	0.63	**1.0**	**1.0**	**0.77**	0.67	**1.0**	**1.0**	**0.80**
GRNCGA	**0.73±**	0.83±	0.68±	0.70±	**0.80±**	0.90±	0.73±	0.76±
(Average± Std Dev)	**0.054**	0.027	0.044	0.044	**0.072**	0.027	0.06	0.058
TDARACNE	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
BNFinder+BDe	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BNFinder+MDL	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50

Table 5.3: Performance comparison of CCIT-based method using IRMA ON dataset

IRMA OFF dataset

As is usually the case, due to the lack of stimulus, the overall performances of all the algorithms suffer when we use the OFF dataset. The comparison among different methods is shown in Table 5.4. Again, we observe that our method reconstructs the gene network with high precision. Specificity is also high, implying that the inference of false positives is low.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
Proposed Method	0.50	**0.94**	**0.80**	**0.62**	0.50	**0.90**	**0.75**	0.60
GRNCGA	**0.65±**	0.78±	0.60±	**0.62±**	**0.77±**	0.87±	0.66±	**0.71±**
(Average± Std Dev)	**0.054**	0.033	0.051	**0.052**	**0.088**	0.027	0.059	**0.065**
TDARACNE	0.60	-	0.37	0.46	0.75	-	0.50	0.60
BNFinder+BDe	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BNFinder+MDL	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BANJO	0.38	0.88	0.60	0.46	0.33	**0.90**	0.67	0.44

Table 5.4: Performance comparison of CCIT-based method using IRMA OFF dataset

5.5.3 Yeast KEGG Pathway Reconstruction

In order to test the proposed method's performance on yeast *S. cerevisiae* cell cycle, we selected an eleven gene network of the G1-phase: CLN3, CDC28, MBP1, SWI4, CLB6, CDC6, SIC1, SWI6, CLN1, CLN2, CLB5. The data used was obtained from the *cdc28* experiment of Spellman *et al.* [214]. In the later stage of the G1-phase, the CLN3-CDC28 protein kinase complex activates two transcription factors, MBF and SBF, and these promote the transcription of some genes important for budding and DNA synthesis [56,261]. Entry into the S-phase requires the activation of the protein kinase Cdc28p through binding with CLB5 or CLB6, and also the destruction of SIC1 [52]. Also, SWI4 becomes associated with SWI6 to form the SCB complex that activates CLN1 and CLN2 in late G1. MBP1 forms the MCB-binding factor complex with SWI6, which activates DNA synthesis genes and S-phase cyclin genes CLB5 and CLB6 in late G1 [261]. In budding yeast, commitment to DNA replication during the normal cell cycle requires degradation of the

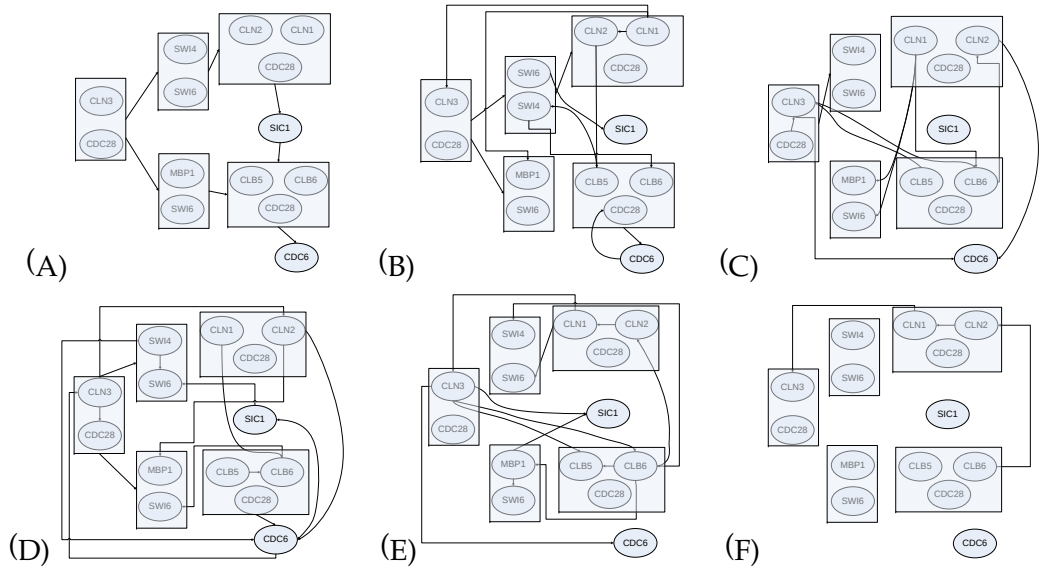

Figure 5.8: Reconstruction of Yeast KEGG Pathway [108]. (A) Target Network. (B) Network Inferred by proposed approach. (C) Network Inferred by TDARACNE. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

cyclin-dependent kinase (CDK) inhibitor SIC1. The G1 cyclin-CDK complexes CLN1-CDK1 and CLN2-CDK1 initiate the process of SIC1 removal by directly catalyzing SIC1 phosphorylation at multiple sites [192, 261].

In Figure 5.8(B)-(F), we report network graphs reconstructed by our proposed approach, TDARACNE, BNFinder(BDe and MDL) and BANJO. We also report the KEGG pathway [108] of the cell cycle in yeast in 5.8(A). Since the ground truth for this network is not known, instead of applying performance measures as a means of determining network accuracy, we refer to the available correct interactions obtained from the KEGG pathway [108] and identify which of the predicted interactions are correct or otherwise. We observe from the results that our approach correctly identifies the regulation of SWI4-SWI6 and MBP1-SWI6 complex by the CLN3-CDC28 complex. Also, the proposed approach infers that the SWI4-SWI6 complex regulates the CLN1-CLN2-CDC28 complex, which is correct. Two more interactions inferred by our approach (CLN1→CLN2 and CLB5-CLB6-CDC28→CDC6) are also correct based on the KEGG pathway. Overall we observe that none of the methods perform particularly well on this network. However, the number of correct predictions by our method (5) is higher than the other three methods.

5.5.4 SOS DNA Repair Network of *E. coli*

We analyze the previously studied SOS DNA repair network in *Escherichia coli* as shown in Figure 5.9(A), which is one of the largest, most complex, and best understood DNA damage-inducible network to be characterized to date.

For our current simulation study, we selected the same 8 genes namely uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB, which were studied in the previous chapters. All four experimental datasets corresponding to various UV light intensities (Exp. 1 and 2:5Jm^{-2}, Exp. 3 and 4:20Jm^{-2}) were considered (separately) for our study.

The results corresponding to Experiment 1 are presented in Figure 5.9(B). Along with our result, we include the results from BANJO, TDARACNE and BNFinder in Figure 5.9(C)-(F) and the target network in 5.9(A). From the results, we observe that our method correctly identifies lexA and recA as the 'hub' genes for this network. Again, the exact ground truth for this network is not precisely known, and hence it is not possible to calculate the four performance measures used for other networks. Instead, using the known interactions obtained from the literature [116, 163], an analysis of correct and incorrect predictions by our method is obtained and shown in Table 5.5. We observe that most of the interactions inferred by our proposed method are correct. It successfully infers lexA as the regulator of uvrD, umuD, uvrA and recA. Also, considering the indirect regulation of RecA through LexA, three more interactions, namely recA$\rightarrow$uvrY, recA$\rightarrow$loxA and recA$\rightarrow$polB can also be considered correct. In contrast, 3 of the 5 identified interactions by TDARACNE [261] are correct. Most of the interactions identified by BANJO and BNFinder+MDL are incorrect. BNFinder+BDe successfully identifies regulation of ruvA, polB and uvrA by lexA. In addition, the regulation of umuD by recA can also be considered correct. However, compared to these methods, our proposed method infers the highest number of correct predictions. The number of incorrect predictions for our method is also very low.

The results for the other three experiments are also satisfactory (see Figure 5.10). We observe that for these experiments, the CCIT-based approach correctly infers 6 (4 direct, 2 indirect), 4 (all direct) and 3 (all direct) interactions from experiments 2, 3 and 4, respectively. The number of incorrect predictions from these datasets are 4, 4 and 5, respectively,

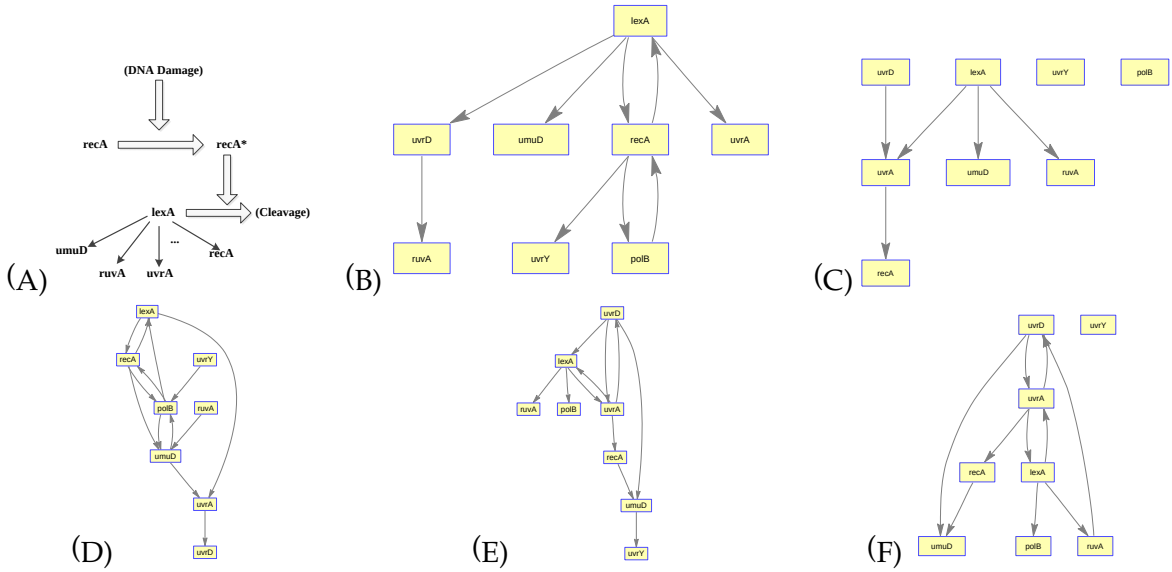

Figure 5.9: Reconstruction of SOS DNA Repair Network (A) Target Network. (B) Network Inferred by proposed approach. (C) Network Inferred by TDARACNE. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

fewer than the incorrect predictions from GRNCGA. Finally, we observe that for all these results, the approach correctly identifies lexA as the 'hub' gene.

5.5.5 Network Analysis of Strongly Cycling Genes in cyanobacteria, *Cyanothece* sp. ATCC 51142

To study our approach on a large scale network, we use a network of a strain of cyanobacteria, namely *Cyanothece* sp. strain ATCC 51142 [217]. Cyanobacteria are oxygen evolving photosynthetic prokaryotes. They play a key role in harvesting solar energy and carbon

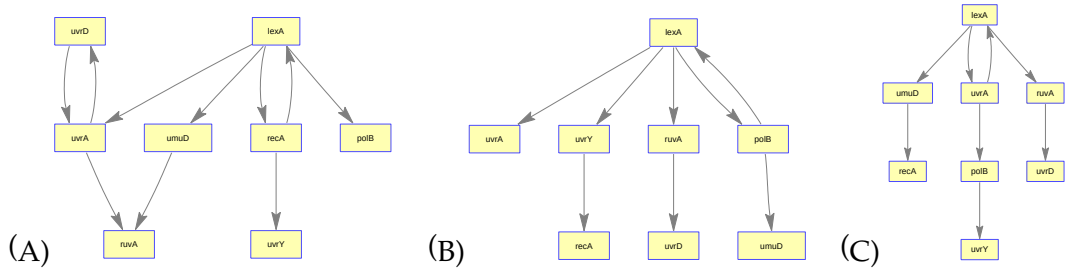

Figure 5.10: Reconstruction of SOS DNA Repair Network by CCIT-based scoring technique (Experiments 2, 3, 4). (A) Experiment 2. (B) Experiment 3. (C) Experiment 4.

Regulator	Target	correct/ incorrect
LexA	uvrD	correct
	umuD	correct
	recA	correct
	uvrA	correct
RecA	uvrY	correct[a]
	polB	correct[a]
	lexA	correct[a]
uvrD	ruvA	incorrect
polB	recA	incorrect

[a] correct considering indirect regulation of RecA through LexA

Table 5.5: Analysis of individual interactions inferred by CCIT-based method - SOS DNA Repair Network

sequestration. They also have the capability of producing biofuel by using the energy from the sun, and combining carbon-dioxide and water. They have become increasingly important recently, due to the possibility of using cyanobacterial pond to naturally sequester carbon-dioxide at source.

The microarray data corresponding to the genes were collected from two publicly available genome-wide microarray datasets of *Cyanothece*, performed in alternating light-dark (LD) cycles with samples collected every 4h over a 48h period: the first one starting with 1h into dark period followed by two DL cycles (DLDL), and the second one starting with two hours into light period, followed by one LD and one continuous LL cycle (LDLL) [237]. In total, there were 24 samples. Using a threshold filter with a 2-fold change cutoff, 730 genes were selected for the analysis. The genes are responsible for performing the major tasks of energy metabolism and respiration, nitrogen fixation, protein translation and folding, and photosynthesis, along with several other tasks. The result obtained using our method is shown in Figure 5.11. The degree distribution is shown in Figure 5.12. To compare our result with the other methods, we applied BANJO, BNFinder(BDe and MDL) and TDARACNE. The results of all the three except BNFinder(BDe) were not satisfactory (for example, the BNFinder+MDL method returns

an empty network when applied; other methods also had similar issues). As a result, we compare our method only with BNFinder+BDe.

Similar to well studied datasets (e.g., yeast [214], *E. coli* [8, 184]), the microarray data set for cyanobacteria also has very few samples. Moreover, not being a well-studied organism, it requires caution in the interpretation of results. We note that GRN reconstruction studies of cyanobacteria reported earlier (e.g., [142, 217, 229]) commonly emphasize an evaluation criteria, namely functional enrichment analysis of sub-networks. Further, another common feature noted for genetic networks [88, 105, 106] is that transcriptional regulatory networks possess the scale free nature of the network topology[5]. Since we have limited samples and also because the ground truth is unknown, we have therefore carried out the evaluation of the inferred network using both: (i) statistical means, i.e., GO functional enrichment analysis (using both $p = 0.05$ and $p = 0.10$), and (ii) R^2 measure of the power-law fit of the network to establish its scale free nature.

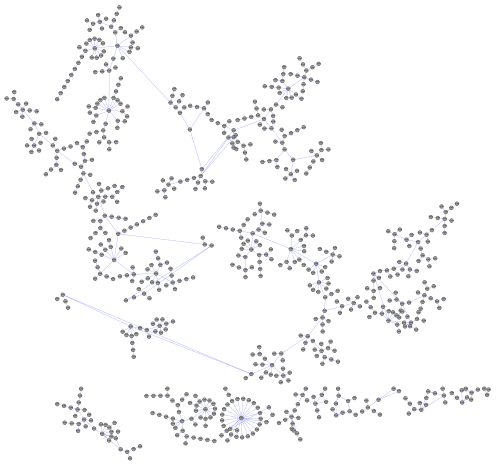

Figure 5.11: Network inferred by CCIT-based approach

The enrichment analysis was done by using gene ontology (GO) database (compiled using two sources: one from the Cyanobase database [111], and another from genome-wide amino sequence matching using the Blast2GO software suite [86]; the compiled database is available as supplementary information of our journal article [154], and freely available online), where every GO term appearing in each sub-network is assessed to find out whether a certain functional category is significantly over-represented in a certain sub-network/cluster, more than what would be expected by chance. The Cytoscape [204]

[5]We clarify that different processes, including genetic networks, will generate scale free networks. However, if a network obtained using microarray data is scale free, it indicates that it is modeling the underlying biological process more accurately.

plugin BiNGO [135] was used for GO functional category enrichment analysis. For BiNGO, we use the combined and filtered gene set as the reference set, the hypergeometric test as the test for functional over-representation, and False Discovery Rate (FDR) as the multiple hypothesis testing correction scheme. A step by step tutorial of how this analysis can be done is shown in Appendix A.

First, we present the results corresponding to $p = 0.05$. The network obtained by BNFinder+BDe has 16 sub-networks each containing at least 3 genes. Of these, 6 sub-networks have significantly enriched functionalities (as determined by the GO functional enrichment test). Of the other 10, we compute the 3 most densely connected hubs for each sub-network, and in 2 of 10 such sub-networks, the hubs have defined significantly enriched functionalities. On the other hand, in our result, there are 14 sub-networks in total having at least 3 genes. Of these, 3 sub-networks have defined enriched functions (the largest sub-network has the role of nitrogen fixation according to the enrichment test). Of the other 11, we compute the 3 most densely connected hubs for each sub-network, and in 5 of the 11 such sub-networks, the hubs have defined significantly enriched functionalities.

The results corresponding to $p = 0.10$ show that for BNFinder+BDe, 7 sub-networks have enriched functionalities (as determined by the test). Of the other 9, we compute the 3 most densely connected hubs for each sub-network, and in 2 of the 9 such sub-networks, the hubs have defined enriched functionalities. In contrast, the result using our approach has 5 sub-networks with defined significantly enriched functions (the largest sub-network has the role of nitrogen fixation, similar to the $p = 0.05$ case). Of the other 9, we compute the 3 most densely connected hubs for each sub-network, and in 6 of the 9 such sub-networks, the hubs have defined significantly enriched functionalities.

We also test the networks to assess whether they are scale free, using a power-law fit. The R^2 value of the fit corresponding to our network is 0.93, which is a better fit compared to BNFinder+BDe (0.62).

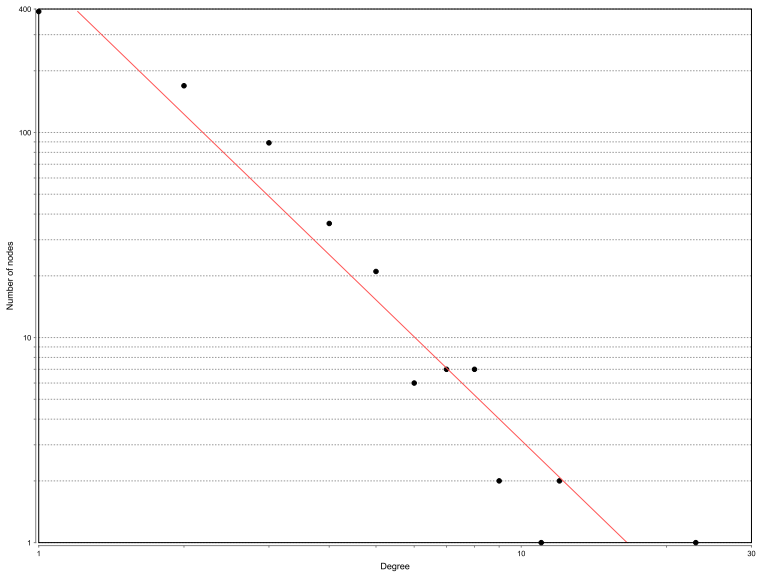

Figure 5.12: Degree distribution analysis of the resultant network of *Cyanothece*. We used a power-law fit which yields $R^2 = 0.93$. The result confirms that the inferred network is scale free.

So far in this chapter, we have introduced an extended modeling framework that can simultaneously represent instantaneous and multi-step time-delayed genetic interactions. Incorporating this framework, we implemented a local search based GRN reconstruction algorithm using a novel scoring metric called CCIT that supports the biological truth that some genes may co-regulate other genes with different orders of regulation. However, the improved framework takes multiple step time-delayed interactions, and that means the search space becomes even larger compared to just single-step time-delayed regulations (note that instantaneous interactions are present throughout alongside). This huge search space will undoubtedly have multiple local optimal solutions. When explored using evolutionary strategies, due to the phenomena of genetic drift, the stochastic variations caused by the genetic operators can result in a population drift to any of these multi-modal peaks [95, 99]. The net effect is that convergence may slow down, and in the worst case, it may get stuck in a local optima. Using a simple local search strategy is not ideal in these scenarios, and thus in the next few sections of this chapter, we will build a novel genetic algorithm based evolutionary two-stage multi-threaded search strategy which explores a significantly larger search space. In the next few sections, we will first elaborate why a multi-threaded strategy is beneficial, and afterwards we will present our design and assess its performance.

5.6 Improving the Search Strategy

As has already been stated, BNs are very effective in dealing with noise, incompleteness and stochastic aspects of gene regulation; however, due to several complexities in learning static BNs (e.g., Chickering [45] showed that learning BNs using BDe scoring metric is NP-complete), when it comes to learning BNs and DBNs, most authors have resorted to greedy hill-climbing techniques [152], evolutionary frameworks (such as genetic algorithms [84,153,176]), metaheuristic methods (simulated annealing [236]), and local search methods (e.g. Tabu search [232]). One major problem with learning BNs and DBNs using the meta optimization frameworks is the multi-modal nature of the search space due to which basic evolutionary strategies often lead to local optima. Although increasing the population size might appear to be an easy option, in practice, increasing the population size does not help because few individuals can continue to dominate the search procedure. This problem is compounded by the computational overhead arising due to increasing the population size [140]. To explore the whole search space so as to improve the chance of converging to the global optima, and at the same time keeping computational resources in check, we propose executing parallel threads of Genetic Algorithms (GA), and then combining relevant patterns from these solutions, which is more likely to obtain the globally optimal/near-optimal GRN. To elaborate the effect of parallel execution, we follow Goldberg *et al.* [85], and propose that the search space for the objective function $f(\cdot)$ can be denoted by ω. Now, assuming,

- $f(\cdot)$ has m optima in ω (i.e., the search space is divided into m regions termed ω_i's, such that $\omega = \omega_1 \cup \omega_2 \ldots \cup \omega_m$), and one of them is the global optima; and
- each run of GA converges to one optima, which is either local or global,

the probability P that at least 1 of k GA runs converges to the global optima is:

$$P = \begin{cases} 1 - (\frac{m-1}{m})^k & if \quad k > 1 \\ \frac{1}{m} & if \quad k = 1 \end{cases} \tag{5.13}$$

assuming independence and uniform distribution. For example, with $m = 6$ and $k = 3$, the parallel search has a higher probability (0.42) of convergence to the global optima,

compared to 0.17 from a single-thread approach. Also, in terms of diversity, the probability that all the k threads of GAs will converge to the same solution is significantly less ($\frac{1}{m^k}$) than a run without parallel execution ($\frac{1}{m}$, $k > 1$). These observations are valid even if the search space is not uniform.

However, the benefits of running multiple threads of GA in parallel is pretty intuitive, and similar effects can be achieved even if we run a non-parallel GA multiple times (or even using multiple threads of GA simultaneously). Unlike these naive approaches, our proposed method makes novel use of the multiple solutions resulting from these parallel threads and extracts the structural similarity patterns so that this information can be used subsequently, thereby fully exploiting the benefits of running in parallel threads. This technique of getting representatives from the entire search space, and using the information subsequently, increases the ability of our method to converge to the global optima compared to these naive approaches. Similar approaches have been used in other research areas, for example, for protein structure prediction problems [99], where parallelly running multiple threads of memetic algorithms and finding 'memes' from these threads were shown to produce significantly better solutions.

One important local property of networks is network motifs, which are recurrent and statistically significant subgraphs that repeat themselves among various networks. Each of these subgraphs, defined by a particular pattern of interactions between vertices, may reflect a framework in which particular functions are achieved efficiently, and thus present interesting functional properties. Network motifs have recently acquired attention as an important means to uncover structural aspects of complex networks [96, 99]. In the remainder of this chapter, we propose a novel two-stage genetic algorithm (GA) based framework using the concept of network motifs. Briefly, the motifs, in the form of frequent subgraphs occurring across a pool of 'local' optimal solutions will capture the characteristics of these solutions in the first stage. In the second stage that follows, these local characteristics are combined to obtain the globally optimal/near-optimal solution. Some of the benefits of the proposed approach include:

1. A higher probability of covering the entire search space.

2. Avoiding going deeper inside a local optima, thereby reducing computational effort. To see how we gain reduction in our approach, consider Figure 5.13. In the

figure, unlike solution (individual) A, solutions B, C and O (local optima) will have a similar score since they are quite close together (similar for O, F, E and D). In our approach, during the first stage of execution, rather than continuing execution until we reach solution O, we detect when execution reaches solutions B/C (or D/F/E), i.e., *near* an optimal solution. We find the common trends present in these solutions and use this information in subsequent calculations.

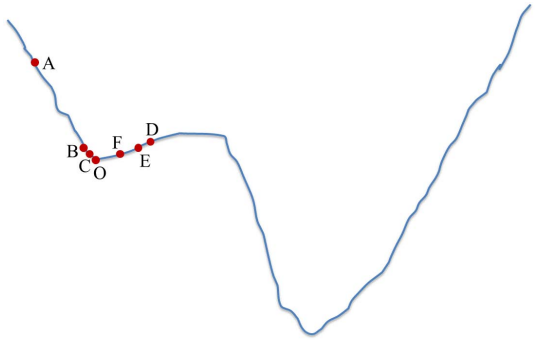

Figure 5.13: One of the benefits of our proposed motif based approach. We do not need to continue execution until we reach an optima. This example considers a minimization problem.

3. The approach builds a knowledge base *ab initio*, i.e., without any prior information. Although no prior external knowledge other than the micorarray data is used, the second stage combines the implicit knowledge acquired from different search regions to increase the probability of convergence to the global optima.

Finally, the whole algorithm runs in parallel threads so there is no additional computational cost incurred in using the proposed framework.

5.7 Techniques for Frequent Subgraph Mining

In this section we discuss the different frequent subgraph mining techniques that can be used for identifying motifs. Informally, these motifs are the frequent subgraphs occurring across a pool of networks. For a frequent subgraph mining technique to be appropriate for our purpose, it should have some desirable properties such as: (*i*) the ability of detecting overlapping subgraphs, (*ii*) the ability of detecting approximate subgraphs (i.e., the connectivity within each subset of nodes is not exactly consistent between graphs), and (*iii*) the technique should be scalable to large scale gene regulatory networks.

Tian *et al.* developed a query based subgraph matching technique called Substructure Index-based Approximate Graph Alignment (SAGA) [228], where an index on small substructures of the graphs are stored in a database. The query graph is broken up into small fragments and then the database is probed using a matching algorithm to produce hits for substructures in the query. This technique allows for node gaps, node mismatches and graph structural differences and does not require any constraints to be designed in advance. However, the disadvantages are that one has to maintain a database of small structures and that it is query based. In applications such as graph mining in biological networks, its possible that we want to extract subgraphs without having identified queries.

GraphScope [223] is another method for finding coherent clusters in graphs over time. It assumes the sequence of graphs are bipartite. It then partitions this sequence into segments using an information theoretic criterion and then finds clusters within each segment. This is an interesting approach but it is limited by the fact that since it partitions the sequence of graphs into segments, it can only find clusters in neighboring time points. However, we seek to find recurring subgraphs that may not occur in adjacent or nearby time points.

Frequent subgraph mining techniques which decompose the networks into smaller pieces and apply pattern expansion techniques have been proposed in Kuramochi *et al.* and Yan *et al.* [125, 251]. Techniques which perform frequent set mining and subsequently check for connectivity have also been proposed [120]. However, these approaches encounter scalability issues when applied to massive biological networks [96]. Interpretability issues also arise because in many cases a discovered frequent dense subgraph may not represent a tight association among its nodes. These two issues can be solved by the CODENSE algorithm, proposed by Hu *et al.* [96]. Using the concept of coherent dense subgraphs, this algorithm produces frequent subgraphs having much better interpretability compared to other algorithms. Also, the design of CODENSE can solve the scalability issue. Instead of mining each biological network individually, CODENSE compresses the networks into two metagraphs and performs clustering in these two graphs only. Thus, CODENSE can handle any large number of networks. Moreover, unlike most other frequent subgraph mining algorithms, CODENSE can discover overlapping subgraphs. One limitation of this algorithm is that it finds subgraphs from a

global static graph constructed from all the original graphs. Thus, it is unable to capture interactions that occur locally to a few neighboring graphs.

5.8 Identifying the Motifs

To find the characteristics or the commonly occurring patterns that exist in various local solutions, we use frequent subgraph mining, available from graph theory. We note that when a genetic algorithm nears an optimal solution, the speed of convergence slows down considerably. Hence, when we detect that for a thread of GA the best solutions are almost the same for a number of generations, or a majority of the individuals in the current population have almost a similar structure (and thus a similar score), we consider that the search is nearing an optimal solution and use these solutions to find common 'patterns' (we call these common patterns motifs) present across them. Formally, the local (or global) optima detection (i.e., whether the search has reached *near* a local/global optima) can be defined by Equation 5.14.

$$C_{opt} = \begin{cases} 1 \quad if \begin{cases} F^*(P(t)) - F^*(P(t+\tau)) = \Delta_1 \\ F^*(P(t+\tau)) - F^{4/5}(P(t+\tau)) = \Delta_2 \end{cases} \\ 0 \qquad\qquad otherwise \end{cases} \tag{5.14}$$

where $F^*(\cdot)$ returns the best fitness of the population ($P(\cdot)$) and $F^{4/5}(\cdot)$ returns fitness of $4/5$th individual.

To detect the 'common patterns' or motifs present in the converging individuals, we use the well known frequent subgraph mining algorithm called CODENSE [96]. As mentioned previously, some unique features of CODENSE that make this algorithm particularly suited to our approach is that it can detect overlapping subgraphs, and it also supports approximate subgraph matching. Moreover, it is scalable to large networks which is a common feature of biological networks. The CODENSE algorithm is based on two observations: (*i*) if a frequent subgraph is dense, then it must be a dense subgraph in the summary graph (a graph where only frequent edges from the original graphs exist), and (*ii*) if the edges in a subgraph show high correlation in their occurrences across a graph set (called coherent graph), then its second-order graph must be dense. The second-order

graph is a special type of graph where vertex set of the second-order graph is the edge set of the original graphs and and an edge connects vertices u and v if the occurrence of the corresponding edges in the original graphs have a similar pattern.

CODENSE is a five-step algorithm for mining coherent (meaning its edges show high correlation in their occurrences across a graph set), dense subgraphs. In the first step, it builds a summary graph which prunes out infrequently occurring edges. The summary graph contains only those edges which occur in more than l graphs in the graph dataset, where l is a user-defined support threshold. Then, it identifies dense subgraphs in the summary graph using a subroutine called MODES which is based on the HCS algorithm by Hartuv *et al.* [90].

The original HCS algorithm uses a minimum-cut criteria for finding dense subgraphs. Instead of using only the minimum-cut criteria of the HCS algorithm, MODES adaptively uses normalized-cut and minimum-cut criteria for finding dense subgraph. This dense subgraph is then condensed to a single vertex, and a condensed graph is built connecting this condensed vertex to all the other vertices of the original graph. In the next step, this condensed graph is re-clustered using the modified HCS algorithm (i.e., two different cuts). Once this clustering is done, if any newly discovered dense subgraph contains condensed vertices, MODES restores the condensed vertices back into subgraphs. Finally, MODES conducts a few tests on the un-condensed vertices to avoid the repetitive discovery of already discovered dense subgraphs.

In the third step, CODENSE builds a second-order graph for each dense summary subgraph obtained from the previous step. Next, CODENSE identifies dense subgraphs in the second-order graph, which is based on the principle that the high connectivity among vertices in the second-order graph indicates that the corresponding edges show high similarity in their occurrences across the original graphs. In the final step, it converts the second-order subgraphs identified previously to first-order graphs and applies the MODES subroutine on each of them to get the final motifs.

One issue with the CODENSE based approach is that the output of CODENSE is an undirected graph. In our case, the transformation of undirected to directed graph is quite straightforward, since we already know the original graphs in which they are occurring. As a terminal case, if corresponding to an undirected edge we find evidence

of both directions in different graphs (e.g., one direction in some graphs and the opposite direction in others), we take a majority voting scheme to get the direction.

In the next section, we describe the *mDBN* algorithm, which makes use of the concepts discussed above.

5.9 mDBN: Motif Based Learning of Gene Regulatory Networks

mDBN (Motif Based Learning of Gene Regulatory Networks using Dynamic Bayesian Network) is an evolutionary approach that primarily operates in two stages (see Figure 5.14). These two stages constitute what we call a *Master Iteration*. In the first stage of each master iteration, it runs k parallel threads of GA. The GA begins with a sample population of randomly selected network structures and their fitness calculated. Iteratively, crossover and mutation of networks within a population are performed and the best fitting individuals of the population are kept for future generations. During crossover, two random edges are chosen and swapped. Mutation is applied on an individual edge of a network. For our study, we incorporate the following three simple operations:

- Deleting a random edge
- Creating a random edge
- Changing direction of a randomly selected edge

We keep a running track of the best individuals from each generation. If the search has reached *near* a local optima (identified using equation 5.14, with $\Delta_1 > 0$), then we apply the CODENSE algorithm to find motifs from the individuals whose fitness scores are within a close range (i.e., within a Δ_1 difference). The undirected substructures obtained from the CODENSE algorithm are then assigned directions to obtain the final motifs, which are then used in the second stage of execution for this master iteration.

In the second stage of each master iteration, mDBN takes the motifs obtained from different threads of execution. The motifs can come from two sources: from GA threads in the same master iteration, or from the stage-2 GA of the previous master iteration. The motifs are then fixed into the corresponding positions within the network, and these are then used as templates for generating the individuals. In order to ensure that the motif

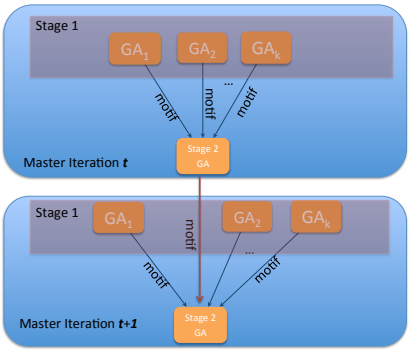

Figure 5.14: A schematic view of mDBN

based individuals do not dominate the search process, at most 80 percent of the overall population gets seeded by these template based individuals. Put more clearly, since the number of motifs generated depends on CODENSE and are not fixed a-priori, for each motif we generate one individual (the fixed motif portion + random arcs). If the number of motifs is higher than $0.8 \times population\_size$, the best scoring $0.8 \times population\_size$ number of individuals are retained, and the rest are discarded. The rest of the population ($0.2 \times population\_size$ individuals) is then seeded randomly. Also, during the execution of the GA, if we see that for consecutive 5 generations the scores are not increasing, we aggregate the five networks using a majority voting scheme with a view to improving the score. Finally, when we detect that the stage-2 GA has reached an optima (Equation 5.14, with $\Delta_1 = 0$), we take the best solution and the motifs from this GA. Since in this case Δ_1 will be zero, we take the individuals who have scores within the range $[0, \Delta_2]$ of the best score, for the motif calculation. We then test for convergence of the master iterations (the best fitness values from different master iterations are almost the same), and when it converges, the best scoring individual is taken as the final solution. The reason behind this iterative execution of the master iterations is that there may be cases where even with the second stage of computation, the two stage formulation may not converge to the global optima. To address such cases, we run the master iterations repeatedly until we see that the results from different master iterations are giving almost similar results (within a small deviation, Δ). This would indicate a high probability that the algorithm has reached the global optima.

The overall execution of the approach is summarized as an algorithm in Tables 5.6 and 5.7.

1: Procedure mDBN

2: $prev\_mi\_motifs \leftarrow \varnothing$
3: repeat
4: $stage1motifs \leftarrow \varnothing$
5: **for** $i \leftarrow 1 \dots k$
6: $stage1motifs(i) \leftarrow thread_i$.GA$(\varnothing, \varnothing, \Delta_1, \Delta_2)$
7: **end for**
8: $[stage2motifs, stage2scores(t), best\_indiv(t)] \leftarrow$
9: GA$(stage1motifs, prev_mi_motifs, 0, \Delta_2)$
10: $prev\_mi\_motifs \leftarrow stage2motifs$
11: $c_{opt} \leftarrow$check_optima$(stage2scores, \Delta_1, 0)$
12: until $(C_{opt} = 1)$

13: end Procedure mDBN

Table 5.6: Algorithm mDBN

5.10 Experimental Results for mDBN

We evaluate the overall accuracy of the proposed method by the well known performance measures of sensitivity, specificity, precision and F-score. For all the experiments, we used an initial population size of 100, and the τ value was set to 3. The number of threads (k) was set to 5, and Δ, Δ_1 and Δ_2 values were set to 0.5. Finally, the crossover and mutation probabilities were set to 0.5 and 0.1, respectively.

5.10.1 Synthetic Networks

Glucose Homeostasis Network

As already stated in Section 5.5.1, glucose homeostasis is maintained via a complex system involving many organs and signaling mechanisms in higher eukaryotes. The liver plays a crucial role in this system by responding to circulating levels of hormones, mainly insulin, epinephrine, glucagon, and glucocorticoids [128].

Based on the study of Le *et al.* [128], a Bayesian Network model of glucose homeostasis containing 35 nodes and 52 interactions was constructed, which has been shown in Figure 5.7. We used the same model for generating datasets (50 samples) having first-order and second-order regulations using the Bayes Net Toolbox [159]. The random multinomial CPDs used by this approach of data generation were obtained by sampling

1. Create initial population of network structures using parameters $stage1motif$ and $stage2motif$. For stage-1 GA, these parameters will be empty. Otherwise, use the motif templates to produce initial individuals. Ensure that all assumptions listed in Section 5.3 are satisfied.
2. Evaluate each network and sort the chromosomes based on equation 5.4.
   (a) Generate new population by applying crossover and mutation on the previous population. Ensure that all assumptions listed in Section 5.3 are satisfied.
   (b) Evaluate each individual using the fitness function and use it to sort the individual networks.
   (c) If the best individual score has not increased for consecutive 5 times, aggregate the 5 best individuals using a majority voting scheme. Ensure that all assumptions listed in Section 5.3 are satisfied.
   (d) Take the best individuals from the two populations based on fitness score and create the population of elite individuals for the next generation.
3. Repeat steps a) - d) until the stopping criteria (using parameters Δ_1 and Δ_2) are satisfied.

   When the GA stops:
   (a) Take the best chromosome and reconstruct the final genetic network.
   (b) Find the motifs for which the local optima test is satisfied. If $\Delta_1 = 0$, use individuals which are within the range $[0, \Delta_2]$ of the best score.

Return *motifs, best score* and *best network*. ◁

Table 5.7: Genetic algorithm with motif based search

from a Dirichlet distribution with hyperparameters chosen by the method described in [47] with a corresponding Equivalent Sample Size (ESS) value of 10, which ensures a reasonable level of dependence between d-connected variables in the generative structure [47, 154, 250].

First, to assess whether the motif based discovery of local patterns indeed provides accurate estimates of the local structures, we compare a few of the discovered motifs with the corresponding structure in the target graph. The comparisons are shown in Figures 5.15, 5.16, 5.17 and 5.18. In these figures, figures (A) represent the relevant portions of the target network (corresponding nodes have been labeled in red), and figures (B)

show the discovered motifs. We observe that the motifs are highly accurate. It may be noted that most of the discovered motifs show similar accuracy.

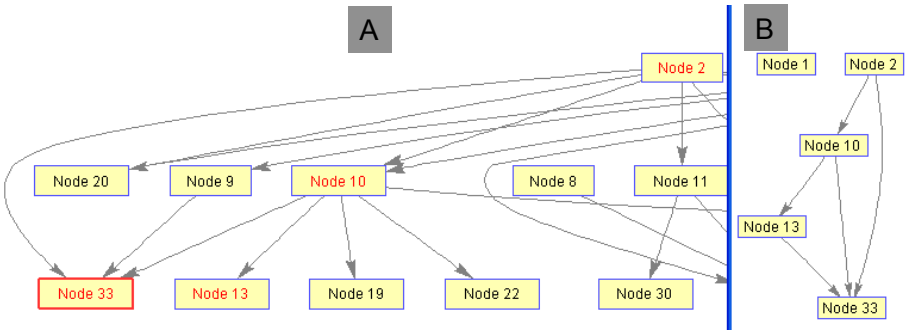

Figure 5.15: A motif discovered by mDBN. (A) Target graph (relevant portion). Corresponding nodes are labeled red. (B) Discovered motif. 4 out of 5 arcs in the motif are correct.

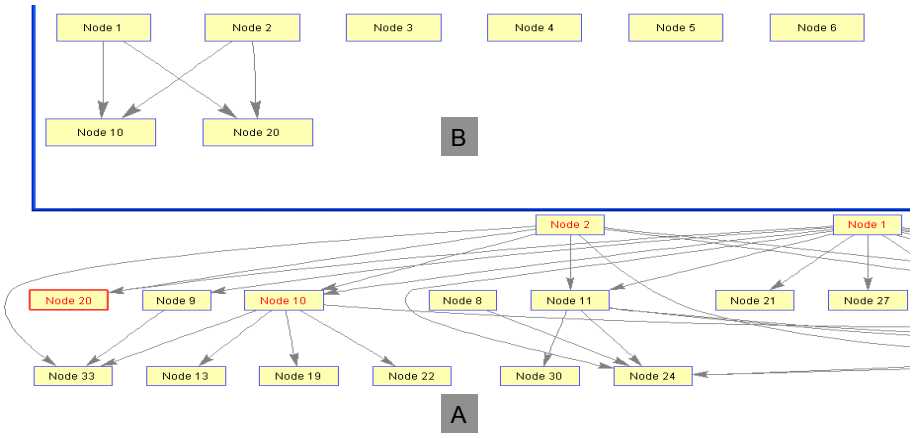

Figure 5.16: Another motif discovered by mDBN. (A) Target graph (relevant portion). Corresponding nodes are labeled red. (B) Discovered motif. All the arcs in the motif are correct.

Next, we compare our approach with three other methods, namely regular GA based on the CCIT score (to find out how much extra gain we achieve in terms of performance due to the motif based scheme), and two other BN based methods, BANJO [255] and BN-Finder [246] (using BDe and MDL). While calculating performance measures for these methods, we ignored the exact orders for the time-delayed interactions in the target network. The results are shown in Table 5.8. We observe that in terms of all the four performance measures, our method outperforms others. We observe that the motif based strategy increases the precision greatly, compared to the non-motif based approach. Also, the F-score with mDBN is the highest in all the cases, indicating a good balance between sensitivity and precision.

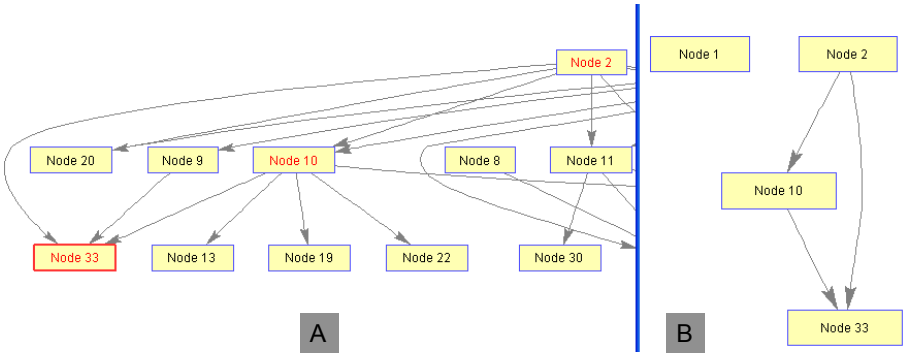

Figure 5.17: Another motif discovered by mDBN. (A) Target graph (relevant portion). Corresponding nodes are labeled red. (B) Discovered motif. All the arcs in the motif are correct.

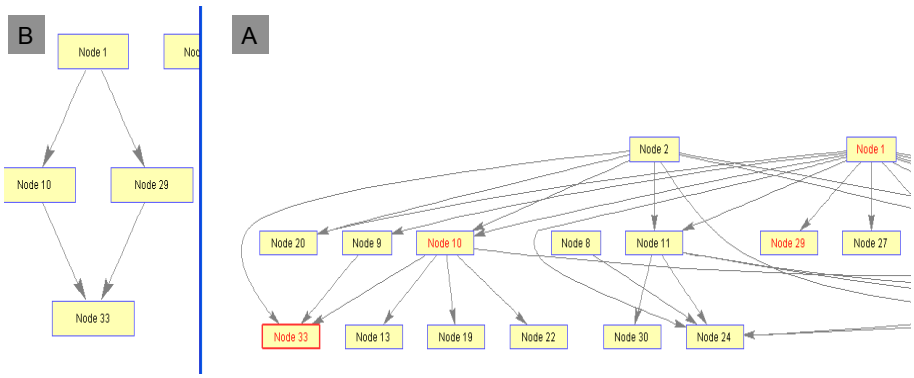

Figure 5.18: Another motif discovered by mDBN. (A) Target graph (relevant portion). Corresponding nodes are labeled red. (B) Discovered motif. All the arcs in the motif are correct.

Yeast Cell Cycle Sub-network

We use a sub-network from the yeast cell cycle (shown in Figure 5.6), taken from Husmeier *et al.* [98]. The network consists of 12 genes (and eight confounder nodes, resulting in 20 nodes in total for the analysis) and 11 interactions. For each interaction, we randomly assigned a regulation order of 0, 1, 2 or 3. We used two different conditional probabilities for the interactions between the genes, namely, the noisy regulation according to a binomial distribution and the noisy XOR-style co-regulation, and the parameter values were the same as those considered in Section 5.5.1 [98, 154].

We used 30 and 50 samples, generated 5 datasets in each case and compared our approach with the same algorithms used in the previous network study: "GA based CCIT score"[6], BANJO [255] and BNFinder [246]. Since the last two methods detect only regulations of order 1, while calculating performance measures for these methods, we

[6]The CCIT score used with basic GA.

	Se	*Sp*	*Pr*	*F*
mDBN	$\mathbf{0.54 \pm 0.009}$	$\mathbf{0.99 \pm 0.004}$	$\mathbf{0.70 \pm 0.11}$	$\mathbf{0.61 \pm 0.04}$
CCIT+GA	0.51 ± 0.01	0.9812 ± 0.004	0.54 ± 0.03	0.52 ± 0.02
BNFinder+BDe	0.48	0.9488	0.29	0.37
BNFinder+MDL	**0.54**	0.948	0.31	0.40
BANJO	0.52	0.97	0.44	0.47

Table 5.8: Comparison based on the 35-gene glucose homeostasis network

ignored the exact orders for the time-delayed interactions in the target network. We show the results for this network in Table 5.9, where we observe that compared to the non-motif based GA using CCIT (CCIT+GA), there is a steady performance improvement in terms of all the measures. The sensitivity and the F-score of mDBN are the highest for both datasets. Also, precision and specificity are quite high, with specificity nearing the best possible value (1), which indicates that inference of false positives is very low.

	N=30				**N=50**			
	Se	*Sp*	*Pr*	*F*	*Se*	*Sp*	*Pr*	*F*
mDBN	$\mathbf{0.66}\pm$ **0.04**	$0.992\pm$ 0.004	$0.61\pm$ 0.07	$\mathbf{0.63}\pm$ 0.03	$\mathbf{0.84}\pm$ **0.04**	$0.998\pm$ 0.004	$\mathbf{0.83}\pm$ **0.08**	$\mathbf{0.83}\pm$ **0.03**
CCIT+ GA	$0.62\pm$ 0.12	$0.992\pm$ 0.0045	$0.57\pm$ 0.11	$0.59\pm$ 0.11	$0.80\pm$ 0.04	$\mathbf{1.0}\pm$ **0.0**	$0.79\pm$ 0.07	$0.79\pm$ 0.05
BNFinder +BDe	$0.53\pm$ 0.04	$\mathbf{0.996}\pm$ **0.0006**	$\mathbf{0.68}\pm$ **0.02**	$0.59\pm$ 0.02	$0.62\pm$ 0.04	$0.997\pm$ 0.0019	$0.74\pm$ 0.13	$0.67\pm$ 0.06
BNFinder +MDL	$0.51\pm$ 0.08	$\mathbf{0.996}\pm$ **0.0006**	$0.63\pm$ 0.07	$0.56\pm$ 0.08	$0.60\pm$ 0.05	$0.996\pm$ 0.0022	$0.68\pm$ 0.15	$0.63\pm$ 0.09
BANJO	$0.51\pm$ 0.08	$0.987\pm$ 0.01	$0.49\pm$ 0.2	$0.46\pm$ 0.15	$0.55\pm$ 0.09	$0.993\pm$ 0.0049	$0.57\pm$ 0.23	$0.55\pm$ 0.16

Table 5.9: Comparison of mDBN on the yeast sub-network

5.10.2 SOS DNA Repair Network of *E. coli*

Next, we analyze the SOS DNA repair network in *E. coli* as shown in Figure 5.19(A), which is well known for its responsibility in repairing the DNA if it gets damaged.

The expression kinetics of the same 8 genes previously studied, namely uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA and polB, was obtained from Uri Alon Lab [8] for this study, which contains 50 samples evenly spaced by 6 minute intervals. As already stated, four datasets are available, and the result corresponding to Experiment 1 is presented in Figure 5.19(B). Along with our result, we include the results from BANJO, TDARACNE and BNFinder in Figure 5.19(C)-(F) and the target network in 5.19(A). From the results, we observe that our method correctly identifies lexA and recA as the 'hub' genes for this network. The exact ground truth for this network is not precisely known, and hence it is not possible to calculate the well known performance measures. Instead, using the known interactions obtained from the literature [116, 163], an analysis of correct and incorrect predictions by our method is obtained and shown in Table 5.10. We observe that most of the interactions inferred by our proposed method are correct. It successfully infers lexA as the regulator of polB, uvrY, uvrD and uvrA. Also, considering the indirect regulation of RecA through LexA, five more interactions, namely recA→polB, recA→lexA, recA→umuD, recA→uvrA and recA→uvrD can also be considered correct. Overall, we observe that compared to other methods, mDBN infers the highest number of correct predictions. The number of incorrect predictions is also low.

The results corresponding to other experiments are shown in Figure 5.20. The results are pretty encouraging, and we observe that for all the experimental datasets the mDBN approach infers 6 correct interactions (4 direct and 2 indirect for Experiment 2, 3 direct and 3 indirect for Experiment 3, and 5 direct and 1 indirect interactions for Experiment 4). When compared to the CCIT-based approach, it inferred 6, 4 and 3 correct interactions, respectively for these three datasets. Also, the number of incorrect inferences is lower when we use mDBN. It infers 4, 3 and 5 incorrect interactions for the Experiments 2, 3 and 4, respectively.

5.11 Summary

In this chapter, we extended the modeling framework proposed in the previous chapter to model both instantaneous and *multi-step* time-delayed interactions. We also proposed a novel scoring metric that scores these two types of interactions simultaneously. Further,

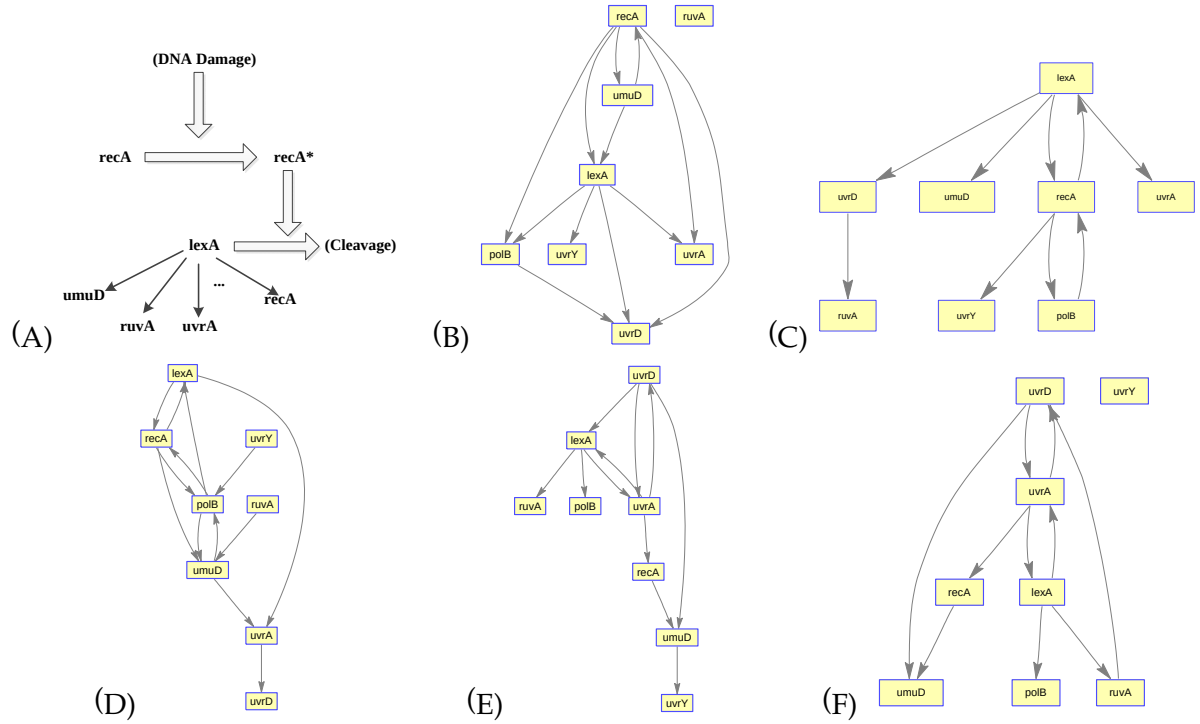

Figure 5.19: Reconstruction of SOS DNA Repair Network (Experiment 1). (A) Target Network. (B) Network Inferred by mDBN. (C) Network Inferred by CCIT. (D) Network Inferred by BANJO. (E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

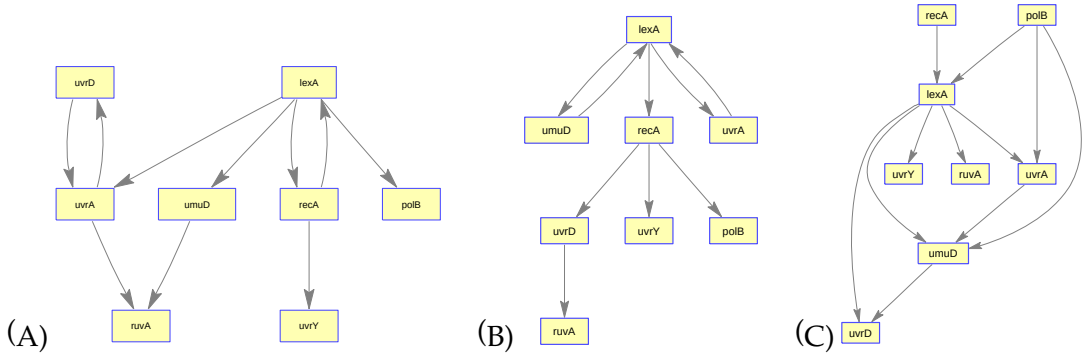

Figure 5.20: Reconstruction of SOS DNA Repair Network by mDBN (Experiments 2, 3, 4). (A) Experiment 2. (B) Experiment 3. (C) Experiment 4.

Regulator	Target	correct/ incorrect
lexA	polB	correct
	uvrY	correct
	uvrD	correct
	uvrA	correct
recA	polB	correct[a]
	lexA	correct[a]
	umuD	correct[a]
	uvrA	correct[a]
	uvrD	correct[a]
umuD	recA	incorrect
polB	uvrD	incorrect
umuD	lexA	incorrect

[a] Correct considering indirect regulation of RecA through LexA

Table 5.10: Analysis of individual interactions inferred by mDBN - SOS DNA Repair Network

we developed a two-stage GA framework that makes use of the above contributions, and identifies network motifs for obtaining near-optimal solutions.

Biologically, the extension of the modeling framework to multi-step time delays implies that different genes can have different time delays with their regulator genes. The proposed scoring metric has the decomposability property, and it also implicitly includes the biological truth that some genes can jointly regulate other genes. Incorporating these novel features of the scoring metric and the extended modeling framework, we performed experiments on different synthetic networks of varying complexities and also on real-life biological networks. Our method showed improved performance compared to other recent methods, both in terms of reconstruction accuracy and number of false predictions, at the same time maintaining comparable or better true predictions. For our previously proposed approaches (e.g., GRNCGA) we noted in a number of cases that it performed satisfactorily in terms of the number of correct predictions (i.e., sensitivity) and also in the overall balance of sensitivity and precision (i.e., F-score); however, it was

not the best performer in terms of specificity and precision. From the results for the real-life experiments, we observe that the performance is greatly enhanced by the new scoring metric proposed in this chapter.

After improving the network learning by an improved scoring metric, a natural enhancement of the overall approach is to improve the search procedure. This becomes increasingly important due to the incorporation of multi-step time-delayed interactions in the framework. To improve the search procedure, we proposed a motif based two-stage genetic algorithm framework where repetitive patterns from multiple local optimal solutions are obtained in the first stage using graph theoretic algorithms, and in the second stage these solutions are combined in a novel way to obtain the near-optimal solution, thereby having a much higher probability of obtaining a better solution compared to basic GA based search approaches. For this part also, experiments have been carried out using both synthetic and real-life gene regulatory networks. The proposed approach shows better performance compared to the two other DBN based algorithms (BANJO and BNFinder), and also with the same algorithm without incorporating the newly proposed concept of motifs.

The reconstruction techniques reported in this and the previous chapters were based on microarray time series datasets. As we mentioned before, DNA microarray data is noisy, and can contain missing values. Moreover, obtaining large number of samples in time series data is usually difficult. In the next chapter, to alleviate the effect of these limitations, we focus our attention on incorporating additional information into the reconstruction process. For this, we propose a novel approach that uses multiple sources of prior knowledge and protein-protein interaction information with a view to integrating these into the GRN reconstruction technique.

Chapter 6

Co-Learning of GRN and PPIN

6.1 Introduction

So far, all the efforts towards the reconstruction of gene regulations has been focused on using the available DNA microarray data. As has been pointed out previously, one of the main obstacles in deciphering the regulatory relationships is the lack of availability of sufficient data - both in terms of quality and quantity. The microarray data is inherently noisy, and moreover, the number of samples from microarray is very low. Further, the problem is also compounded by the presence of missing values. Due to these difficulties, it would be useful for the GRN reconstruction process to use additional sources of information rather than relying on microarray data alone.

Attempts to use prior knowledge from location binding data have already been reported [24] for better reconstructing gene regulatory networks. Information from protein-protein interaction networks (PPINs) has also been used as a source of additional information [160,161]. However, due to the fact that the data for PPI may itself be erroneous, it is often considered appropriate to use diverse knowledge sources for the reconstruction in conjunction with using protein-protein interaction (PPI) data[1]. Genomic data, such as essentiality phenotype information and functional category databases, are considered important in this regard [104,161]. In this chapter, we present an algorithm for the reconstruction of gene regulatory networks that incorporates the knowledge obtained from PPINs and diverse sources of information to improve network accuracy.

[1]Meaning in addition to PPI data, other information sources should be used during PPIN reconstruction.

The rest of the chapter is organized as follows. In Section 6.2, we discuss related approaches which have a similar objective as ours, i.e., simultaneous use of DNA microarray and PPI data. Section 6.2.1 describes various formalizations used to model PPINs. We use the Bayes theorem for formulating the joint probability distribution of our proposed fusion based approach, and hence this theorem is presented in Section 6.2.2. Next, we introduce our proposed approach in Section 6.3. Relevant resources that can be used to probabilistically integrate multiple sources of PPI data are presented in Section 6.4. The results and comparison of our approach with other approaches are shown in Section 6.5. Section 6.6 concludes with relevant observations and remarks.

6.2 Background

A number of techniques have been proposed in the literature with a view to using both microarray and PPI data for reconstructing GRNs and PPINs. Segal *et al.* [199] proposed a method for identifying pathways from microarray data and protein-protein interaction (PPI) data. They proposed a clustering method for grouping genes that could be on the same pathway based on microarray data and PPI data. For using PPI information, they used it in a binary manner to indicate whether an interaction is present or not. However, the quality of each protein-protein interaction, which should be quantified according to its reliability, is not considered in the approach. Further, their main objective had been to find co-functioning genes on the same pathway, rather than a fully-fledged GRN-PPIN network reconstruction.

Nariai *et al.* [160] propose a static BN based framework, using the concept of formation of protein complexes (from individual gene pairs), which are formed based on results from principal component analysis (PCA). When a gene is regulated by a protein complex, the authors use virtual nodes corresponding to protein complexes in the BN model. In other words, if $gene_A$ and $gene_B$ make a protein complex and regulate $gene_C$, a new variable $complex_{AB}$ is constructed from the expression data of $gene_A$ and $gene_B$ (by projecting the expression data onto the first principal component). In the BN model, then, we consider the relation $complex_{AB} \rightarrow gene_C$ instead of $gene_A \rightarrow gene_C \leftarrow gene_B$. They use a non-parametric regression based scoring technique using the Laplace approximation for integrals, and greedy hill-climbing techniques for the search. However, it is

difficult to interpret the results to know about whether the estimated causal relationships show gene regulations or protein-protein interactions [161]. Also, the proposed greedy algorithm only merges protein pairs based on PCA. Modeling a larger protein complex in the gene network is an important problem which is not dealt with by the approach. Further, the proposed approach is not flexible enough to use multiple sources of prior knowledge such as DNA-protein interactions, binding site information, and so on [160].

The improvement of the method, presented in [161], considers it as a three component model, consisting of: a GRN part (using Bayesian networks), a PPIN part (undirected, using binary Markov networks), and a structural connection between these two. The last part realizes the connection between gene regulatory networks and PPINs, giving a penalty to coexistence of a directed edge and an undirected edge between genes. Noting that physically interacting proteins are often coexpressed [161], this approach tries to overcome the drawback of estimating the coexpressed relationship as a gene regulation instead of a protein-protein interaction. These three components are combined as one statistical model under a Bayes statistics in order to distinguish gene regulations from protein-protein interactions clearly in the estimated network. Using B-spline based non-parametric regression and likelihood ratio [104] based approximation of binary Markov networks, this approach maximizes the joint posterior probability with a view to getting the optimal GRN and PPIN. However, similar to their first approach [160], the method works on static Bayesian networks. As a result, they cannot properly use the dynamics information available from time series data. Also, the model uses parameters for controlling the balance between microarray and PPI data, which need to be set up heuristically, and there is no theoretical means of determining the optimal value of the parameter. Finally, the calculation of likelihood ratio for probabilistically assessing confidence in protein interaction pairs is done in a way that gives rise to divide by zero problems. As a result, the whole process may get dominated by only a few high confidence arcs.

Chaturvedi *et al.* [39] model time-delayed gene interactions using a skip-chain based dynamic Bayesian network model, that finds missing edges between non-consecutive time points based on knowledge from PPIN using Viterbi approximation. However, the method does not work with multiple sources of prior knowledge (e.g., both PPI data and TF binding location data). Further, knowledge sources for GRNs and PPINs might contain noisy information and thus unlike Chaturvedi *et al.* [39], it is better to consider the

information probabilistically. Again, Hartemink *et al.* [24] use information from TF binding location data probabilistically, but the method neither uses multiple sources of prior knowledge, nor does it use information from the available PPI data. Hence, it happens to be a purely GRN based technique, and does not deal with PPI networks.

In this chapter, we propose a probabilistic framework for jointly constructing a GRN and a PPIN. We use information from multiple sources of prior knowledge (PPI data, functional category data, essentiality phenotype information etc.) probabilistically. Although similar, the proposed approach has major differences from the work of Nariai *et al.* [161]. One of the main differences is the formulation of the objective function. Rather than using a *conjunctive* approach of formulating the objective function, like the one in Nariai *et al.* [161], we employ a *disjunctive* approach, noting that given the PPIN, the GRN depends only on the microarray, and does not depend on the PPI data (and vice versa). Further, compared to [161], where the joint posterior probability of the whole system (consisting of all the three parts mentioned above) is optimized, our approach in effect has two parts (the GRN part and the PPIN part, with our information fusion approach effectively eliminating the need for the structural connection part of Nariai *et al.* [161]). For each part, we make use of information from the other part, but optimize the posterior probability separately (it works iteratively, and uses the current estimation of the GRN for the estimation of the PPIN for the next stage, and vice versa). This disjunctive approach essentially allows us to work in parallel for the PPIN and GRN construction, while maintaining coherent and flexible fusion of information among the parallel threads. The approach is efficient and naturally amenable to parallel computation. This also has the advantage that effectively we have to deal with roughly half of the structure space (considering PPI networks are non-directional) compared to approaches where both networks are considered simultaneously. Because our method marginalizes over the parameters, we do not need to include the balance parameter during PPI network's posterior probability calculation, as is done in Nariai *et al.* [161]. Finally, the calculation of confidence scores for individual protein-protein interactions is done in a way that effectively eliminates the problem of only a few high scoring arcs dominating the whole search process. We show the effectiveness of our approach by using different networks from yeast.

6.2.1 Representing Protein-Protein Interactions: Binary Markov Networks

PPIN interactions can be conveniently represented by Binary Markov networks, which are a type of undirected graphical model. As has been discussed previously, probabilistic graphical models use a graph based representation as the basis for compactly encoding a complex distribution over a high-dimensional space. Similar to Bayesian networks, in this graphical representation, the nodes correspond to the variables (proteins) in our domain, and the edges correspond to direct probabilistic interactions between proteins. An example of a Markov network structure is shown in Figure 6.1.

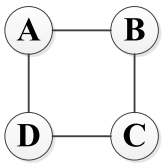

Figure 6.1: Graphical representation of a Markov Network for representing PPINs

There is a dual perspective to interpret the structure of this graph. From one perspective, the graph is a compact representation of a set of independencies that hold in the distribution; these properties take the form A is independent of B given C, denoted $(A \perp B|C)$, for a subset of variables A, B, C. For example, The independence relations induced from Figure 6.1 are: (*i*) $(A \perp C|B, D)$, and (*ii*) $(B \perp D|A, C)$. Also, similar to Bayesian networks, the other perspective of Markov networks is that the graph structure defines the factorization of a distribution P associated with it (i.e., the set of factors and the variables that they represent). The graph, in effect, defines a skeleton for compactly representing a high-dimensional distribution: rather than encoding the probability of every possible assignment to all of the variables in the factor domain, the distribution can be "broken up" into smaller factors, each over a much smaller space of possibilities. We can then define the overall joint distribution as a product of these factors. For example, the factorization induced by the graph in Figure 6.1 is:

$$P(A, B, C, D) = \frac{1}{Z} f_1(A, B) \times f_2(B, C) \times f_3(C, D) \times f_4(A, D) \tag{6.1}$$

Formally, given a set of n nodes, let Y denote any random graph on those nodes and y denote a particular graph on those nodes. A general form for binary Markov networks

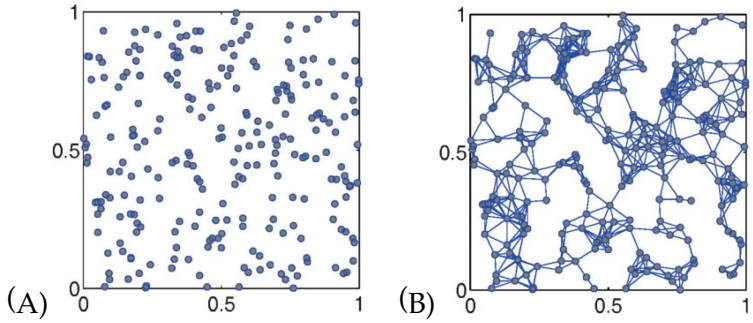

Figure 6.2: Illustration of GEO. (A) 250 points in the unit cube. (B) The resulting geometric graph with a cut-off distance of 0.1. Source: [174].

can then be defined as follows [31, 103, 199]:

$$P(Y = y) = \frac{1}{Z(\theta)} e^{\sum_t \theta_t s_t(y)} \tag{6.2}$$

where θ_t is the unknown parameter related to $s_t(y)$, and $s_t(y)$ is a known vector of graph statistic (of type t) on y. $Z(\theta)$ is the normalizing constant, which ensures that the probabilities sum to unity. The quantities θ are the unknown regression coefficients. Calculation of the $Z(\theta)$ quantity is particularly difficult and intractable because it is defined over the entire graph search space. Albeit difficult, for our current problem the $Z(\theta)$ quantity is essentially constant for all possible networks, and thus it can be safely ignored when we do comparison based network searching.

Alternative approaches to PPIN modeling exist. For example, Pržulj [174] proposed a biologically motivated model for PPINs (called GEO), based on the concept of "geometric graphs". To explain the idea of geometric graph, let us assume we have a collection of points distributed in space. We pick a constant distance ϵ and say that two points are "related" if they are within a distance ϵ of each other. This relationship can be represented as a graph, where each point in space is a node and two nodes are connected if they are within distance ϵ. This is called a "geometric graph"; if the points are distributed at random, then it is a "geometric random graph" (see Figure 6.2).

To build a GEO that corresponds to the PPIN, Pržulj used a function of the shortest path length between proteins, as the distance. It was subsequently refined to fit PPINs even better: by learning the distribution of proteins in the embedding space, or by replicating the principles of gene duplications and mutations in a geometric space [174]. From

a conceptual point of view, the reason behind the good fit of GEO to PPINs lies in the observation that all biological entities, including genes and proteins as gene products, exist in some multidimensional biochemical space (although it is difficult to hypothesize about the nature or dimensionality of that space). Gene duplications and mutations, when modeled in the biochemical space, a duplicated gene starts at the same point in the space as its parent and then natural selection acts on either to eliminate one, or cause them to slowly separate in the space. This means that the child inherits some of the interactors of its parent, while possibly gaining new connections as well. The further the child is moved away from its parent in this abstract space, the more different are their biochemical properties. As noted, it is difficult to substantiate this concept and, more importantly, there is no provision for incorporating additional sources of knowledge in this approach.

In the next section, we present our proposed approach for fusing multiple sources of prior knowledge to the GRN reconstruction process. The mathematical results which we use for the algorithm, makes use of the Bayes theorem. Hence, in the next subsection, we present this theorem.

6.2.2 The Bayes Theorem

The Bayes theorem can be used to compute conditional probability of dependent events, and expresses how a subjective degree of belief should rationally change to account for evidence. Suppose we have q events $E_1, E_2, \ldots, E_q$ such that

$$E_i \cap E_j = \varnothing, \quad for\ i \neq j$$

and

$$E_1 \cup E_2 \cup \ldots \cup E_q = \Omega$$

Such events are called mutually exclusive and exhaustive. To compute the conditional probability of such events, we need to use Bayes Theorem:

Theorem 6.2.1. ***Bayes Theorem.*** *Given two events E and F such that $P(E) \neq 0$, and $P(F) \neq 0$, we have*

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)} \tag{6.3}$$

Furthermore, given q mutually exclusive and exhaustive events $E_1, E_2, \ldots, E_q$ *such that* $P(E) \neq 0$ *for all* i*, we have for* $1 \leq i \leq q$*,*

$$P(E_i|F) = \frac{P(F|E_i)P(E_i)}{P(F|E_1)P(E_1) + \ldots P(F|E_q)P(E_q)} \tag{6.4}$$

The first equation in the Bayes Theorem enables us to compute $P(E|F)$ if we know $P(F|E)$, $P(E)$, and $P(F)$; the second equation enables us to compute $P(E_i|F)$ if we know $P(F|E_j)$ and $P(E_j)$, for $1 \leq j \leq q$.

6.3 Fusion of Gene Regulatory Networks and Protein-Protein Interaction Networks

Consider Figure 6.3, where the symbols D_r, D_p, G_r and G_p denote GRN data, PPI information, GRN and PPIN, respectively. The dashed arrows among gene regulatory networks (GRN) and PPINs denote transfer of structural information between the corresponding structures.

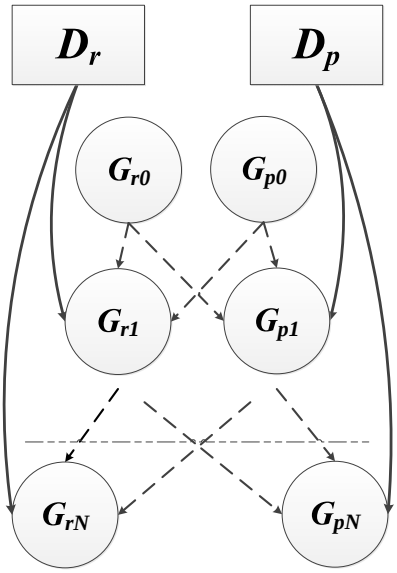

Figure 6.3: Schematic of the fusion based co-learning approach

From the figure, noting that G_r depends only on D_r and G_p, and similar for G_p, and using the Bayes theorem and the law of conditional probability, the posterior probability of the gene regulatory networks and protein interaction networks can be defined by the following formula:

$$P(G_p|\{G_r, D_p\}) \propto \frac{P\left(G_p\right) P(D_p|G_p) P\left(G_r|G_p\right)}{P(G_r)} \tag{6.5}$$

and similarly for G_r,

$$P(G_r|\{G_p, D_r\}) \propto \frac{P\left(G_r\right) P(D_r|G_r) P\left(G_p|G_r\right)}{P(G_p)} \tag{6.6}$$

Using the above relationships, we can optimize the posterior probability iteratively. In this chapter, we propose an evolutionary computation based iterative fusion/co-learning algorithm that achieves this task. We will describe how we calculate different quantities in these equations, and then provide algorithms that can optimize these quantities.

First, we describe how the $P(D_r|Gr)$ quantity is calculated, using multinomial conditionals and Dirichlet priors. Assuming parameter independence, it can be defined as follows:

$$P(D_r|G_r) = \prod_{i=1}^{p} \prod_{j=1}^{q_i} \frac{\lceil \eta_{ij} \rceil}{\lceil \eta_{ij}+N_{ij} \rceil} \prod_{k=1}^{r_i} \frac{\lceil \eta_{ijk}+N_{ijk} \rceil}{\lceil N_{ijk} \rceil} \tag{6.7}$$

where N_{ijk} is the number of observations in which gene X_i takes the value k, given that $Pa(X_i)$ has configuration j; q_i is the number of possible configurations of parents $Pa(X_i)$; and r_i is the number of possible values of X_i. η_{ijk} are Dirichlet hyperparameters. Finally,

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{6.8}$$

and

$$\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk} \tag{6.9}$$

Regarding the prior probability of G_r under a given G_p, it can be defined as follows [161]:

$$P(G_r|G_p) \propto e^{-\sum_{e(i,j)\in G_r} \xi_{c_{ij}}} \tag{6.10}$$

where

$$c_{ij} = \begin{cases} 1 \quad if \quad e\{i,j\} \in G_p \\ \\ 2 \quad if \quad e\{i,j\} \notin G_p \end{cases} \tag{6.11}$$

Similarly, the inverse ($P(G_p|G_r)$) can be defined as:

$$P(G_p|G_r) \propto e^{-\sum_{e(i,j)\in G_p} \xi_{c_{ij}}} \tag{6.12}$$

where

$$c_{ij} = \begin{cases} 1 \quad if \quad e(i,j) \in G_r \\ \\ 2 \quad if \quad e(i,j) \notin G_r \end{cases} \tag{6.13}$$

and the quantities $e(i,j)$ and $e\{i,j\}$ denote edges between genes X_i and X_j in the GRN and the PPIN, respectively.

Before we calculate $P(D_p|G_p)$, let us define a few relevant quantities that are required in its calculation. First, we discuss the concept of likelihood ratio [104]. As a measure of reliability, the overlap of information sources (i.e., "interaction datasets", which could either be noisy experimental data or sets of genomic features) with the gold-standards can be expressed in terms of a "likelihood ratio". For example, consider a genomic feature f expressed in binary terms (i.e., 'present' or 'absent'). The likelihood ratio $L(f)$ is then defined as the fraction of gold-standard positives having feature f divided by the fraction of negatives having f. For two features f_1 and f_2 with uncorrelated evidence, the likelihood ratio of the combined evidence is simply the product[2] $L(f_1, f_2) = L(f_1) \times L(f_2)$. The combined likelihood ratio is thus proportional to the estimated odds that two proteins are in the same complex, given multiple sources of information.

[2]For correlated evidence, $L(f_1, f_2)$ cannot be factorized in this way.

Now, we formally define likelihood ratio. Assuming $y_{ij}(k)$ is an element of D_p that shows a genomic feature of protein pair X_i and X_j, the reliability of the protein-protein interaction between $gene_i$ and $gene_j$ is then given by the likelihood ratio:

$$L(i,j) = \frac{P(y_{ij}(1), ..., y_{ij}(N)|pos)}{P(y_{ij}(1), ..., y_{ij}(N)|neg)} \tag{6.14}$$

where '*pos*' and '*neg*' are respectively the positive and negative sets of protein pairs constructed in advance, and N is the number of genomic features that we consider. If each genomic feature is conditionally independent, the likelihood ratio can be re-written as:

$$L(i,j) = \frac{P(y_{ij}(1)|pos)}{P(y_{ij}(1)|neg)} ... \frac{P(y_{ij}(N)|pos)}{P(y_{ij}(N)|neg)} \tag{6.15}$$

Now, noting that the likelihood ratio provides noisy evidence regarding the existence of edges among proteins, we define the $l-$value of an edge $e\{i,j\}$ on the interval $[0,1]$, which is inversely related to the probability of an edge being present in the true PPIN. Formally,

$$l(i,j) = \frac{\min_{i,j}\{L(i,j)\}}{L(i,j)} \tag{6.16}$$

Based on this definition, we define:

$$\beta(i,j) = \frac{\lambda e^{-\lambda l(i,j)}}{\lambda e^{-\lambda l(i,j)} + 1 - e^{-\lambda}} \tag{6.17}$$

where λ is the parameter controlling the scale of the truncated exponential distribution, and acts as a tunable parameter indicating the degree of confidence in the evidence provided by the prior knowledge. As the parameter λ increases, the mass of this distribution becomes more concentrated at smaller values of $l(i,j)$; conversely, as λ decreases, the distribution spreads out and flattens. Rather than using 6.17 in the raw format, we use marginalization [24] over the parameter λ, to get:

$$\beta(i,j) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_H}^{\lambda_L} \frac{\lambda e^{-\lambda l(i,j)}}{\lambda e^{-\lambda l(i,j)} + 1 - e^{-\lambda}} d\lambda \tag{6.18}$$

which is numerically tractable.

Since we have a finite set of $L(i,j)$ values, we can pre-compute these integrals for each $L(i,j)$ value and store the results in memory for later use. The computational overhead associated with marginalizing over λ is thus constant. The net effect of marginalization is an edge probability distribution that is a smoother function of the reported $\beta(i,j)$ values than without marginalization. This results in a much heavier tailed distribution [24], which is advantageous. Also, using this technique, even when the '*neg*' set is empty, we don't have a likelihood value of infinity (unlike the approach of Nariai *et al.* [161]), so only a few high confidence interactions cannot dominate the whole search process.

Based on these definitions, we can now define the probability $P(D_p|G_p)$:

$$P(D_p|G_p) \propto \prod_{e\{i,j\}\in G_p} \beta(i,j) \tag{6.19}$$

Since the $\beta(i,j)$ values are marginalized probabilities, we do not need to use parametrized values of the likelihood ratio ($L(i,j)^{\alpha}$). This novel approach reduces the number of parameters that we need to consider during computation, and thus saves computation time (unlike the method described in Nariai *et al.* [161]).

Finally, the prior probability of G_p is defined to encourage sparsity, using the following equation:

$$P(G_p) \propto e^{\sum\limits_{e(i,j)} \xi_p} \tag{6.20}$$

Based on the above definitions, we propose an iterative Bayesian co-learning algorithm, *FusGP*. A schematic of the overall execution of the parallel algorithm is shown in Figure 6.4. The algorithm first generates initial estimates for the GRN and PPIN. The estimates are then fed to a routine called *FGP*, which performs the task of co-learning based fusion. In the next subsection, we describe in detail the working procedure of the algorithm.

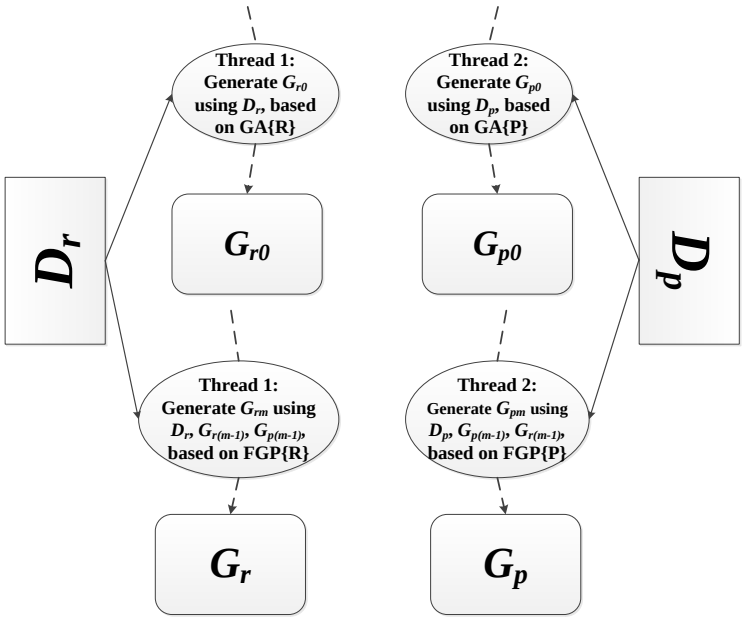

Figure 6.4: Parallel execution of the FusGP algorithm

6.3.1 The Search Strategy for Initial Network Generation

A genetic algorithm (GA), applied to explore this structure space, begins with a sample population of randomly selected network structures and their fitness calculated. Iteratively, crossovers and mutations of networks within a population are performed and the best fitting individuals of the population are kept for future generations.

During crossover, two random edges are chosen and swapped. Mutation is applied on a randomly chosen individual edge of the network. For our study, we incorporate the following three types of mutations:

1. Deleting a random edge from the network.
2. Creating a random edge in the network.
3. Changing direction of a randomly selected edge.

The genetic algorithm can be called with either of two parameters: R and P. If it is called with the R parameter, this means it is supposed to build a GRN, given the PPIN. However, the P parameter denotes constructing a PPIN. In this case, the operations that the GA can perform become restricted (e.g., Markov networks are non-directional; during

1. Create initial population of network structures (100 in our case). If input is R, for each individual, genes and set of parent genes are selected based on a Poisson distribution and edges are created. On the other hand, if input is P, random binary Markov networks are generated.
2. Evaluate each network and sort the chromosomes based on the fitness score. If input is R, use 5.4 for fitness calculation. If input is P, use 6.19 and 6.20 (to get the posterior) for fitness calculation.
   (a) Generate new population by applying crossover and mutation on the previous population. If input is P, use only the first two operations for mutation. Otherwise, use all three possible operations.
   (b) Evaluate each individual using the fitness function and use it to sort the individual networks. If the best individual score has not increased for 5 consecutive times, aggregate the 5 best individuals using a majority voting scheme.
   (c) Take best individuals from the two populations based on fitness score and create the population of elite individuals for next generation.
3. Repeat steps a) - c) until the stopping criteria (400 generations in our case) is reached.

   When the GA stops, take the best chromosome and reconstruct the initial network (G_{r0} or G_{p0}) to be used for subsequent computation. ◁

Table 6.1: Genetic algorithm for GRN and PPIN co-learning

mutation, this needs to be taken into consideration). Keeping this in mind, the overall genetic algorithm for the stochastic search of the network space is shown in Table 6.1.

6.3.2 The Algorithm for Co-Learning, *FGP*

After the initial networks have been generated, we start the co-learning of the two networks in two parallel threads. Based on 6.5 and 6.6, at each step (m) of iteration, one of the quantities in 6.21 and 6.22 are calculated (based on the input parameter).

$$P(G_{pm}|\{G_{r(m-1)}, D_p\}) \propto \frac{P\left(G_{pm}\right) P(D_p|G_{pm}) P\left(G_{r(m-1)}|G_{pm}\right)}{P(G_{r(m-1)})} \tag{6.21}$$

$$P(G_{rm}|\{G_{p(m-1)}, D_r\}) \propto \frac{P(D_r|G_{rm}) P\left(G_{p(m-1)}|G_{rm}\right)}{P(G_{p(m-1)})} \tag{6.22}$$

Based on the observation that a PPIN cannot give any direction information for use in a GRN, there are only two possible operations at any particular stage of the algorithm: adding an edge if it was not there, and vice versa (flip operation). In a similar manner,

1. Evaluate the network from the previous iteration. If input is R, evaluate score of $G_{r(m-1)}$ based on $G_{p(m-1)}$, using 6.6 for the calculation. If input is P, score $G_{p(m-1)}$ based on $G_{r(m-1)}$, using 6.5.
2. Generate new network by applying flip operation on each possible edge. Store the score of the changed network, calculated using 6.21 (input is P) or 6.22 (input is R).
3. Find the changed network with the maximum score. Keep this as the new "best solution" if score increases compared to the "best solution" from the previous iteration. Otherwise, set the "best solution" from the previous iteration as the new "best solution" and send an "end flag" to the counterpart thread.
4. Repeat steps 1) - 3) until the stopping criteria (new "best solution" is same as previous "best solution", or an "end flag" from the counterpart thread) is reached. ◁

Table 6.2: Algorithm *FGP*

since the direction information of a GRN is not useful while constructing a PPIN, only flip of edge existence operations are permitted while constructing PPINs. The overall algorithm is shown in Table 6.2.

As has been discussed, integrating diverse sources of PPI data is necessary for a successful fusion of GRN and PPIN. In the next section, we discuss two data sources which can be used to probabilistically integrate multiple PPI data sources.

6.4 Probabilistic Aggregation of Multiple Sources of PPI Data

Lee *et al.* [129] developed a conceptual framework for integrating diverse functional genomics data by re-interpreting experiments to provide numerical likelihoods that genes are functionally linked. This allows direct comparison and integration of different classes of data. In the framework, functional genomics datasets are first benchmarked for their relative accuracies; these are used as weights in a probabilistic integration of the data. Several raw datasets already have intrinsic scoring schemes. These data are rescored with a log-likelihood based scoring called LLS, then integrated into an initial network (called "IntNet"). Lee *et al.* used 8 different sources to construct the initial integrated network, including physical and gcnetic interaction data sets, mRNA co-expression linkages, functional linkages from literature mining, and computational linkages from two comparative genomics methods, Rosetta stone (gene fusion) linkages and phylogenetic profiles. Additional linkages from the genes' network context using functional genomics data produce

a "ContextNet", which are then integrated with the "IntNet" to create a final network, "FinaINet". Thresholding techniques (scoring higher than the gold-standard small-scale assays of protein interactions) are then applied on the "FinalNet" to produce a "ConfidentNet", with ~ *34,000* linkages between *4,681* genes. Hierarchical clustering of "ConfidentNet" defined 627 modules of functionally linked genes spanning $3,285$ genes (called "ModularNet"), approximating the set of cellular systems in yeast.

Jansen *et al.* [104] proposed a Bayesian approach for integrating interaction information that allows for the probabilistic combination of multiple datasets and demonstrated its application to yeast [67]. This approach can be used for combining noisy interaction data sets and for predicting interactions *de novo*, from other genomic information. The basic idea behind the approach is to assess each source of evidence for interactions by comparing it against samples of known positives and negatives ("gold standards"), yielding a statistical reliability. Then, extrapolating genome-wide, the chance of possible interactions for every protein pair can be predicted by combining each independent evidence source according to its reliability. The authors term the results as "probabilistic interactomes" (PIs), in which each protein pair is associated with a probability measure for being in the same complex (called likelihood ratio). The procedure not only allows combining existing experimental interaction datasets (resulting in a PI-experimental or "PIE"), but also the *de-novo* prediction of protein complexes from genomic datasets (when the input data are not interaction datasets per se, resulting in a PI-predicted or "PIP").

Jansen *et al.* combined four interaction datasets from high-throughput experiments into the PIE [82, 94, 101, 231] dataset. PIP was computed from several genomic data sources: the correlation of mRNA amounts in two expression datasets (one with temporal profiles during the cell cycle, one of expression levels under 300 cellular conditions), two sets of information on biological function, and information about whether proteins are essential for survival [14, 49, 97, 146]. For computing the PIE and the PIP, two different types of Bayesian networks were used: a "naive" network for the PIP and a fully connected one for the PIE [104].

Finally, the PIP, PIE, and gold standard were combined into a total PI (PIT), which represents a comprehensive view of the known and putative protein complexes in yeast

[183]. Because the PIP and PIE data provide essentially uncorrelated evidence for protein-protein interactions, a naive network was chosen to construct the PIT. The authors verified their predictions by comparing them against existing experimental interaction data (not in the gold standard) as well as new TAP (tandem affinity purification) tagging experiments.

6.5 Experimental Results

Since the PPI information needed to build the knowledge base is not available for organisms like *E. coli* and cyanobacteria, in this chapter we consider yeast (*Saccharomyces cerevisiae*) networks for the evaluation of the proposed approach. We consider two different networks, the KEGG pathway of yeast consisting of 11 genes, and the genetically modified network of yeast called IRMA. For the KEGG network, we used the cell cycle data obtained from Spellman *et al.* [214]. Both these networks have been used in the previous chapter for assessment of reconstruction techniques. For all the experiments, the λ_H and λ_L parameters were set to 1 and 1000 respectively, to avoid problems near terminal values [24]. The ζ_p values were set to the log of the cut-off parameter (calculated to be 600) obtained from the PPIN datasets of Jansen *et al.* [104], which were used in this study. Finally, to set the ζ_1 and ζ_2 parameters, we made the practical assumption that physical protein-protein interactions should be considered as part of PPIN instead of GRN, making them mutually exclusive [161], and as a result ζ_1 can be set to 0 and ζ_2 to ∞. To compare our algorithm, we consider two other methods, namely, BANJO [255] and BNFinder [246] (with both BDe and MDL).

6.5.1 Yeast KEGG Pathway Reconstruction

In order to validate the proposed method's performance on yeast *S. cerevisiae* cell cycle, we selected the eleven gene network of the G1-phase: CLN3, CDC28, MBP1, SWI4, CLB6, CDC6, SIC1, SWI6, CLN1, CLN2, CLB5. The data used was obtained from the *cdc28* experiment of Spellman *et al.* [214].

In Figure 6.5(B)-(F), we report network graphs reconstructed by our proposed approach, CCIT-based approach, BNFinder(BDe and MDL) and BANJO. We also report the

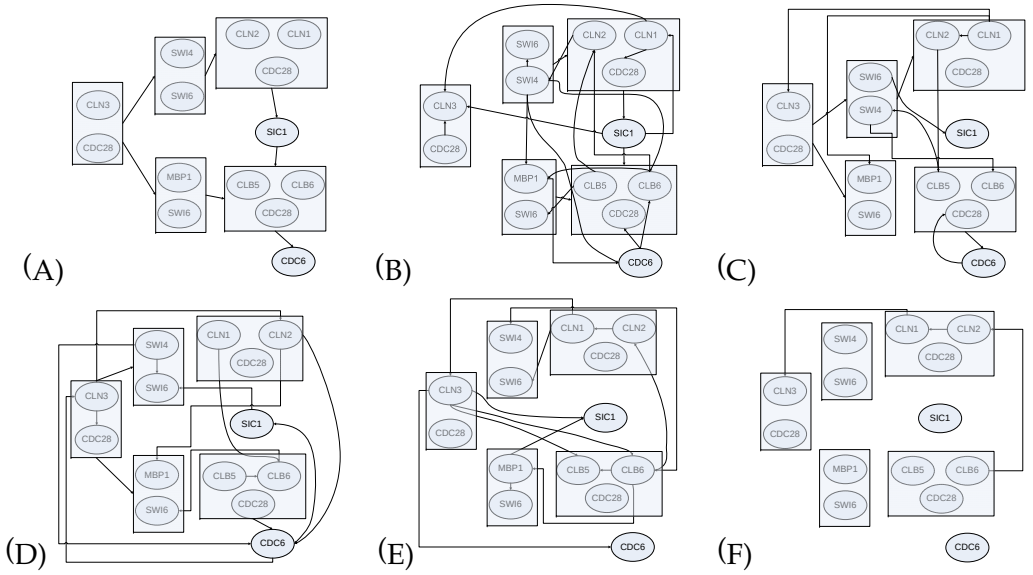

Figure 6.5: Reconstruction of Yeast KEGG Pathway [108]. (A) Target Network. (B) Network Inferred by FusGP. (C) Network Inferred by CCIT based method. (D) Network Inferred by BANJO.(E) Network Inferred by BNFinder+BDe. (F) Network Inferred by BNFinder+MDL.

KEGG pathway [108] of the cell cycle in yeast in Figure 6.5(A). Since the exact ground truth for this network is not known, instead of applying performance measures as a means of determining network accuracy, we refer to the available correct interactions obtained from the KEGG pathway [108] and identify which of the predicted interactions are correct or otherwise. We observe that our approach correctly identifies the regulation of CLN1-CLN2-CDC28 complex by the SWI4-SWI6 complex. Also, the proposed approach infers that the MBP1-SWI6 complex regulates the CLB5-CLB6-CDC28 complex, which is correct. Some other interactions inferred by our approach (CLN1→CDC28, CDC28→CLN3 and SWI4→SWI6) are also correct based on the KEGG pathway. Overall, we observe that none of the methods perform particularly well on this network. However, the number of correct predictions by our method (7) is higher than the other methods (the second best among these methods, the CCIT-based approach, inferred 5 correct interactions).

6.5.2 Real-life Biological Data of yeast, IRMA

To validate our method with another real-life biological gene regulatory network, we investigate the recent network reported in Cantone *et al.* [35] called IRMA, which, in

addition to the original network, contains a 'simplified' network, ignoring some protein level interactions. Since the exact ground truth for this network is known, we use the four well known performance measures used earlier, namely sensitivity (Se) , specificity (Sp), precision (Pr), and F-score (F), to assess the algorithms.

IRMA ON Dataset

The performance comparison amongst various method based on the ON dataset is shown in Table 6.3. We observe that FusGP outperforms the other methods in terms of all the performance measures. In terms of specificity and and precision, it ties with the CCIT-based approach. Also, the increase in sensitivity compared to the CCIT-based approach implies that it correctly learns additional interactions from the protein-protein interaction network.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
FusGP	**0.75**	**1.0**	**1.0**	**0.86**	**0.83**	**1.0**	**1.0**	**0.91**
CCIT	0.63	**1.0**	**1.0**	0.77	0.67	**1.0**	**1.0**	0.80
TDARACNE	0.63	0.88	0.71	0.67	0.67	0.90	0.80	0.73
BNFinder+BDe	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BNFinder+MDL	0.13	0.82	0.25	0.17	0.17	0.80	0.33	0.22
BANJO	0.25	0.76	0.33	0.27	0.50	0.70	0.50	0.50

Table 6.3: Performance comparison of FusGP based on IRMA ON dataset

IRMA OFF Dataset

The OFF dataset lacks the presence of 'stimulus' (applied during the experiments); however, it contains more samples compared to the ON dataset (21 versus 16). The comparison of the performance of the algorithms using the OFF dataset is shown in Table 6.4. We observe from the results that FusGP and CCIT have the same performance. In fact, for this dataset having larger number of samples, the information from the PPIN cannot sufficiently influence the score of the GRN, and thus the resultant GRN essentially remains the same.

	Original Network				**Simplified Network**			
	Se	***Sp***	***Pr***	***F***	***Se***	***Sp***	***Pr***	***F***
FusGP	0.50	**0.94**	**0.80**	**0.62**	0.50	**0.90**	**0.75**	**0.60**
CCIT	0.50	**0.94**	**0.80**	**0.62**	0.50	**0.90**	**0.75**	**0.60**
TDARACNE	**0.60**	-	0.37	0.46	**0.75**	-	0.50	**0.60**
BNFinder+BDe	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BNFinder+MDL	0.13	0.82	0.25	0.17	0.33	0.80	0.50	0.40
BANJO	0.38	0.88	0.60	0.46	0.33	0.90	0.67	0.44

Table 6.4: Performance comparison of FusGP based on IRMA OFF dataset

6.6 Summary

In this chapter, we proposed a novel approach for fusing the knowledge from PPINs to GRNs, and vice versa. We also proposed an algorithm that executes in parallel threads to achieve a coherent transfer of information during the building of the GRN and PPIN. Experiments were carried out using different real-life networks of yeast and the results showed that, in most cases, the information from the PPIN has a positive effect on the reconstruction accuracy of the GRN method, providing support to the superiority of our approach. The protein-protein interaction information for other organisms such as *E. coli* and cyanobacteria is not rich at this stage, which prohibited us from studying these organisms. However, when this information for these and other organisms becomes available in future, the proposed approach can easily be extended to construct accurate GRNs for these organisms. Along with the extensions proposed in this chapter, the CCIT and mDBN-based method of inferring both instantaneous and multi-step time-delayed interactions would improve the accuracy of gene regulatory network reconstruction and thus induce further research in systems biology.

Chapter 7

Conclusion

Gene regulatory networks (GRN) depict the regulatory interactions among the genes in a living cell. A system level view of gene functionalities provided by these networks is very important in understanding biological processes. This thesis is devoted to realistic modeling and accurate reconstruction of gene regulatory networks using dynamic Bayesian network (DBN) and information theoretic measures. Dynamic Bayesian network as the modeling technique offers various benefits when used in conjunction with the noisy and uncertain DNA microarray data. Relevant standalone features of mutual information (MI) and conditional mutual information (CMI) contribute significantly in assessing regulatory relations.

Using the DBN modeling technique and the association measures, we first proposed a novel algorithm called BITGRN (Bayesian Information Theoretic GRN Reconstruction Algorithm) that uses MI/CMI based conditional independence (CI) tests for inferring regulatory relations. The algorithm provides a superior performance, since at each point of the execution of the algorithm, it uses the current estimate of the parents of a gene under consideration, to build the condition set for CMI calculation. However, the algorithm was only suited to detecting time-delayed interactions. Thus, after evaluating its performance, we extended the basic DBN modeling technique to include both instantaneous and time-delayed interactions. We proposed two approaches, which can use the new modeling framework, and learn the two types of interactions, albeit sequentially. The first technique, called GRNCIT (learning GRNs with Contemporaneous arcs using Information Theory), uses the properties of MI/CMI in a hill-climbing fashion for CI tests (i.e.,

the approach of BITGRN), and the later technique, called GRNCGA (learning GRNs with Contemporaneous arcs using Genetic Algorithm), uses similar principles but employs a score and search strategy. We then improved these learning techniques of GRNCIT and GRNCGA, by proposing a novel scoring metric called CCIT (Combined Conditional Independence Tests), that can score the two types of interactions simultaneously. Moreover, a GA based search algorithm was devised that uses the scoring metric and the concept of network motifs for effectively exploring a significantly larger search space.

Due to the scarcity of the number of samples in DNA microarray data and also because the data contains noise and missing values, it is considered appropriate to supplement the DNA microarray data using other diverse data sources. For this, we used protein-protein interaction information and various other genomic data. We devised a novel co-learning based fusion algorithm that learns a GRN and a PPIN in an iterative manner, and exchanges structural information between the GRN and PPIN during this process. The methods developed in the thesis do not depend on the actual time difference of the individual samples; hence it is not affected by samples taken at irregular time intervals. As a result, it can be adapted to deal with irregular time intervals.

In terms of performance, we have shown that the method performs well for small to medium scale and also showed results for a relatively large network containing 730 genes. For still larger networks (say, network containing more than 1000 genes), the performance will depend on the number of samples and their quality. As long as the datasets used for the experiments are of good quality, the methods should perform similar to the results for the medium sized network. In terms of computational cost, it can be noted that all the methods developed in the thesis makes use of the MI and CMI, and then uses these measures in intelligent ways, either in an iterative manner (in which case if the network has n genes and t samples, we have to compute $O(Kn^2)$ estimations of the mutual information between two vectors of samples having t samples or less, K being the maximum order of the DBN, which is less than 3), or in evolutionary manner, where the computational cost is proportional to the number of functional evaluations per generation ($O(Mn^2)$ evaluations, where $M = d.K$, K being the number of chromosomes). Since we have restricted the number of generations, the functional evaluation is also bounded and hence the computational cost is not high in any case.

It may be noted here that the largest synthetic network we have done our analysis on contains 50 genes. Although we can generate larger synthetic networks, the current tools used for generating synthetic networks usually cannot adequately replicate the real dynamics of large-scale biological systems. Also, some methods perform better on synthetic networks but do not perform well for larger real-life networks. Hence, instead of using large synthetic networks, we have emphasized on using real-life networks of large-scale (e.g., cyanobacteria network, containing 730 genes) to assess the performance of the algorithm. The analyses on real-life biological networks are capable of giving more correct insight on the performance of reconstruction algorithms compared to synthetic networks.

In the following sections, we elaborate on the contributions and observations made during the development and evaluation of the techniques. Finally, we conclude this thesis with future directions for further research.

7.1 Information theory based CI tests for Detecting Time-Delayed interactions

In Chapter 3, we proposed a novel information theory based GRN learning algorithm, called BITGRN. The algorithm iteratively adds parents (transcription factors) to a gene (child) under consideration, and uses MI/CMI based CI tests to assess regulatory interactions. With respect to modeling, although DBNs have been used in various other approaches, those techniques either did not use MI based association measures or, in limited cases when they used MI, it was not applied as a means of CI testing as done in the present research work. They were merely used in a threshold based setup [248], or for learning three node substructures [15, 42]. Similarly, MI has been used by other techniques, e.g., relevance network based approaches, but these were essentially pairwise association based analysis techniques, and none of these methods exploited the strength of MI and CMI as rigorous statistical significance testing tools (i.e., for CI tests). The algorithm was evaluated with the aid of synthetic as well as real-life networks of yeast and *E. coli*, and it was observed that this algorithm performed better than other DBN based techniques, and also compared to model-free techniques (e.g., relevance network based approaches) which use MI.

7.2 Realistic Modeling of Genetic Interactions

In Chapter 4, we postulated that in biological gene regulatory networks, interactions can be time-delayed, or they can be instantaneous. In regular DBN based techniques, it is assumed that interactions are time-delayed. However, they do not consider instantaneous interactions. We proposed a novel modeling framework using the DBN formalism that can model both these types of interactions. To see whether this proposed modeling framework can improve the performance of reconstruction experiments, we devised the following two learning algorithms to learn the two different types of interactions sequentially:

- A greedy hill-climbing approach using the principles proposed in Chapter 3, called GRNCIT. It directly extends the theoretical results employed in Chapter 3, and learns the time-delayed interactions at first, and afterwards learns the instantaneous interactions, conditional on the already constructed time-delayed interaction network. Applying this algorithm improved the number of correct predictions, accompanied with slight increase in the number of incorrect predictions.
- As an alternative way of utilizing the modeling framework proposed in Chapter 4, we employed a GA based score and search technique called GRNCGA, using an enhanced version (modified so that it can score both instantaneous and time-delayed interactions, and adheres to the criteria set out by the modeling framework) of the popular scoring metric called MIT [60]. We again obtained an improved performance, both compared to the BITGRN algorithm (Chapter 3) and also the GRNCIT algorithm (Chapter 4). Although both GRNCIT and GRNCGA performed better overall compared to BITGRN and other related techniques, the GA based approach made fewer incorrect predictions compared to GRNCIT.

7.3 Simultaneous Learning of Instantaneous and Time-Delayed Genetic Interactions

In Chapter 5, we made the following contributions:

- We extended the modeling framework proposed in Chapter 4, to allow it to model multi-step time-delayed interactions. The reason behind this is that only single-step time-delayed interaction implies that all regulations take place with the same amount of time delay. This assumption, although widely used, is restrictive, and by removing this limitation, we are effectively allowing different interactions to have different time delays.

- To allow the learning of instantaneous and time-delayed interactions *simultaneously*, compared to the sequential learning strategy employed by the algorithms in Chapter 4, we proposed a scoring metric called CCIT. The scoring metric makes use of the MI/CMI based CI testing principles, and it is also decomposable.

  We applied the scoring metric using the extended modeling framework on a wide range of networks, and it resulted in a better balance between correct and incorrect predictions, along with a general increase in the number of correct predictions. Alongside small to medium sized networks, we also applied it to a large scale commercially important network of cyanobacteria, and GO (Gene Ontology) and degree distribution based analysis was used to show that, indeed, the approach is capable of making biologically relevant predictions.

- In Chapter 5, we also proposed a new and improved search technique, which is motivated by the fact that with the adoption of multi-step time-delayed interactions, the already large search space of the DBN framework becomes even larger. We used the concept of network motifs to propose a two-stage genetic algorithm that obtains representative solutions from local optimal solutions in the first stage, and combines these solutions in the second stage to obtain the globally optimal/near-optimal solution. We tested the algorithm on a number of networks, and performance improvement was observed, indicating that this idea of network motifs combined with the proposed two-stage searching strategy can be used to effectively explore a significantly larger search space.

7.4 Using Multiple Sources of Prior Knowledge for Supplementing DNA Microarray Data

In Chapter 6, we proposed a novel approach that can integrate diverse sources of prior knowledge to be used in conjunction with DNA microarray data for GRN reconstruction. The proposed algorithm uses a disjunctive approach based on the Bayes theorem, and works in two threads to construct a GRN and a PPIN simultaneously. During each iteration, it constructs an improved estimate of the GRN compared to the previous iteration, using the PPIN from the previous step, and vice versa. We assessed the effectiveness of the algorithm by using a knowledge base where four different PPI data sources and two genomic data sources were combined probabilistically. We observed that this co-learning based algorithm successfully integrates this information, and using this approach increased the inference of correct interactions without an increase in the number of false predictions. In previous approaches, an increase in the number of correct inferences were accompanied by an increase (usually small, but the trend existed) in the number of incorrect inferences. This result essentially confirms that external sources of knowledge indeed enable better reconstruction of GRNs.

7.5 Future Directions

While the thesis has addressed many important issues for realistic representation of genetic interactions and their reconstruction, in this section, we present some selected, potential directions for future research.

- One of the assumptions that is made when using BN/DBN based methods is stationarity, that is, interactions remain existent irrespective of time. However, in practice, interactions may change with time. Therefore, one important way of reconstructing GRNs is using models that have provision for changing the structure over time. Although intuitive, the use of such models has until recently been plagued due to the lack of data. There has been a recent interest in DBN methods that are essentially time varying [182,212,213,259]. However, these approaches are computationally expensive, and hence using these methods for GRN learning becomes infeasible even for moderate sized networks [182,213]. In their 2009 article [213], Song

et al. proposed a kernel re-weighted l_1 regularized auto regressive approach for modeling such time varying DBNs, and showed its application to learning GRNs. However, these methods need to make various regularization assumptions to make them applicable to GRN reconstruction. Hence, further research in this direction can be pursued.

- The time varying DBN models use just single-step time-delayed interactions between genes. Although extending it to multi-step time-delayed interactions can become computationally intensive to the learning routine, a natural and viable extension can be the allowance of both instantaneous and time-delayed interactions to such models, and devising efficient techniques that can learn GRNs using such models.

- During our experiments with co-learning of GRNs and PPINs (Chapter 6), we observed that PPI and other genomic data information is scarce for many organisms. However, such knowledge sources can play a vital role in accurate reconstruction of GRNs. To build such knowledge bases one needs to integrate knowledge from different sources, and compile these into one database. Although such databases are available for several organisms (e.g., for the human genome [193,230] and for *E. coli* [221]), there still remain many organisms for which these integrated knowledge sources are scarce. Building new or improved collections of such knowledge sources will undoubtedly assist in better reconstruction of gene regulatory networks.

- The recent discoveries mentioned in Section 2.1.9 indicate that microRNA molecules may constitute a new layer of regulatory control over gene expression programs in many organisms. The discovery of the roles that miRNA play in conferring robust GRNs, in particular in the case of feedback and feedforward loops is going to be an important research trend.

Appendix A

Using BiNGO for GO Based Enrichment Analysis

In Chapter 5, we used the Cytoscape [204] plugin BiNGO [135] for GO analysis. In this appendix, we will give a step by step tutorial of how it can be done. For this analysis, we need an annotation database, for example the database we used for the 730-gene cyanobacteria network study. Let us name this annotation database as 'AnnoDB.txt'. The input graph should be of a specific format to be recognizable by Cytoscape and BiNGO, and for simplicity we recommend using the .sif format. sif-format networks specify nodes and interactions only, and take the general form: *nodeX* <*relationship type*> ***nodeY***, where *nodeX* is a gene and ***nodeY*** is a set of genes. The tag <*relationship type*> can be any string.

With these inputs, a GO analysis can be done in the following steps:

1. In Cytoscape: import the network (the .sif network file), using *File* → *import* → *network* or *CTRL+L*).
2. Laying out the network using some layout algorithm (Layout menu). The organic (Layout→Yfile→organic) and force directed (Layout→Cytoscape layout) layouts are very useful ones for our study.
3. Start Bingo (Plugin→BiNGO). In BiNGO,
   (a) Enter a name into the cluster name, check ”get cluster from network”, ”over-representation” and ”no visualization”.

(b) In select reference set box choose "use network as reference set".

(c) In select ontology file select "GO_Full".

(d) In select organisms select the annotation database file (AnnoDB.txt).

(e) Leave all other boxes at default (i.e., Hypergeometric test, FDR, significance level 0.05, overrepresented categories after correction).

4. Now we are ready to do the enrichment analysis. Just select the genes/gene groups in the graph that are to be analysed for enrichment and click start BinGO. It will display the functions that are significantly enriched within that group of genes. If the table is empty, then it means that these genes/gene groups do not have any significantly enriched functionalities.

Appendix B

Comparing CCIT Performance Using the GeneNetWeaver Tool

In this appendix, we compare the performance of CCIT with three different algorithms, using the GeneNetWeaver (GNW) tool [196]. In Chapter 5, we presented the performance comparison using the F-score. In this appendix, we also present the results corresponding to the other performance measures. We make observations similar to Chapter 5, that there is no clear winner in all the cases. However, our proposed approach is among the top performers and it shows consistent performance.

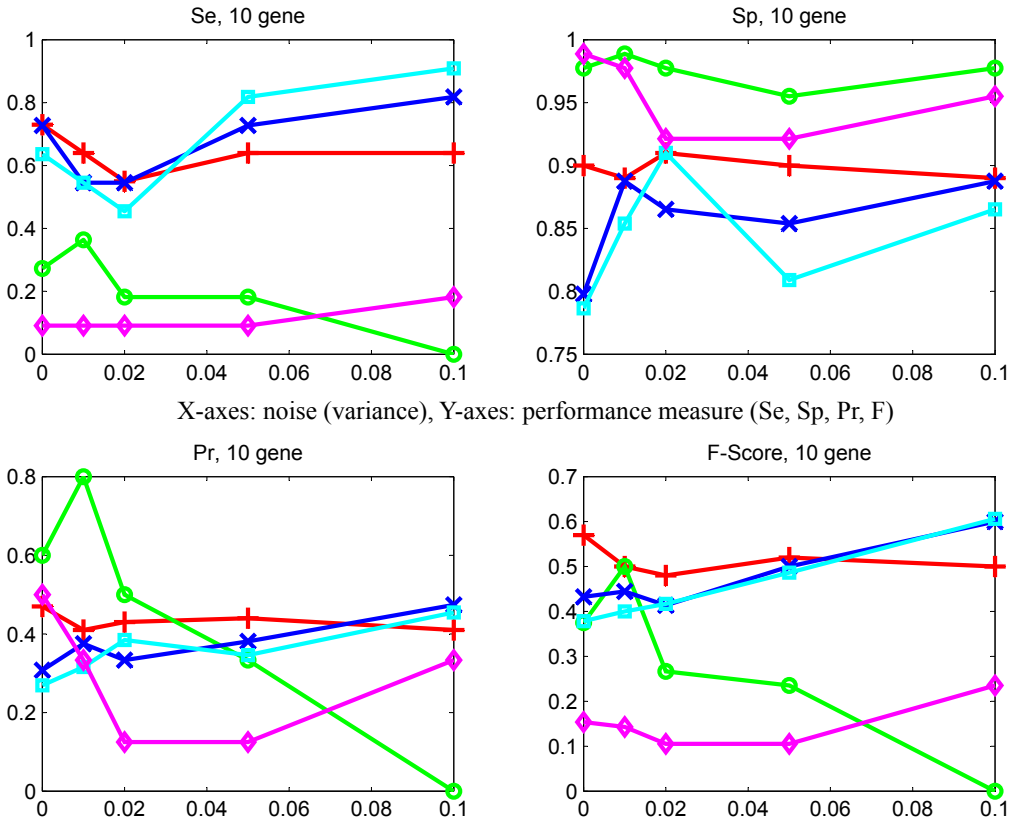

Figure B.1: Comparison of performance with 3 other methods for the 10-gene synthetic network generated using GeneNetWeaver [196]. Red(+) - CCIT, Green(o) - Banjo, Blue(x) - BNFinder+BDe, Cyan(square) - BNFinder+MDL, Magenta(diamond) - TDARACNE.

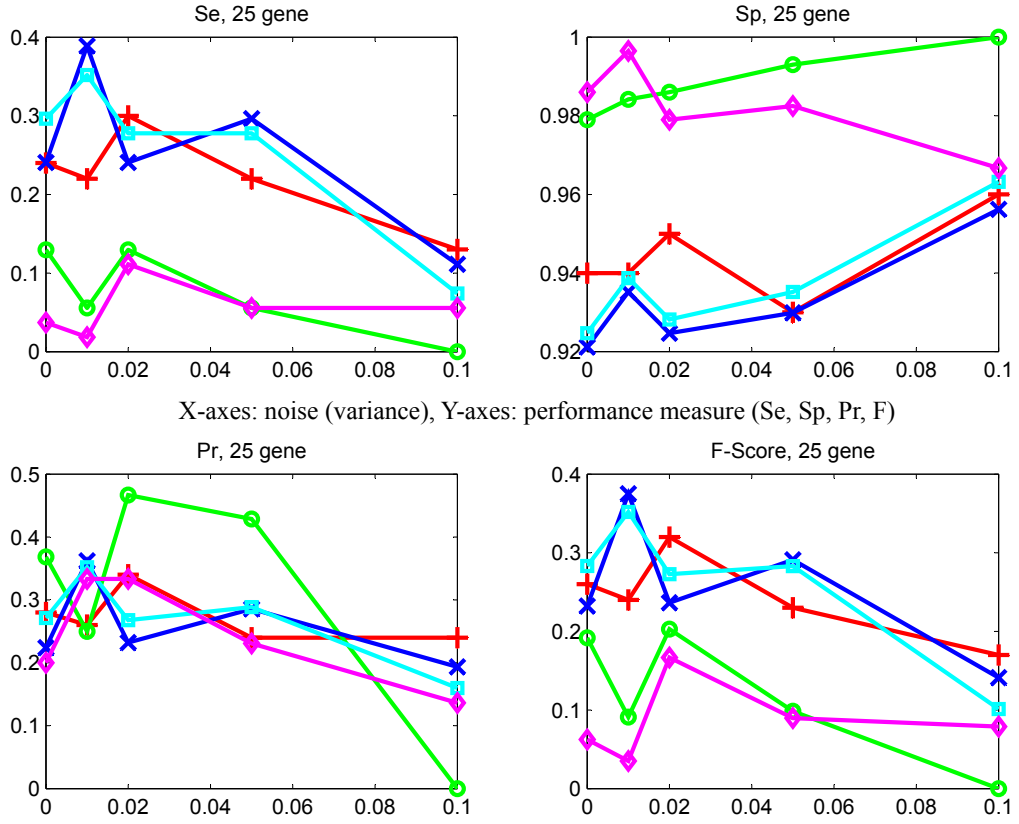

Figure B.2: Comparison of performance with 3 other methods for the 25-gene synthetic network generated using GeneNetWeaver [196]. Red(+) - CCIT, Green(o) - Banjo, Blue(x) - BNFinder+BDe, Cyan(square) - BNFinder+MDL, Magenta(diamond) - TDARACNE.

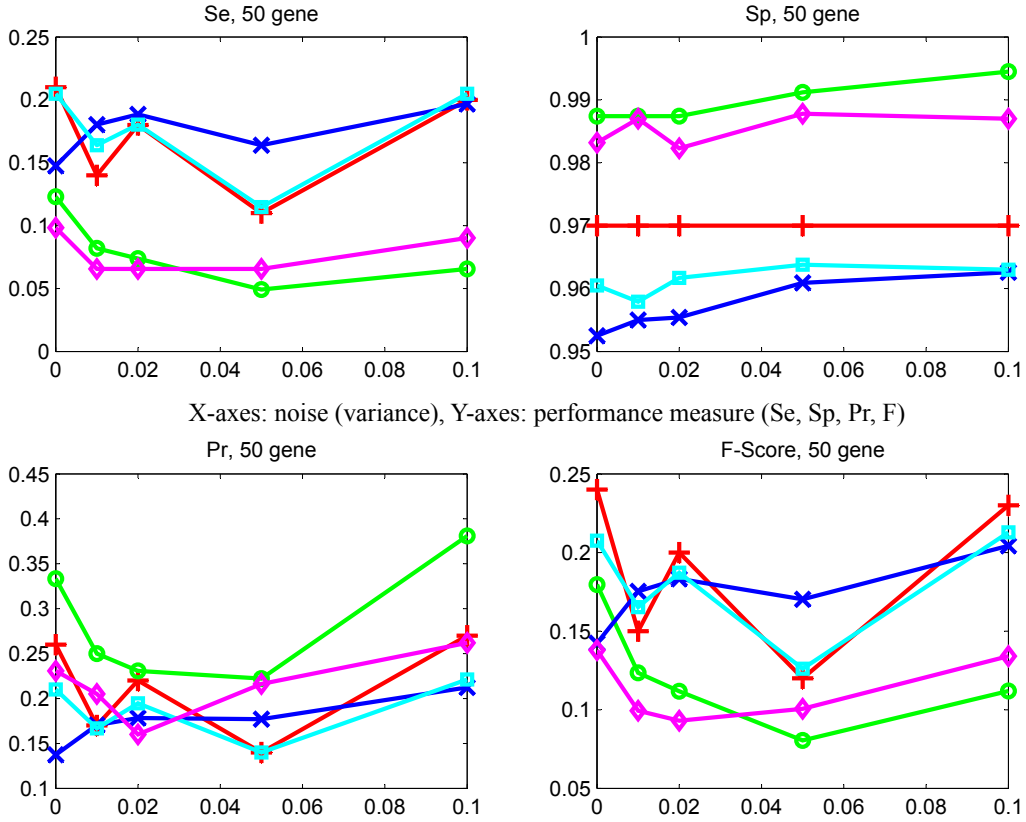

Figure B.3: Comparison of performance with 3 other methods for the 50-gene synthetic network network generated using GeneNetWeaver [196]. Red(+) - CCIT, Green(o) - Banjo, Blue(x) - BNFinder+BDe, Cyan(square) - BNFinder+MDL, Magenta(diamond) - TDARACNE.

Appendix C

Parametric Settings in Methods Used for Comparison

In this thesis, we have done the comparison of our approach with three related methods: BANJO, BNFinder, and TDARACNE. Unless otherwise stated, the parameter settings for these three methods are given in the following tables:

BANJO Primary Settings	
discretizationPolicy	q3
discretizationExceptions	NONE
searcherChoice	Simulated Annealing
proposerChoice	AllLocalMoves
minMarkovLag	1
maxMarkovLag	1
equivalentSampleSize	1
maxParentCount	3
maxTime	10 minute

For simulated annealing, the following parameters were used:

BANJO Settings for simulated annealing	
initialTemperature	1000
coolingFactor	0.8
maxAcceptedNetworksBeforeCooling	1000
maxProposedNetworksBeforeCooling	10000
minAcceptedNetworksBeforeReannealing	200
reannealingTemperature	500

BNFinder settings are as follows:

BNFinder Settings	
Scoring Function	BDe / MDL
Number of parents limit	3
Number of suboptimal parent sets to consider	1
prior pseudocount factor	1

Finally, we used the following settings for TDARACNE:

TDARACNE Settings	
N (number of bins in normalization)	15
Delta (maximum time delay)	3
Likehood (fold change to be used for IcE)	1.2
Norm	1 (percentile normalization)
Logarithm	0
Threshold	0
ksd (standard deviation multiplier)	1
DPI tolerance	0

Glossary

Abbreviation	Meaning
2D	Two-dimensional
AIC	Akaike Information Criterion
BD	Bayesian Dirichlet (score)
BIC	Bayesian Information Criterion
BN	Bayesian Network
CD	Conditional Dependence
CI	Conditional Independence
CLM	Constraint Logic Minimization
CMI	Conditional Mutual Information
CPD	Conditional Probability Distribution
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DE	Differential Equation
DFS	Depth First Search
DNA	Deoxyribonucleic acid
DPI	Data Processing Inequality
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GEO	Geometric Graph
GGM	Graphical Gaussian Model
GNW	GeneNetWeaver Tool
GO	Gene Ontology
HMM	Hidden Markov Model

Continued on next page...

Glossary – Continued

Abbreviation	Meaning
KL	Kullback-Leibler (divergence)
LL	Log-Likelihood
MAP	Maximum a Posteriori
MB	Markov Blanket
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length
MI	Mutual Information
MIT	Mutual Information Tests (score)
ML	Maximum Likelihood
mRNA	Messenger RNA
PCA	Principal Component Analysis
PCC	Partial Correlation Coefficient
PI	Probabilistic Interactomes (for PPIN)
PIE	PI-Experimental
PIP	PI-Predicted
PIT	Total PI
PPI	Protein-Protein Interaction
PPIN	Protein-Protein Interaction Network
PPMC	Pearson Product Moment Correlation
Pr	Precision
RNA	Ribonucleic acid
Se	Sensitivity
Sp	Specificity
SVD	Singular Value Decomposition
TAP	Tandem Affinity Purification Experiment
TDE	Trigonometric Differential Evolution
tRNA	Transfer RNA
UAS	Upstream Activating Sequence
Y2H	Yeast Two-hybrid assay

References

[1] S. ACID AND L.M. DE CAMPOS. **Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs**. *Journal of Artificial Intelligence Research*, **18**(445-490):68, 2003.

[2] H. AKAIKE. **A new look at the statistical model identification**. *Automatic Control, IEEE Transactions on*, **19**(6):716–723, 1974.

[3] T. AKUTSU, S. KUHARA, O. MARUYAMA, AND S. MIYANO. **Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions**. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 695–702. Society for Industrial and Applied Mathematics, 1998.

[4] T. AKUTSU, S. MIYANO, AND S. KUHARA. **Algorithms for inferring qualitative models of biological networks**. In *Pacific Symposium on Biocomputing*, **5**, pages 290–301, 2000.

[5] TATSUYA AKUTSU, SATORU MIYANO, SATORU KUHARA, ET AL. **Identification of genetic networks from a small number of gene expression patterns under the Boolean network model**. In *Pacific Symposium on Biocomputing*, **4**, pages 17–28. World Scientific Maui, Hawaii, 1999.

[6] B. ALBERTS, A. JOHNSON, J. LEWIS, M. RAFF, K. ROBERTS, P. WALTER, ET AL. **Molecular Biology of the Cell**. 4th edition, 2002.

[7] CF ALIFERIS AND I. TSAMARDINOS. **Algorithms for Large-scale Local Causal Discovery and Feature Selection in the Presence of Limited Sample or Large Causal Neighbourhoods**. Technical report, Technical report DSL-02-08, Department of Biomedical Informatics, Vanderbilt University, 2002.

[8] U. ALON. **Uri Alon's SOS Dataset webpage**. Available from: http://www.weizmann.ac.il/mcb/UriAlon/Papers/SOSData/. Last Accessed: 10-12-2012.

[9] U. ALON, N. BARKAI, D.A. NOTTERMAN, K. GISH, S. YBARRA, D. MACK, AND A.J. LEVINE. **Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays**. *Proceedings of the National Academy of Sciences*, **96**(12):6745–6750, 1999.

[10] G. AMOUTZIAS AND Y. VAN DE PEER. **Single-Gene and Whole-Genome Duplications and the Evolution of Protein–Protein Interaction Networks**. *Evolutionary Genomics and Systems Biology*, pages 413–429, 2010.

[11] IP ANDROULAKIS, E YANG, AND RR ALMON. **Analysis of time-series gene expression data: Methods, challenges, and opportunities**. *Annu. Rev. Biomed. Eng.*, **9**:205–228, 2007.

[12] B. ARANDA, P. ACHUTHAN, Y. ALAM-FARUQUE, I. ARMEAN, A. BRIDGE, C. DEROW, M. FEUERMANN, AT GHANBARIAN, S. KERRIEN, J. KHADAKE, ET AL. **The IntAct molecular interaction database in 2010**. *Nucleic acids research*, **38**(suppl 1):D525–D531, 2010.

[13] ADAM ARKIN, PEIDONG SHEN, AND JOHN ROSS. **A test case of correlation metric construction of a reaction pathway from measurements**. *Science*, **277**(5330):1275–1279, 1997.

[14] MICHAEL ASHBURNER, CATHERINE A BALL, JUDITH A BLAKE, DAVID BOTSTEIN, HEATHER BUTLER, J MICHAEL CHERRY, ALLAN P DAVIS, KARA DOLINSKI, SELINA S DWIGHT, JANAN T EPPIG, ET AL. **Gene Ontology: tool for the unification of biology**. *Nature genetics*, **25**(1):25–29, 2000.

[15] DAVIDE BACCIU, TERENCE A ETCHELLS, PAULO JG LISBOA, AND JOE WHITTAKER. **Efficient identification of independence networks using mutual information**. *Computational Statistics*, pages 1–26, 2012.

[16] M. BANSAL, V. BELCASTRO, A. AMBESI-IMPIOMBATO, AND D. DI BERNARDO. **How to infer gene networks from expression profiles**. *Molecular systems biology*, **3**(1), 2007.

[17] M. BANSAL, G. DELLA GATTA, AND D. DI BERNARDO. **Inference of gene regulatory networks and compound mode of action from time course gene expression profiles**. *Bioinformatics*, **22**(7):815–822, 2006.

[18] ZIV BAR-JOSEPH. **Analyzing time series gene expression data**. *Bioinformatics*, **20**(16):2493–2503, 2004.

[19] SD BAY, L. CHRISMAN, A. POHORILLE, AND J. SHRAGER. **Temporal aggregation bias and inference of causal regulatory networks**. *Journal of Computational Biology*, **11**(5):971–985, 2004.

[20] ATTILA BECSKEI, BERTRAND SÉRAPHIN, AND LUIS SERRANO. **Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion**. *The EMBO journal*, **20**(10):2528–2535, 2001.

[21] G. BEJERANO, N. FRIEDMAN, AND N. TISHBY. **Efficient exact p-value computation for small sample, sparse, and surprising categorical data**. *Journal of Computational Biology*, **11**(5):867–886, 2004.

[22] A. BEN-DOR, R. SHAMIR, AND Z. YAKHINI. **Clustering gene expression patterns**. *Journal of computational biology*, **6**(3-4):281–297, 1999.

[23] TANYA Z BERARDINI, VARSHA K KHODIYAR, RUTH C LOVERING, AND PHILIPPA TALMUD. **The Gene Ontology in 2010: extensions and refinements**. *Nucleic acids research*, **38**(Database Issue):D331–D335, 2010.

[24] A. BERNARD, A.J. HARTEMINK, ET AL. **Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data**. In *Pac Symp Biocomput*, **10**, pages 459–470, 2005.

[25] H. BOLOURI AND E.H. DAVIDSON. **Modeling transcriptional regulatory networks**. *BioEssays*, **24**(12):1118–1129, 2002.

[26] P. BORK, L.J. JENSEN, C. VON MERING, A.K. RAMANI, I. LEE, AND E.M. MARCOTTE. **Protein interaction networks: from yeast to human**. *Current opinion in structural biology*, **14**(3):292–299, 2004.

[27] R. BOUCKAERT. **Probabilistic network construction using the minimum description length principle**. *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 41–48, 1993.

[28] R.R. BOUCKAERT. *Bayesian belief networks: from construction to inference*. Universiteit Utrecht, Faculteit Wiskunde en Informatica, 1995.

[29] P. BRAZHNIK, A. DE LA FUENTE, AND P. MENDES. **Gene networks: how to put the function in genomics**. *Trends in Biotechnology*, **20**(11):467–472, 2002.

[30] KC BRENNAN, EMILY A BATES, ROBERT E SHAPIRO, JEKATERINA ZYUZIN, WILLIAM C HALLOWS, YONG HUANG, HSIEN-YANG LEE, CHRISTOPHER R JONES, YING-HUI FU, ANDREW C CHARLES, ET AL. **Casein Kinase Iδ Mutations in Familial Migraine and Advanced Sleep Phase**. *Science translational medicine*, **5**(183):183ra56–183ra56, 2013.

[31] S. BULASHEVSKA, A. BULASHEVSKA, AND R. EILS. **Bayesian statistical modelling of human protein interaction network incorporating protein disorder information**. *BMC bioinformatics*, **11**(1):46, 2010.

[32] W. BUNTINE. **Theory refinement on Bayesian networks**. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

[33] A.J. BUTTE AND I.S. KOHANE. **Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements**. In *Pac Symp Biocomput*, **5**, pages 418–429, 2000.

[34] A.J. BUTTE, P. TAMAYO, D. SLONIM, T.R. GOLUB, AND I.S. KOHANE. **Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks**. *Proceedings of the National Academy of Sciences*, **97**(22):12182–12186, 2000.

[35] I. CANTONE, L. MARUCCI, F. IORIO, M.A. RICCI, V. BELCASTRO, M. BANSAL, S. SANTINI, M. DI BERNARDO, D. DI BERNARDO, AND M.P. COSMA. **A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches**. *Cell*, **137**(1):172–181, 2009.

[36] A. CEOL, A.C. ARYAMONTRI, L. LICATA, D. PELUSO, L. BRIGANTI, L. PERFETTO, L. CASTAGNOLI, AND G. CESARENI. **MINT, the molecular interaction database: 2009 update**. *Nucleic acids research*, **38**(suppl 1):D532–D539, 2010.

[37] V. CHAITANKAR, P. GHOSH, E. PERKINS, P. GONG, Y. DENG, AND C. ZHANG. **A novel gene network inference algorithm using predictive minimum description length approach**. *BMC Systems Biology*, **4**(Suppl 1):S7, 2010.

[38] S.S. CHANTHAPHAVONG AND M. CHETTY. **Binary-Organoid Particle Swarm optimisation for inferring genetic networks**. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–10. IEEE, 2010.

[39] I. CHATURVEDI AND J. RAJAPAKSE. **Fusion of Gene Regulatory and Protein Interaction Networks Using Skip-Chain Models**. *Pattern Recognition in Bioinformatics*, pages 214–224, 2008.

[40] J.Y. CHEN, S.R. MAMIDIPALLI, AND T. HUAN. **HAPPI: an online database of comprehensive human annotated and predicted protein interactions**. *BMC genomics*, **10**(Suppl 1):S16, 2009.

[41] T. CHEN, H.L. HE, G.M. CHURCH, ET AL. **Modeling gene expression with differential equations**. In *Pacific symposium on biocomputing*, **4**, page 4, 1999.

[42] XUE-WEN CHEN, GOPALAKRISHNA ANANTHA, AND XIAOTONG LIN. **Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm**. *Knowledge and Data Engineering, IEEE Transactions on*, **20**(5):628–640, 2008.

[43] J. CHENG, D.A. BELL, AND W. LIU. **An algorithm for Bayesian belief network construction from data**. In *proceedings of AI & STAT97*, pages 83–90, 1997.

[44] Y. CHENG AND G.M. CHURCH. **Biclustering of expression data**. In *Proceedings of the eighth international conference on intelligent systems for molecular biology*, **8**, pages 93–103, 2000.

[45] D.M. CHICKERING. **Learning Bayesian networks is NP-complete**. *Lecture Notes in Statistics-New York, Springer Verlag*, pages 121–130, 1996.

[46] D.M. CHICKERING, D. HECKERMAN, AND C. MEEK. **Large-sample learning of Bayesian networks is NP-hard**. *The Journal of Machine Learning Research*, **5**:1287–1330, 2004.

[47] D.M. CHICKERING AND C. MEEK. **Finding optimal Bayesian networks**. In *Proc. UAI*, 2002.

[48] SHIN C CHIN, ASOK RAY, AND VENKATESH RAJAGOPALAN. **Symbolic time series analysis for anomaly detection: a comparative evaluation**. *Signal Processing*, **85**(9):1859–1868, 2005.

[49] R.J. CHO, M.J. CAMPBELL, E.A. WINZELER, L. STEINMETZ, A. CONWAY, L. WODICKA, T.G. WOLFSBERG, A.E. GABRIELIAN, D. LANDSMAN, D.J. LOCKHART, ET AL. **A genome-wide transcriptional analysis of the mitotic cell cycle**. *Molecular cell*, **2**(1):65–73, 1998.

[50] A.R. CHOWDHURY AND M. CHETTY. **An improved method to infer gene regulatory network using s-system**. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1012–1019. IEEE, 2011.

[51] T. CHU, C. GLYMOUR, R. SCHEINES, AND P. SPIRTES. **A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays**. *Bioinformatics*, **19**(9):1147–1152, 2003.

[52] K CHUN AND M GOEBL. **Mutational analysis of Cak1p, an essential protein kinase that regulates cell cycle progression**. *Molecular and General Genetics MGG*, **256**(4):365–375, 1997.

[53] JENNIFER I CLARK, CATH BROOKSBANK, AND JANE LOMAX. **It's all GO for plant scientists**. *Plant physiology*, **138**(3):1268–1279, 2005.

[54] G.F. COOPER AND E. HERSKOVITS. **A Bayesian method for the induction of probabilistic networks from data**. *Machine learning*, **9**(4):309–347, 1992.

[55] T.M. COVER, J.A. THOMAS, J. WILEY, ET AL. *Elements of information theory*, **306**. Wiley Online Library, 1991.

[56] FR CROSS. **Starting the cell cycle: what's the point?** *Current Opinion in Cell Biology*, **7**(6):790–797, 1995.

[57] R. DALY AND Q. SHEN. **Learning bayesian network equivalence classes with ant colony optimization**. *Journal of Artificial Intelligence Research*, **35**(1):391, 2009.

[58] RÓNÁN DALY, KIERON EDWARDS, JOHN ONEILL, STUART AITKEN, ANDREW MILLAR, AND MARK GIROLAMI. **Using Higher-Order Dynamic Bayesian Networks to Model Periodic Data from the Circadian Clock of Arabidopsis Thaliana**. *Pattern Recognition in Bioinformatics*, pages 67–78, 2009.

[59] C.P. DE CAMPOS AND Q. JI. **Efficient Structure Learning of Bayesian Networks using Constraints**. *Journal of Machine Learning Research*, **12**:663–689, 2011.

[60] L.M. DE CAMPOS. **A scoring function for learning Bayesian networks based on mutual information and conditional independence tests**. *The Journal of Machine Learning Research*, **7**:2149–2187, 2006.

[61] L.M. DE CAMPOS, J.M. FERNÁNDEZ-LUNA, AND J.M. PUERTA. **An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests**. *International Journal of Intelligent Systems*, **18**(2):221–235, 2003.

[62] L.M. DE CAMPOS, JM PUERTA, ET AL. **Learning bayesian networks by ant colony optimisation: searching in two different spaces**. *Mathware & soft computing*, **9**(3):251–268, 2008.

[63] H. DE JONG. **Modeling and simulation of genetic regulatory systems: a literature review**. *Journal of computational biology*, **9**(1):67–103, 2002.

[64] P. D'HAESELEER. *Reconstructing gene networks from large scale gene expression data*. PhD thesis, The University of New Mexico, 2000.

[65] D. DI BERNARDO, M.J. THOMPSON, T.S. GARDNER, S.E. CHOBOT, E.L. EASTWOOD, A.P. WOJTOVICH, S.J. ELLIOTT, S.E. SCHAUS, AND J.J. COLLINS. **Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks**. *Nature biotechnology*, **23**(3):377–383, 2005.

[66] N. DOJER. **Learning Bayesian networks does not have to be NP-hard**. *Mathematical Foundations of Computer Science 2006*, pages 305–314, 2006.

[67] A. DRAWID AND M. GERSTEIN. **A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome**. *Journal of molecular biology*, **301**(4):1059–1075, 2000.

[68] P. DHAESELEER, S. LIANG, AND R. SOMOGYI. **Genetic network inference: from co-expression clustering to reverse engineering**. *Bioinformatics*, **16**(8):707–726, 2000.

[69] PATRIK DHAESELEER, SHOUDAN LIANG, AND ROLAND SOMOGYI. **Gene expression data analysis and modeling**. In *Pacific Symposium on Biocomputing*, **99**, 1999.

[70] D. EATON AND K. MURPHY. **Bayesian structure learning using dynamic programming and MCMC**. *Proceedings of the Twenty-Third Confererence on Uncertainty in Artificial Intelligence (UAI 2007)*, 2007.

[71] M.B. EISEN, P.T. SPELLMAN, P.O. BROWN, AND D. BOTSTEIN. **Cluster analysis and display of genome-wide expression patterns**. *Proceedings of the National Academy of Sciences*, **95**(25):14863–14868, 1998.

[72] NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION (NCBI). **Illustration of a cell**. Available from: http://www.ncbi.nlm.nih.gov/About/primer/genetics_cell.html. Last Accessed: 08-05-2013.

[73] EUROPEAN SCIENCE FOUNDATION. **Functional Genomics: Protein Arrays Resource Page**. Available from: http://www.functionalgenomics.org.uk/sections/resources/protein_arrays.htm. Last Accessed: 23-05-2013.

[74] THE ANNENBERG FOUNDATION. **Rediscovering Biology: Protein Microarrays**. Available from: http://www.learner.org/courses/biology/textbook/proteo/proteo_11.html. Last Accessed: 23-05-2013.

[75] N. FRIEDMAN. **Inferring cellular networks using probabilistic graphical models**. *Science Signalling*, **303**(5659):799, 2004.

[76] N. FRIEDMAN, M. LINIAL, I. NACHMAN, AND D. PE'ER. **Using Bayesian networks to analyze expression data**. *Journal of computational biology*, **7**(3-4):601–620, 2000.

[77] N. FRIEDMAN, K. MURPHY, AND S. RUSSELL. **Learning the structure of dynamic probabilistic networks**. In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI98)*, pages 139–147. Citeseer, 1998.

[78] T.S. GARDNER, C.R. CANTOR, AND J.J. COLLINS. **Construction of a genetic toggle switch in Escherichia coli**. *Nature*, **403**:339–342, 2000.

[79] T.S. GARDNER AND J.J. FAITH. **Reverse-engineering transcription control networks**. *Physics of life reviews*, **2**(1):65–88, 2005.

[80] A.P. GASCH, M.B. EISEN, ET AL. **Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering**. *Genome Biol*, **3**(11):1–22, 2002.

[81] AUDREY P GASCH, PAUL T SPELLMAN, CAMILLA M KAO, ORNA CARMEL-HAREL, MICHAEL B EISEN, GISELA STORZ, DAVID BOTSTEIN, AND PATRICK O BROWN. **Genomic expression programs in the response of yeast cells to environmental changes**. *Science Signalling*, **11**(12):4241, 2000.

[82] A.C. GAVIN, M. BÖSCHE, R. KRAUSE, P. GRANDI, M. MARZIOCH, A. BAUER, J. SCHULTZ, J.M. RICK, A.M. MICHON, C.M. CRUCIAT, ET AL. **Functional organization of the yeast proteome by systematic analysis of protein complexes**. *Nature*, **415**(6868):141–147, 2002.

[83] C.E. GIACOMANTONIO AND G.J. GOODHILL. **A Boolean model of the gene regulatory network underlying mammalian cortical area development**. *PLoS computational biology*, **6**(9):e1000936, 2010.

[84] DAVID E GOLDBERG. **Genetic algorithms in search, optimization, and machine learning**. *Addison-Wesley Professional*, 1989.

[85] D.E. GOLDBERG AND S. VOESSNER. **Optimizing global-local search hybrids**. *Urbana*, **51**:61801, 1999.

[86] S. GÖTZ, J.M. GARCÍA-GÓMEZ, J. TEROL, T.D. WILLIAMS, S.H. NAGARAJ, M.J. NUEDA, M. ROBLES, M. TALÓN, J. DOPAZO, AND A. CONESA. **High-throughput functional annotation and data mining with the Blast2GO suite**. *Nucleic Acids Research*, **36**(10):3420–3435, 2008.

[87] M. GRZEGORCZYK AND D. HUSMEIER. **Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move**. *Machine Learning*, **71**(2):265–305, 2008.

[88] N. GUELZIM, S. BOTTANI, P. BOURGINE, AND F. KÉPÈS. **Topological and causal structure of the yeast transcriptional regulatory network**. *Nature genetics*, **31**(1):60–63, 2002.

[89] D.A. HALL, J. PTACEK, AND M. SNYDER. **Protein microarray technology**. *Mechanisms of ageing and development*, **128**(1):161–167, 2007.

[90] E. HARTUV AND R. SHAMIR. **A clustering algorithm based on graph connectivity**. *Information processing letters*, **76**(4):175–181, 2000.

[91] J. HAUSSER AND K. STRIMMER. **Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks**. *The Journal of Machine Learning Research*, **10**:1469–1484, 2009.

[92] D. HECKERMAN. **A tutorial on learning with Bayesian networks**. *Innovations in Bayesian Networks*, pages 33–82, 2008.

[93] D. HECKERMAN, D. GEIGER, AND D.M. CHICKERING. **Learning Bayesian networks: The combination of knowledge and statistical data**. *Machine learning*, **20**(3):197–243, 1995.

[94] Y. HO, A. GRUHLER, A. HEILBUT, G.D. BADER, L. MOORE, S.L. ADAMS, A. MILLAR, P. TAYLOR, K. BENNETT, K. BOUTILIER, ET AL. **Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry**. *Nature*, **415**(6868):180–183, 2002.

[95] C. HOCAOGLU AND A.C. SANDERSON. **Multimodal function optimization using minimal representation size clustering and its application to planning multipaths**. *Evolutionary Computation*, **5**(1):81–104, 1997.

[96] H. HU, X. YAN, Y. HUANG, J. HAN, AND X.J. ZHOU. **Mining coherent dense subgraphs across massive biological networks for functional discovery**. *Bioinformatics*, **21**(suppl 1):i213–i221, 2005.

[97] T.R. HUGHES, M.J. MARTON, A.R. JONES, C.J. ROBERTS, R. STOUGHTON, C.D. ARMOUR, H.A. BENNETT, E. COFFEY, H. DAI, Y.D. HE, ET AL. **Functional discovery via a compendium of expression profiles**. *Cell*, **102**(1):109–126, 2000.

[98] D. HUSMEIER. **Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks**. *Bioinformatics*, **19**(17):2271, 2003.

[99] M. ISLAM AND M. CHETTY. **Clustered Memetic Algorithm with Local Heuristics for ab initio Protein Structure Prediction**. *IEEE Transactions on Evolutionary Computation*, **PP, Issue: 99**, 2012.

[100] R. ISSERLIN, R.A. EL-BADRAWI, AND G.D. BADER. **The biomolecular interaction network database in PSI-MI 2.5**. *Database: the journal of biological databases and curation*, **2011**, 2011.

[101] T. ITO, T. CHIBA, R. OZAWA, M. YOSHIDA, M. HATTORI, AND Y. SAKAKI. **A comprehensive two-hybrid analysis to explore the yeast protein interactome**. *Proceedings of the National Academy of Sciences*, **98**(8):4569–4574, 2001.

[102] NATALIA B IVANOVA, JOHN T DIMOS, CHRISTOPH SCHANIEL, JASON A HACKNEY, KATERI A MOORE, AND IHOR R LEMISCHKA. **A stem cell molecular signature**. *Science*, **298**(5593):601–604, 2002.

[103] A. JAIMOVICH, G. ELIDAN, H. MARGALIT, AND N. FRIEDMAN. **Towards an integrated protein-protein interaction network: a relational markov network approach**. *Journal of Computational Biology*, **13**(2):145–164, 2006.

[104] R. JANSEN, H. YU, D. GREENBAUM, Y. KLUGER, N.J. KROGAN, S. CHUNG, A. EMILI, M. SNYDER, J.F. GREENBLATT, AND M. GERSTEIN. **A Bayesian networks approach for predicting protein-protein interactions from genomic data**. *Science*, **302**(5644):449–453, 2003.

[105] H. JEONG, S.P. MASON, A.L. BARABASI, AND Z.N. OLTVAI. **Lethality and centrality in protein networks**. *Arxiv preprint cond-mat/0105306*, 2001.

[106] H. JEONG, B. TOMBOR, R. ALBERT, Z.N. OLTVAI, AND A.L. BARABÁSI. **The large-scale organization of metabolic networks**. *Nature*, **407**(6804):651–654, 2000.

[107] J.K. JOUNG, E.I. RAMM, AND C.O. PABO. **A bacterial two-hybrid selection system for studying protein–DNA and protein–protein interactions**. *Proceedings of the National Academy of Sciences*, **97**(13):7382–7387, 2000.

[108] M. KANEHISA, S. GOTO, S. KAWASHIMA, AND A. NAKAYA. **The KEGG databases at GenomeNet**. *Nucleic acids research*, **30**(1):42–46, 2002.

[109] S.A. KAUFFMAN. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[110] STUART A KAUFFMAN. **Metabolic stability and epigenesis in randomly constructed genetic nets**. *Journal of theoretical biology*, **22**(3):437–467, 1969.

[111] KAZUSA DNA RESEARCH INSTITUTE (KDRI). **The cyanobacteria database**. Available from: http://genome.kazusa.or.jp/cyanobase. Last Accessed: 23-05-2013.

[112] EDWARD KEEDWELL AND AJIT NARAYANAN. **Discovering gene networks with a neural-genetic hybrid**. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **2**(3):231–242, 2005.

[113] H. KIM, J.K. LEE, AND T. PARK. **Boolean networks using the chi-square test for inferring large-scale gene regulatory networks**. *BMC bioinformatics*, **8**(1):37, 2007.

[114] S. KIM, S. IMOTO, AND S. MIYANO. **Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data**. *Biosystems*, **75**(1-3):57–65, 2004.

[115] S. KIMURA, Y. AMANO, K. MATSUMURA, AND M. OKADA-HATAKEYAMA. **Effective parameter estimation for S-system models using LPMs and evolutionary algorithms**. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[116] S. KIMURA, K. IDE, A. KASHIHARA, M. KANO, M. HATAKEYAMA, R. MASUI, N. NAKAGAWA, S. YOKOYAMA, S. KURAMITSU, AND A. KONAGAYA. **Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm**. *Bioinformatics*, **21**(7):1154–1163, 2005.

[117] HIROHISA KISHINO, PETER J WADDELL, ET AL. **Correspondence analysis of genes and tissue types and finding genetic links from microarray data**. *Genome Informatics Series*, pages 83–95, 2000.

[118] HIROAKI KITANO. **Systems biology: a brief overview**. *Science*, **295**(5560):1662–1664, 2002.

[119] T. KOČKA AND R. CASTELO. **Improved learning of Bayesian networks**. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 269–276. Morgan Kaufmann Publishers Inc., 2001.

[120] MEHMET KOYUTÜRK, ANANTH GRAMA, AND WOJCIECH SZPANKOWSKI. **An efficient algorithm for detecting frequent subgraphs in biological networks**. *Bioinformatics*, **20**(suppl 1):i200–i207, 2004.

[121] LANEY KUENZEL. **Gene clustering methods for time series microarray data**. 2010.

[122] DON KULASIRI, LAN K NGUYEN, SANDHYA SAMARASINGHE, AND ZHI XIE. **A review of systems biology perspective on genetic regulatory networks with examples**. *Current Bioinformatics*, **3**(3):197–225, 2008.

[123] S. KULLBACK. *Information theory and statistics*. Dover Publications, 1968.

[124] S. KULLBACK AND R.A. LEIBLER. **On information and sufficiency**. *The Annals of Mathematical Statistics*, **22**(1):79–86, 1951.

[125] MICHIHIRO KURAMOCHI AND GEORGE KARYPIS. **Finding frequent patterns in a large sparse graph***. *Data mining and knowledge discovery*, **11**(3):243–271, 2005.

[126] W. LAM AND F. BACCHUS. **Learning Bayesian belief networks: An approach based on the MDL principle**. *Computational intelligence*, **10**(3):269–293, 1994.

[127] P. LARRAÑAGA, M. POZA, Y. YURRAMENDI, R.H. MURGA, AND C.M.H. KUIJPERS. **Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **18**(9):912–926, 1996.

[128] P.P. LE, A. BAHL, AND L.H. UNGAR. **Using prior knowledge to improve genetic network reconstruction from microarray data**. *silico biology*, **4**(3):335–353, 2004.

[129] I. LEE, S.V. DATE, A.T. ADAI, AND E.M. MARCOTTE. **A probabilistic functional network of yeast genes**. *science*, **306**(5701):1555–1558, 2004.

[130] THE NEUROSCIENCE LEXICON. **Gene Ontology Analysis Resources**. Available from: http://neurolex.org/wiki/Category:Resource:Gene_Ontology_Tools. Last Accessed: 23-05-2013.

[131] X. LI AND G. OUYANG. **Estimating coupling direction between neuronal populations with permutation conditional mutual information**. *NeuroImage*, **52**(2):497–507, 2010.

[132] S. LIANG, S. FUHRMAN, R. SOMOGYI, ET AL. **REVEAL, a general reverse engineering algorithm for inference of genetic network architectures**. In *Pacific symposium on biocomputing*, **3**, page 2, 1998.

[133] HARVEY LODISH, ARNOLD BERK, S LAWRENCE ZIPURSKY, PAUL MATSUDAIRA, DAVID BALTIMORE, AND JAMES DARNELL. **Molecular cell biology**. *New York*, 2000.

[134] D. MADIGAN, S.A. ANDERSSON, M.D. PERLMAN, AND C.T. VOLINSKY. **Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs**. *Communications in Statistics–Theory and Methods*, **25**(11):2493–2519, 1996.

[135] S. MAERE, K. HEYMANS, AND M. KUIPER. **BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks**. *Bioinformatics*, **21**(16):3448–3449, 2005.

[136] PAUL M MAGWENE, JUNHYONG KIM, ET AL. **Estimating genomic coexpression networks using first-order conditional independence**. *Genome Biol*, **5**(12):R100, 2004.

[137] I.A. MARAZIOTIS, A. DRAGOMIR, AND D. THANOS. **Gene regulatory networks modelling using a dynamic evolutionary hybrid**. *BMC bioinformatics*, **11**(1):140, 2010.

[138] D. MARBACH, T. SCHAFFTER, C. MATTIUSSI, AND D. FLOREANO. **Generating realistic in silico gene networks for performance assessment of reverse engineering methods**. *Journal of Computational Biology*, **16**(2):229–239, 2009.

[139] A. MARGOLIN, I. NEMENMAN, K. BASSO, C. WIGGINS, G. STOLOVITZKY, R. FAVERA, AND A. CALIFANO. **ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context**. *BMC bioinformatics*, **7**(Suppl 1):S7, 2006.

[140] A.C. MARTÍNEZ-ESTUDILLO, C. HERVÁS-MARTÍNEZ, F.J. MARTÍNEZ-ESTUDILLO, AND N. GARCÍA-PEDRAJAS. **Hybridization of evolutionary algorithms and local search by means of a clustering method**. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **36**(3):534–545, 2005.

[141] ANA MARGARIDA MARTINS, PEDRO MENDES, CARLOS CORDEIRO, AND ANA PONCES FREIRE. **In situ kinetic analysis of glyoxalase I and glyoxalase II in Saccharomyces cerevisiae**. *European Journal of Biochemistry*, **268**(14):3930–3936, 2001.

[142] J.E. MCDERMOTT, C.S. OEHMEN, L.A. MCCUE, E. HILL, D.M. CHOI, J. STÖCKEL, M. LIBERTON, H.B. PAKRASI, AND L.A. SHERMAN. **A model of cyclic transcriptomic behavior in the cyanobacterium Cyanothece sp. ATCC 51142**. *Mol. BioSyst.*, **7**(8):2407–2418, 2011.

[143] NOBEL MEDIA. **Chemical Structure of DNA**. Available from: http://www.nobelprize.org/educational/medicine/dna/b/replication/dna_structure.html. Last Accessed: 23-05-2013.

[144] NOBEL MEDIA. **Illustration of DNA**. Available from: http://www.nobelprize.org/educational/medicine/dna/b/replication/dna_base.html. Last Accessed: 13-05-2013.

[145] L. MELTON. **Protein arrays: proteomics in multiplex**. *Nature*, **429**(6987):101–107, 2004.

[146] H.W. MEWES, D. FRISHMAN, U. GÜLDENER, G. MANNHAUPT, K. MAYER, M. MOKREJS, B. MORGENSTERN, M. MÜNSTERKÖTTER, S. RUDD, AND B. WEIL. **MIPS: a database for genomes and protein sequences**. *Nucleic acids research*, **30**(1):31–34, 2002.

[147] G.S. MICHAELS, D.B. CARR, M. ASKENAZI, S. FUHRMAN, X. WEN, AND R. SOMOGYI. **Cluster analysis and data visualization of large-scale gene expression data**. In *Pacific symposium on biocomputing*, **3**, pages 42–53, 1998.

[148] P. MITCHELL ET AL. **A perspective on protein microarrays**. *Nature biotechnology*, **20**(3):225–229, 2002.

[149] K. MITRA, N. NASIMUL, AND I. HITOSHI. **Reverse engineering gene regulatory network from microarray data using linear time-variant model**. *BMC Bioinformatics*, **11**, 2010.

[150] F. MORCHEN AND A. ULTSCH. **Optimizing time series discretization for knowledge discovery**. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 660–665. ACM, 2005.

[151] M. MORIYAMA, Y. HOSHIDA, M. OTSUKA, S.I. NISHIMURA, N. KATO, T. GOTO, H. TANIGUCHI, Y. SHIRATORI, N. SEKI, AND M. OMATA. **Relevance Network between Chemosensitivity and Transcriptome in Human Hepatoma Cells1**. *Molecular Cancer Therapeutics*, **2**(2):199–205, 2003.

[152] N. MORSHED AND M. CHETTY. **Combining instantaneous and time-delayed interactions between genes-a two phase algorithm based on information theory**. *AI 2011: Advances in Artificial Intelligence*, pages 102–111, 2011.

[153] N. MORSHED AND M. CHETTY. **Reconstructing genetic networks with concurrent representation of instantaneous and time-delayed interactions**. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1840–1847. IEEE, 2011.

[154] N. MORSHED, M. CHETTY, AND V.X. NGUYEN. **Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique**. *BMC Systems Biology*, **6**(1):62, 2012.

[155] N. MORSHED, M. CHETTY, V.X. NGUYEN, AND T. CAELLI. **mDBN: Motif Based Learning of Gene Regulatory Networks Using Dynamic Bayesian Networks**. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2013)*. ACM, 2013.

[156] N. MORSHED, M. CHETTY, AND N. VINH. **FusGP: Bayesian Co-learning of Gene Regulatory Networks and Protein Interaction Networks**. In *Neural Information Processing*, pages 369–377. Springer, 2012.

[157] NIZAMUL MORSHED AND MADHU CHETTY. **Information Theoretic Dynamic Bayesian Network Approach for reconstructing genetic networks**. In *Proceedings of the 11h IASTED International Conference on Artificial Intelligence and Applications*, pages 236–243, 2011.

[158] NIZAMUL MORSHED, MADHU CHETTY, AND NGUYEN XUAN VINH. **Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique**. In *Proceedings of the 18th international conference on Neural Information Processing - Volume Part II*, ICONIP'11, pages 248–257, Berlin, Heidelberg, 2011. Springer-Verlag. Available from: http://dx.doi.org/10.1007/978-3-642-24958-7_29.

[159] K. MURPHY. **The bayes net toolbox for matlab**. *Computing science and statistics*, **33**(2):1024–1034, 2001.

[160] N. NARIAI, S. KIM, S. IMOTO, S. MIYANO, ET AL. **Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks**. In *Pacific Symposium on Biocomputing*, **9**, pages 336–347, 2004.

[161] N. NARIAI, Y. TAMADA, S. IMOTO, AND S. MIYANO. **Estimating gene regulatory networks and protein–protein interactions of Saccharomyces cerevisiae from multiple genome-wide data**. *Bioinformatics*, **21**(suppl 2):ii206–ii212, 2005.

[162] N. NOMAN AND H. IBA. **On the Reconstruction of Gene Regulatory Networks from Noisy Expression Profiles**. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2543–2550. IEEE, 2006.

[163] N. NOMAN AND H. IBA. **Inferring gene regulatory networks using differential evolution with local search heuristics**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 634–647, 2007.

[164] N. NOMAN, H. IBA, ET AL. **Reverse engineering genetic networks using evolutionary computation**. *GENOME INFORMATICS SERIES*, **16**(2):205, 2005.

[165] US DEPARTMENT OF ENERGY. **Genomic Science Program Research**. Available from: http://genomicscience.energy.gov/research/index.shtml. Last Accessed: 05-09-2011.

[166] P. PAGEL, S. KOVAC, M. OESTERHELD, B. BRAUNER, I. DUNGER-KALTENBACH, G. FRISHMAN, C. MONTRONE, P. MARK, V. STÜMPFLEN, H.W. MEWES, ET AL. **The MIPS mammalian protein–protein interaction database**. *Bioinformatics*, **21**(6):832–834, 2005.

[167] SATCHIDANANDA PANDA, MARINA P ANTOCH, BROOKE H MILLER, ANDREW I SU, ANDREW B SCHOOK, MARTY STRAUME, PETER G SCHULTZ, STEVE A KAY, JOSEPH S TAKAHASHI, AND JOHN B HOGENESCH. **Coordinated transcription of key pathways in the mouse by the circadian clock**. *Cell*, **109**(3):307–320, 2002.

[168] J. PEARL. *Causality: models, reasoning, and inference*. Cambridge Univ Press, 2000.

[169] J. PEARL AND T.S. VERMA. **A theory of inferred causation**. 1991.

[170] JUAN M PEDRAZA AND ALEXANDER VAN OUDENAARDEN. **Noise propagation in gene networks**. *Science Signalling*, **307**(5717):1965, 2005.

[171] B.E. PERRIN, L. RALAIVOLA, A. MAZURIE, S. BOTTANI, J. MALLET, AND F. DALCHÉ BUC. **Gene networks inference using dynamic Bayesian networks**. *Bioinformatics*, **19**(suppl 2):ii138, 2003.

[172] R.J. PRILL, D. MARBACH, J. SAEZ-RODRIGUEZ, P.K. SORGER, L.G. ALEXOPOULOS, X. XUE, N.D. CLARKE, G. ALTAN-BONNET, AND G. STOLOVITZKY. **Towards a rigorous assessment of systems biology models: the DREAM3 challenges**. *PloS one*, **5**(2):e9202, 2010.

[173] R.J. PRILL, J. SAEZ-RODRIGUEZ, L.G. ALEXOPOULOS, P.K. SORGER, AND G. STOLOVITZKY. **Crowdsourcing Network Inference: The DREAM Predictive Signaling Network Challenge**. *Science's STKE*, **4**(189):mr7, 2011.

[174] NATAŠA PRŽULJ. **Protein-protein interactions: Making sense of networks via graph-theoretic modeling**. *Bioessays*, **33**(2):115–123, 2011.

[175] R. RAM AND M. CHETTY. **Learning Structure of a Gene Regulatory Network**. *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*, 2007.

[176] R. RAM AND M. CHETTY. **A Markov-Blanket-Based Model for Gene Regulatory Network Inference**. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **8**(2):353–367, 2011.

[177] R. RAM, M. CHETTY, AND T.I. DIX. **Causal Modeling of Gene Regulatory Network**. In *Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB'06. 2006 IEEE Symposium on*, pages 1–8. IEEE, 2006.

[178] R. RAM, M. CHETTY, AND T.I. DIX. **Fuzzy model for gene regulatory network**. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1450–1455. IEEE, 2006.

[179] A.K. RAMANI, R.C. BUNESCU, R.J. MOONEY, AND E.M. MARCOTTE. **Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome**. *Genome biology*, **6**(5):R40, 2005.

[180] JONATHAN M RASER AND ERIN K O'SHEA. **Control of stochasticity in eukaryotic gene expression**. *Science Signalling*, **304**(5678):1811, 2004.

[181] J. RISSANEN. **Stochastic complexity and modeling**. *The Annals of Statistics*, pages 1080–1100, 1986.

[182] JOSHUA W ROBINSON AND ALEXANDER J HARTEMINK. **Non-stationary dynamic Bayesian networks**. In *Procedding of Advances in Neural Information Processing Systems Conference*, 2008.

[183] DOV GREENBAUM YUVAL KLUGER NEVAN J. KROGAN SAMBATH CHUNG ANDREW EMILI MICHAEL SNYDER JACK F. GREENBLATT MARK GERSTEIN RONALD JANSEN, HAIYUAN YU. **Yale Gerstein Lab webpage**. Available from: http://networks.gersteinlab.org/intint/. Last Accessed: 23-05-2013.

[184] M. RONEN, R. ROSENBERG, B.I. SHRAIMAN, AND U. ALON. **Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics**. *Proceedings of the National Academy of Sciences*, **99**(16):10555, 2002.

[185] NITZAN ROSENFELD, JONATHAN W YOUNG, URI ALON, PETER S SWAIN, AND MICHAEL B ELOWITZ. **Gene regulation at the single-cell level**. *Science Signalling*, **307**(5717):1962, 2005.

[186] MICHAEL J ROSS, CHENG FAN, MICHAEL D ROSS, TE-HUATEARINA CHU, YUEYUE SHI, LEWIS KAUFMAN, WEIJIA ZHANG, MARY E KLOTMAN, AND PAUL E KLOTMAN. **HIV-1 infection initiates an inflammatory cascade in human renal tubular epithelial cells**. *JAIDS Journal of Acquired Immune Deficiency Syndromes*, **42**(1):1–11, 2006.

[187] E. SAKAMOTO AND H. IBA. **Inferring a system of differential equations for a gene regulatory network by using genetic programming**. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, **1**, pages 720–726. IEEE, 2001.

[188] L. SALWINSKI, C.S. MILLER, A.J. SMITH, F.K. PETTIT, J.U. BOWIE, AND D. EISENBERG. **The database of interacting proteins: 2004 update**. *Nucleic acids research*, **32**(suppl 1):D449–D451, 2004.

[189] U. SAUER, M. HEINEMANN, AND N. ZAMBONI. **GENETICS: Getting closer to the whole picture**. *Science Signalling*, **316**(5824):550, 2007.

[190] M.A. SAVAGEAU. **20 years of S-systems**. *Canonical Nonlinear Modeling: S-system Approach to Understand Complexity*, pages 1–44.

[191] M.A. SAVAGEAU AND R. ROSEN. *Biochemical systems analysis: a study of function and design in molecular biology*. Addison-Wesley Publishing Company, Advanced Book Program, 1976.

[192] KE SAWARYNSKI, A KAPLUN, G TZIVION, AND GS BRUSH. **Distinct activities of the related protein kinases Cdk1 and Ime2**. *Biochimica Et Biophysica Acta*, **1773**(3):450–456, 2007.

[193] M.H. SCHAEFER, J.F. FONTAINE, A. VINAYAGAM, P. PORRAS, E.E. WANKER, AND M.A. ANDRADE-NAVARRO. **HIPPIE: integrating protein interaction networks with experiment based quality scores**. *PloS one*, **7**(2):e31826, 2012.

[194] J. SCHÄFER AND K. STRIMMER. **An empirical Bayes approach to inferring large-scale gene association networks**. *Bioinformatics*, **21**(6):754–764, 2005.

[195] J. SCHÄFER AND K. STRIMMER. **Learning Large-Scale Graphical Gaussian Models from Genomic Data**. In *AIP Conference Proceedings*, **776**, page 263, 2005.

[196] T. SCHAFFTER, D. MARBACH, AND D. FLOREANO. **GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods**. *Bioinformatics*, 2011.

[197] G. SCHWARZ. **Estimating the dimension of a model**. *The annals of statistics*, **6**(2):461–464, 1978.

[198] E. SEGAL, M. SHAPIRA, A. REGEV, D. PE'ER, D. BOTSTEIN, D. KOLLER, AND N. FRIEDMAN. **Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data**. *Nature genetics*, **34**(2):166–176, 2003.

[199] E. SEGAL, H. WANG, AND D. KOLLER. **Discovering molecular pathways from protein interaction and gene expression data**. *Bioinformatics*, **19**(suppl 1):i264–i272, 2003.

[200] ROSS D SHACHTER AND MARK A PEOT. **Simulation approaches to general probabilistic inference on belief networks**. In *Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*, pages 221–234. North-Holland Publishing Co., 1990.

[201] NIALL SHANKS AND C RAY GREEK. *Animal models in light of evolution*. Brown Walker Press, 2009.

[202] C.E. SHANNON, W. WEAVER, R.E. BLAHUT, AND B. HAJEK. *The mathematical theory of communication*, **117**. University of Illinois press Urbana, 1949.

[203] E. SHANNON. **A mathematical theory of Communication**. *Bell Systems Technical Journal*, **27**:379–423, 1948.

[204] P. SHANNON, A. MARKIEL, O. OZIER, N.S. BALIGA, J.T. WANG, D. RAMAGE, N. AMIN, B. SCHWIKOWSKI, AND T. IDEKER. **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome research*, **13**(11):2498–2504, 2003.

[205] A. SHERMIN AND M.A. ORGUN. **Using dynamic bayesian networks to infer gene regulatory networks from expression profiles**. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 799–803. ACM, 2009.

[206] B. SHIPLEY. *Cause and correlation in biology: a user's guide to path analysis, structural equations and causal inference*. Cambridge University Press, 2002.

[207] I. SHMULEVICH, E.R. DOUGHERTY, AND W. ZHANG. **From Boolean to probabilistic Boolean networks as models of genetic regulatory networks**. *Proceedings of the IEEE*, **90**(11):1778–1792, 2002.

[208] ILYA SHMULEVICH, EDWARD R DOUGHERTY, SEUNGCHAN KIM, AND WEI ZHANG. **Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks**. *Bioinformatics*, **18**(2):261–274, 2002.

[209] G.P. SMITH ET AL. **Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface.** *Science (New York, NY)*, **228**(4705):1315, 1985.

[210] E.P. SOMEREN, LFA WESSELS, E. BACKER, AND MJT REINDERS. **Genetic network modeling**. *Pharmacogenomics*, **3**(4):507–525, 2002.

[211] R. SOMOGYI, S. FUHRMAN, M. ASKENAZI, AND A. WUENSCHE. **The gene expression matrix: towards the extraction of genetic network architectures**. *Nonlinear Analysis*, **30**(3):1815–1824, 1997.

[212] LE SONG, MLADEN KOLAR, AND ERIC P XING. **KELLER: estimating time-varying interactions between genes**. *Bioinformatics*, **25**(12):i128–i136, 2009.

[213] LE SONG, MLADEN KOLAR, AND ERIC P XING. **Time-varying dynamic bayesian networks**. *Advances in Neural Information Processing Systems*, **22**:1732–1740, 2009.

[214] P.T. SPELLMAN, G. SHERLOCK, M.Q. ZHANG, V.R. IYER, K. ANDERS, M.B. EISEN, P.O. BROWN, D. BOTSTEIN, AND B. FUTCHER. **Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization**. *Molecular biology of the cell*, **9**(12):3273–3297, 1998.

[215] P. SPIRTES, C. GLYMOUR, AND R. SCHEINES. *Causation, prediction, and search*, **81**. MIT press, 2001.

[216] C. STARK, B.J. BREITKREUTZ, A. CHATR-ARYAMONTRI, L. BOUCHER, R. OUGHTRED, M.S. LIVSTONE, J. NIXON, K. VAN AUKEN, X. WANG, X. SHI, ET AL. **The BioGRID interaction database: 2011 update**. *Nucleic acids research*, **39**(suppl 1):D698–D704, 2011.

[217] J. STÖCKEL, E.A. WELSH, M. LIBERTON, R. KUNNVAKKAM, R. AURORA, AND H.B. PAKRASI. **Global transcriptomic analysis of Cyanothece 51142 reveals robust diurnal oscillation of central metabolic processes**. *Proceedings of the National Academy of Sciences*, **105**(16):6156, 2008.

[218] KAI-FLORIAN STORCH, OVIDIU LIPAN, IGOR LEYKIN, N VISWANATHAN, FRED C DAVIS, WING H WONG, CHARLES J WEITZ, ET AL. **Extensive and divergent circadian gene expression in liver and heart**. *Nature*, **417**(6884):78–83, 2002.

[219] ROLAND B STOUGHTON. **Applications of DNA microarrays in biology**. *Annu. Rev. Biochem.*, **74**:53–82, 2005.

[220] J.W. STUCKI. **Stability analysis of biochemical systems–a practical guide.** *Progress in biophysics and molecular biology*, **33**(2):99, 1978.

[221] CHONG SU, JOSE M PEREGRIN-ALVAREZ, GARETH BUTLAND, SADHNA PHANSE, VINCENT FONG, ANDREW EMILI, AND JOHN PARKINSON. **Bacteriome. organ integrated protein interaction database for E. coli**. *Nucleic acids research*, **36**(suppl 1):D632–D636, 2008.

[222] N. SUGIMOTO AND H. IBA. **Inference of gene regulatory networks by means of dynamic differential bayesian networks and nonparametric regression**. *GENOME INFORMATICS SERIES*, **15**(2):121, 2004.

[223] JIMENG SUN, CHRISTOS FALOUTSOS, SPIROS PAPADIMITRIOU, AND PHILIP S YU. **Graphscope: parameter-free mining of large time-evolving graphs**. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696. ACM, 2007.

[224] J. SUZUKI. **A construction of Bayesian networks from databases based on an MDL principle**. In *Proceedings of the Ninth international conference on Uncertainty in artificial intelligence*, pages 266–273. Morgan Kaufmann Publishers Inc., 1993.

[225] D. SZKLARCZYK, A. FRANCESCHINI, M. KUHN, M. SIMONOVIC, A. ROTH, P. MINGUEZ, T. DOERKS, M. STARK, J. MULLER, P. BORK, ET AL. **The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored**. *Nucleic acids research*, **39**(suppl 1):D561–D568, 2011.

[226] J. TEGNER, M.K.S. YEUNG, J. HASTY, AND J.J. COLLINS. **Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling**. *Proceedings of the National Academy of Sciences*, **100**(10):5944–5949, 2003.

[227] J. TIAN. **A branch-and-bound algorithm for MDL learning Bayesian networks**. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 580–588. Morgan Kaufmann Publishers Inc., 2000.

[228] YUANYUAN TIAN, RICHARD C MCEACHIN, CARLOS SANTOS, JIGNESH M PATEL, ET AL. **SAGA: a subgraph matching tool for biological graphs**. *Bioinformatics*, **23**(2):232–239, 2007.

[229] J. TOEPEL, E. WELSH, T.C. SUMMERFIELD, H.B. PAKRASI, AND L.A. SHERMAN. **Differential transcriptional analysis of the cyanobacterium Cyanothece sp. strain ATCC 51142 during light-dark and continuous-light growth**. *Journal of bacteriology*, **190**(11):3904–3913, 2008.

[230] B. TURNER, S. RAZICK, A.L. TURINSKY, J. VLASBLOM, E.K. CROWDY, E. CHO, K. MORRISON, I.M. DONALDSON, AND S.J. WODAK. **iRefWeb: interactive analysis of consolidated protein interaction data and their supporting evidence**. *Database: the journal of biological databases and curation*, **2010**, 2010.

[231] P. UETZ, L. GIOT, G. CAGNEY, T.A. MANSFIELD, R.S. JUDSON, J.R. KNIGHT, D. LOCKSHON, V. NARAYAN, M. SRINIVASAN, P. POCHART, ET AL. **A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae**. *Nature*, **403**(6770):623–627, 2000.

[232] RJP VAN BERLO, EP VAN SOMEREN, AND MJT REINDERS. **Studying the conditions for learning dynamic Bayesian networks to discover genetic regulatory networks**. *Simulation*, **79**(12):689–702, 2003.

[233] E.P. VAN SOMEREN, L.F.A. WESSELS, AND M.J.T. REINDERS. **Linear modeling of genetic networks from experimental data**. In *Proc Int Conf Intell Syst Mol Biol (ISMB)*, **8**, pages 355–366, 2000.

[234] D.P.S. VERMA, R.B. GOLDBERG, ET AL. *Temporal and spatial regulation of plant genes.* Springer-Verlag, 1988.

[235] C. WALLACE, K.B. KORB, AND H. DAI. **Causal discovery via MML**. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 516–524. Citeseer, 1996.

[236] TIE WANG, JEFFREY W TOUCHMAN, AND GUOLIANG XUE. **Applying two-level simulated annealing on Bayesian structure learning to infer genetic networks**. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 647–648. IEEE, 2004.

[237] W. WANG, B.K. GHOSH, AND H. PAKRASI. **Identification and Modeling of Genes with Diurnal Oscillations from Microarray Time Series Data**. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **8**(1):108–121, 2011.

[238] THE STATISTICS HOW TO WEBSITE. **Pearson Correlation Coefficient**. Available from: http://www.statisticshowto.com/articles/what-is-the-pearson-correlation-coefficient/. Last Accessed: 23-05-2013.

[239] A.V. WERHLI, D. HUSMEIER, ET AL. **Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge**. *Stat Appl Genet Mol Biol*, **6**(1):15, 2007.

[240] M.L. WHITFIELD, G. SHERLOCK, A.J. SALDANHA, J.I. MURRAY, C.A. BALL, K.E. ALEXANDER, J.C. MATESE, C.M. PEROU, M.M. HURT, P.O. BROWN, ET AL. **Identification of genes periodically expressed in the human cell cycle and their expression in tumors**. *Molecular biology of the cell*, **13**(6):1977–2000, 2002.

[241] WIKIPEDIA. **The central dogma of molecular biology**. Available from: http://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology. Last Accessed: 23-05-2013.

[242] WIKIPEDIA. **Gene**. Available from: http://en.wikipedia.org/wiki/Gene. Last Accessed: 23-05-2013.

[243] WIKIPEDIA. **Hemoglobin**. Available from: http://en.wikipedia.org/wiki/Hemoglobin. Last Accessed: 23-05-2013.

[244] WIKIPEDIA. **Protein-protein interaction**. Available from: http://en.wikipedia.org/wiki/Protein-protein_interaction. Last Accessed: 23-05-2013.

[245] WIKIPEDIA. **RNA**. Available from: http://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg. Last Accessed: 23-05-2013.

[246] B. WILCZYŃSKI AND N. DOJER. **BNFinder: exact and efficient method for learning Bayesian networks**. *Bioinformatics*, **25**(2):286, 2009.

[247] ZHI XIE AND DON KULASIRI. **Modelling of circadian rhythms in¡ i¿ Drosophila¡/i¿ incorporating the interlocked PER/TIM and VRI/PDP1 feedback loops**. *Journal of theoretical biology*, **245**(2):290–304, 2007.

[248] Z. XING AND D. WU. **Modeling multiple time units delayed gene regulatory network using dynamic Bayesian network**. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 190–195. IEEE, 2006.

[249] XIE L XU, JAMES M OLSON, AND LUE PING ZHAO. **A regression-based method to identify differentially expressed genes in microarray time course studies and its application in an inducible Huntington's disease transgenic model**. *Human Molecular Genetics*, **11**(17):1977–1985, 2002.

[250] N. XUAN, M. CHETTY, R. COPPEL, AND P.P. WANGIKAR. **Gene regulatory network modeling via global optimization of high-order dynamic Bayesian network**. *BMC bioinformatics*, **13**(1):131, 2012.

[251] XIFENG YAN, X ZHOU, AND JIAWEI HAN. **Mining closed relational graphs with connectivity constraints**. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 324–333. ACM, 2005.

[252] YEE HWA YANG, SANDRINE DUDOIT, PERCY LUU, DAVID M LIN, VIVIAN PENG, JOHN NGAI, AND TERENCE P SPEED. **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation**. *Nucleic acids research*, **30**(4):e15–e15, 2002.

[253] M.K.S. YEUNG, J. TEGNÉR, AND J.J. COLLINS. **Reverse engineering gene networks using singular value decomposition and robust regression**. *Proceedings of the National Academy of Sciences*, **99**(9):6163–6168, 2002.

[254] KH YOUNG. **Yeast two-hybrid: so many interactions,(in) so little time...** *Biology of reproduction*, **58**(2):302–311, 1998.

[255] J. YU, V.A. SMITH, P.P. WANG, A.J. HARTEMINK, AND E.D. JARVIS. **Advances to Bayesian network inference for generating causal networks from observational biological data**. *Bioinformatics*, **20**(18):3594, 2004.

[256] Y. YUAN, C.T. LI, AND O. WINDRAM. **Directed partial correlation: inferring large-scale gene regulatory network through induced topology disruptions**. *PLoS One*, **6**(4):e16835, 2011.

[257] M.Q. ZHANG ET AL. **Promoter analysis of co-regulated genes in the yeast genome**. *Computers & chemistry*, **23**(3-4):233–250, 1999.

[258] W. ZHAO, E. SERPEDIN, AND E.R. DOUGHERTY. **Inferring connectivity of genetic regulatory networks using information-theoretic criteria**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 262–274, 2007.

[259] SHUHENG ZHOU, JOHN LAFFERTY, AND LARRY WASSERMAN. **Time varying undirected graphs**. *arXiv preprint arXiv:0802.2758*, 2008.

[260] GEFENG ZHU, PAUL T SPELLMAN, TOM VOLPE, PATRICK O BROWN, DAVID BOTSTEIN, TRISHA N DAVIS, AND BRUCE FUTCHER. **Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth**. *Nature*, **406**(6791):90–94, 2000.

[261] P. ZOPPOLI, S. MORGANELLA, AND M. CECCARELLI. **TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach**. *BMC bioinformatics*, **11**(1):154, 2010.

[262] M. ZOU AND S.D. CONZEN. **A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data**. *Bioinformatics*, **21**(1):71, 2005.

[263] N. Bushati AND S. M. Cohen. **microRNA functions.** Annu. Rev. Cell Dev. Biol. 23 : 175-205, 2007.