REVEALING GENE INTERACTIONS USING BAYESIAN NETWORKS

by

Şenol İşçi

B.S., in Physics, Boğaziçi University, 2000

M.S., in Biomedical Engineering, Boğaziçi University, 2003

Submitted to the Institute of Biomedical Engineering

in partial fulfillment of the requirements

for the degree of

Doctor

of

Philosophy

Boğaziçi University

2013

ACKNOWLEDGMENTS

I would like to offer my sincerest gratitude to my thesis advisors Assist. Prof. Hasan H. Otu and Prof. Cengizhan Öztürk for their continuous support and guidance. They inspired me not only for this thesis but also in every aspects of my academic and professional life. I am grateful to thank Prof. Ahmet Ademoğlu, Assoc.Prof. Ata Akın, Assoc.Prof. Albert Güveniş and Assoc.Prof. Duran Üstek for being in my thesis committee and provide valuable suggestions.

I would like to thank all Biomedical Engineering faculty, staff and students for providing me with valuable experiences.

I would like to dedicate this thesis to my loving wife Dilek, my sun Onur Eren and my daughter Elif Sude.

ABSTRACT

REVEALING GENE INTERACTIONS USING BAYESIAN NETWORKS

High throughput biological data (HTBD) targeting understanding of biochemical interactions in the cell can best be analyzed, and explained within the context of networks and pathways. Such data generally represents stochastic nonlinear relations embedded in noise. Bayesian Network (BN) theory provides a framework to analyze the data regarding gene regulation measurements, as this framework naturally handles the aforementioned obstacles.

In this dissertation, we provide a two faceted approach to the applications of BNs to HTBD. In the first facet, a novel method is provided, which models known biological pathways as BNs, and uses given HTBD to find pathways that best explain underlying interactions. During this process, biological pathways are converted to directed acyclic graphs, and a score measuring fitness of the observed HTBD to a given network is calculated. Statistical significance of these scores is assessed by "randomization via bootstrapping", and relevant pathways are identified with a certainty that can be used as a comparative measure. Simulations using synthetic and real data demonstrated robustness of the proposed approach, called Bayesian Pathway Analysis (BPA). BPA provides improvement over existing similar approaches by not considering genes in a pathway simply as a list, but incorporating to its model the topology via which genes in a given pathway interact with each other. Although network learning techniques are very useful to reveal the underlying biological phenomena with the help of HTBD, these techniques do not always perform well. This is due to the problems created by the small number of samples, inconvenient initial choice for the network structures, noise inherent in the data, and the complexity of the networks. To improve their performance, the learning techniques can be supported by prior biological knowledge, which are already verified by experimental assays.

In the second facet explored in this dissertation, we established a global approach to integrate known biological information to Bayesian learning in order to reveal gene interactions. The proposed framework makes use of external biological knowledge to predict if two given genes interact with each other. To this end, prior knowledge about interaction of two genes is utilized by generating a Bayesian Network Prior (BNP) model, using existing external biological knowledge. External knowledge types to be utilized were obtained from interaction databases such as BioGrid and Reactome, and consist of protein-protein, protein-DNA/RNA, and gene interactions. The resulting model is incorporated into greedy search algorithm for learning networks from HTBD, and interacting genes are represented in the form of a network. In this process of network generation, the BNP model deducing gene interactions from external knowledge are used to calculate the probability of candidate networks to enhance the structure learning task. Simulations on both synthetic and real data sets showed that the proposed framework can successfully enhance identification of the true network, and be used in predicting gene interactions.

Keywords: Bayesian Networks, Gene Networks, Gene Interaction, Microarray, Data Integration, Pathway Analysis, Probabilistic Graphical Models

ÖZET

GEN ETKİLEŞİMLERİNİN BAYEZYEN AĞLAR İLE ORTAYA ÇIKARILMASI

Hücre içindeki biyokimyasal etkileşimlerin anlaşılmasını hedefleyen yüksek çıktılı biyolojik veriler, en iyi ağ ve patikalar bağlamında analiz edilebilir ve açıklanabilir. Bu veri, genelde gürültü içine gömülü lineer olmayan stokastik ilişkileri temsil eder. Bayezyen Ağ teorisi, gen düzenleme ölçümleriyle ilgili verileri analiz etmek için bir çatı sağlar. Çünkü bu çatı bahsi geçen engelleri doğal olarak ele alır. Bu tezde, altta yatan biyolojik etkileşimleri en iyi açıklayan patikaları bulmak için, bilinen biyolojk patikaları Bayezyen ağlar olarak modelleyen ve verilen bir mikrodizi deneyinin sonuçlarını yansıtan yeni bir yöntem verilmiştir. Bu işlem sırasında, biyolojik patikalar, yönlü çevrimsiz graflara dönüştürülür ve gözlenmiş mikrodizi verisinin verilen bir ağa ne kadar uyduğunu ölçen bir skor hesaplanır. Bu skorların istatistiksel önemi, "önyükleme ile rasgeleleştirme yöntemi" ile değerlendirilir ve uygun patikalar, karşılaştırma ölçüsü olarak kullanılabilecek bir kesinlik ile tespit edilir. Sentetik ve gerçek veri kullanılan simülasyonlar, Bayezyen Patika Analizi (BPA) olarak adlandırılan bu önerilen yöntemin sağlamlığını göstermiştir. Önerilen yöntem, var olan benzer yöntemlere göre gelişme sağlamıştır. Çünkü, bir patikada bulunan genler basitçe bir liste şeklinde düşünülmez ve verilen bir patikadaki hangi genlerin birbiriyle etkileştiğinin topolojisi modelle birleştirilir.

Ağ öğrenme teknikleri, altta yatan biyolojik olayları mikrodizi deneyleri yardımıyla ortaya çıkarmaya çok yararlı olmasına rağmen, bu ağlar gerçek biyolojik patikalara çok uzak olabilir. Bunun sebebi, veri içinde genler için az sayıda örnek olmasından dolayı ve uygun olmayan başlangıç ağ yapısı seçiminden kaynaklanan problemlerdir. Öğrenme teknikleri, önceden deneysel testlerle doğrulanmış öncül biyolojik bilgi ile desteklenmelidir. Bilinen genler ve düzenleyici patikalar hakkındaki bilgilerin kullanılması, statik ve dinamik Bayezyen ağların öğrenilmesinin doğruluk ve performansına en ileri zemini sağlayabilir. Ama, bilindiği kadarıyla, gen etkileşim ağlarında her tip

bilgi kaynağından gelen harici öncül bilginin kullanılmasına yönelik genel bir kurulu metodoloji bulunmamaktadır.

Bu tezin amaçlarında biri, gen etkileşimlerinin ortaya çıkarılması için hem statik hem de dinamik Bayezyen ağ öğrenme işlemine, bilinen biyolojik bilgileri entegre etmek için global bir yaklaşımın kurulmasıdır. Harici biyolojik verilerden faydalanarak iki genin birbiriyle etkileşip etkileşmediğini tahmin etmek için bir çatı önerilmiştir. Buna yönelik olarak, bilinen biyolojik veriyi, gen etkileşimlerinin ortaya çıkarılmasında kullanabilmek adına, var olan biyolojik verilerden Bayezyen ağ üretilmiştir. Kullanılan harici veri tipleri, protein-protein, protein-DNA/RNA ve gen etkileşimlerinden oluşmakta olup, BioGrid ve Reactome gibi etkileşim veritabanlarından elde edilmiştir. İlk olarak, bilinen gen etkileşimlerini kullanarak parametre öğrenme yöntemi kullanılarak, verilecek iki genin etkileşip etkileşmediğini tahmin edecek Öncül Bayezyen Ağ (ÖBA) modeli inşa edilmiştir. Elde edilen model, yüksek çıktılı biyolojik veriden ağ öğrenme için greedy arama algoritmasına entegre edilir ve etkileşen genler bir ağ formunda sunulur. Bu ağ üretimi işleminde, harici biyolojik veriden etkileşen genleri bulan Öncül Bayezyen Ağ (ÖBA), yapı öğrenme görevinde aday ağların olasılığını hesaplamak için kullanılır. Hem sentetik hem gerçek veri setleri ile yapılan simülasyonlar göstermiştir ki önerilen çatı, gerçek ağların belirlenmesini başarılı bir şekilde geliştirebilir ve gen etkileşimlerinin tahmin edilmesinde kullanılabilir.

Anahtar Sözcükler: Bayezyen Ağlar, Gen Ağları, Gen Etkileşimi, Mikrodizi, Veri Bütünleştirme, Patika Analizi, Olasılıksal Grafsal Modelleme

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_{ijk}	Dirichlet distribution hyper-parameters
A_G	The adjacency matrix of the candidate graph G.
$\boldsymbol{B}$	The prior information matrix
c	Constant
C_1	Number of samples in the first group in a dataset
C_2	Number of samples in the second group in a dataset
d	Number of parameters in BN model
$\boldsymbol{D}$	Dataset
e	An edge of a graph
E	The energy function
F	A random variable with (ρ) density function
g_{i1j}	Values of the i^th node in j^{th} samples in SG_1
g_{i2k}	Values of the i^th node in k^{th} samples in SG_2
G	Graph
G^*	Valid directed graph
I	Indicator function in bootstrapping
M_{ij}	Count of parents' j configuration cases of node i
N	Number of nodes in BN model
N_{ij}	Sum of Dirichlet distribution hyper-parameters a_{ijk}
O	An ordered set of observations
o_l	The l^{th} element of O
P(G|D)	Posterior probability
$P(D\|G)$	Marginal probability
$P(G)$	Prior probability of the graph
$P(D)$	Probability of observed data
$Pa(X_i)$	Vector of parent states of a node
q_i	Number of different states of node's parents
r_i	Set of values a node can take on.

$\boldsymbol{R}$	Randomized data sets in bootstrapping
s_{ijk}	Total count for node i is observed
SG_1	The first group of samples in the data set
SG_2	The second group of samples in the dataset
S_n	BDe score calculated for n^{th} BN in bootstrapping
X_i	Random variable in a Bayesian network
v	A vertex of a graph
Z	Partition function
$\beta(f;a,b)$	Beta distribution for F with parameters
β	Hypermarameter in $P(G)$ formula
β_H	Upper bound of hyperparameter β
β_L	Lower bound of hyperparameter β
C	Scaling constant in informative structure prior formula
$\Delta\beta$	Range of values in the interval $[\beta_L, \beta_H]$
δ	Symmetrical difference
H_I	Categories for interaction energy
κ	A fixed integer
π_n	Parent set of a node n in DBN model
ρ	Beta density function
θ_i	Parameters of the local probability distribution
ρ	Set of all possible network structures
τ	Time delay/the order of the DBN model
$\boldsymbol{U}$	The matrix derived from $\boldsymbol{B}$ and A_G
$\boldsymbol{U}$(i,j)	Elements of $\boldsymbol{U}$ for genes i to j
$\boldsymbol{U}_{ij}$	Interaction energy of the edge

LIST OF ABBREVIATIONS

A	Adenine
AUC	Area Under the Curve
BDe	Bayesian Dirichlet Equivalent
BIC	Bayesian Information Criterion
BPA	Bayesian Pathway Analysis
BN	Bayesian Network
BNP	Bayesian Network Prior
BNT	Bayes Net Toolbox
C	Cytosine
CO	Cut-off Values
CPD	Conditional Probability Distribution
CPT	Conditional Probability Table
cRCC	Clear-cell Renal Cell Carcinoma
chRCC	Chromophobe Renal Cell Carcinoma
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DNA	Deoxyribonucleic Acid
EM	Expectation- Maximization
FDR	False Discovery Rate
FAN	Functional Association Network
FC	Fold Change
G	Guanine
GEO	Gene Expression Omnibus
GI	Gene Interaction
GGM	Gaussian Graphical Models
GO	Gene Ontology
GRN	Gene Regulatory Network
GSEA	Gene Set Enrichment Analysis

HTBD	High Throughput Biological Data
IGA	Individual Gene Analysis
KEGG	Kyoto Encyclopedia of Genes and Genomes
LPD	Local Probability Distribution
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
NCBI	National Center for Biotechnology Information
OC	Oncocytomas
pRCC	Papillary Renal Cell Carcinoma
PDAG	Partially Directed Acyclic Graph
PPI	Protein-Protein Interaction
RCC	Renal Cell Carcinoma
RefExA	Reference Database for Gene Expression Analysis
REVEAL	REVerse Engineering ALgorithm
RNA	Ribonucleic Acid
ROC	Receiver Operating Characteristic
SEM	Structural Equation Model
T	Thymine
TCC	Transitional Cell Cancer
TF	Transcription Factor
TR	Transcriptional Regulatory
U	Uracil
WT	Wilms' tumors

1. INTRODUCTION

1.1 Motivation

In order to understand the function of a cell or of higher units of biological organization, the roles of genes and their interactions should be described. In a systems biology perspective, it is useful to treat them as systems of interacting elements, which needs the following information: the identity of the genes that constitute the underlying system; the dynamic behavior of these genes (i.e., how their abundance or activity changes over time in various conditions); and the interactions among these components [1]. It is known that each cell in a biological organism contains the same genes, but only a fraction of genes are expressed at a given time. Many diseases result from the deregulated interactions of genes. Therefore, in order to develop new treatments for diseases such as discovering new drugs, it is important to understand the mechanism that determines which genes are expressed, when these genes are expressed, the sequence of their expression and the level of their individual expression. Also, a better comprehension of the roles performed by genes in cellular functions and processes can be best illustrated in the form of gene networks, such as pathways.

1.2 The Problem Statement

Gene network inference/learning problem is defined as follows; how to uncover underlying gene network using probabilistic and machine learning techniques and integration of HTBD (e.g. gene expression microarray data) with prior biological knowledge. Construction and revelation of the gene networks is useful for answering what genomic functions are performed by interacting genes, how do these genes perform their functions, and when these genes are expressed. The inference process of gene interactions from the microarray expression data is non-trivial and remains as one of the most challenging tasks of systems biology.

The learning process is highly challenging due to the complexity in finding possible network structures from data in which the number of observed variables (e.g. genes) is higher as opposed to the low number of observations (samples). Problem related to this is that the computational complexity to estimate the real network among a vast number of probable network structures for a dataset containing thousands of genes overwhelms current computational capabilities. This led to research on computational techniques that are necessary to estimate a gene network which contains multiple variables, parameters and constraints. Because of the complexity of gene interaction networks in addition to sparse and noisy nature of experimental data, machine learning and statistical methods may lead to poor reconstruction accuracy of the underlying network, and therefore it is advantageous to make network inferences using prior biological knowledge.

The inference of gene interaction networks from HTBD is an important and challenging task in systems biology. Several machine learning and statistical methods have been proposed for the problem [2], and BN models have gained popularity for the task of inferring gene networks [3]. Most BN structure learning algorithms are based on heuristic search techniques utilizing maximum likelihood or marginal likelihood because of the infeasible computational complexity. However, structure learning with the likelihood approximation may lead to a false model not only because of the heuristic nature of the algorithm, but also because of the assumption that each candidate graph has the same probability. Informative priors generated from existing biological information can improve learning to get better models to describe the underlying gene interactions. In several studies, the use of prior biological knowledge of the gene interaction network in conjunction with gene expression data has been suggested to improve the fidelity of network reconstruction. Hartemink *et al.* [4] incorporated genomic location data to guide the Bayesian network model inference. Tamada *et al.* [5] proposed a method which iteratively detects consensus motifs based on the structure of the estimated network model, and then evaluates the network using the result of the motif detection until the inferred network becomes stable. Imoto *et al.* [6] proposed a framework for inferring gene networks using prior biological knowledge in addition to gene expression data. To do this, they introduced an energy function, where each edge

in the network was assigned an energy value, and assumed that the probability of the network depends on the Gibbs distribution. Werhli *et al.*[7] extended this approach to integrate multiple sources of prior knowledge into dynamic Bayesian network (DBN) learning via MCMC sampling. Murkherjee *et al.* [8] proposed a scheme to incorporate known network features including edges, classes of edges, degree distributions, and sparsity into gene network reconstruction within a Bayesian learning framework with MCMC sampling. However, these studies were limited in the use of external biological knowledge by incorporating only certain features, such as network topology or binding sites in promoter regions. Furthermore, in the aforementioned approaches, manual curation and/or manual incorporation of the external knowledge are employed.

In brief, how to reveal dynamics of biological phenomena by building or inferring gene networks from experimental data has become prominent, and this dissertation is aimed to solve this problem.

1.3 Aims of the Study

High throughput biological data (HTBD) is generated in a variety of ways including through deep sequencing and microarrays. This data can provide a snapshot of regulatory processes in the cell. This is a significant change from the traditional molecular biology approach of focusing on single molecules and reactions. The priority is now more data-driven and more computationally intensive, and there is great need to find methods that can handle the massive data in a global manner and that can analyze data originating from large systems. Arguably, the most popular HTBD type is microarrays, where identification of differentially expressed genes between two groups of samples initially relied on individual gene analysis (IGA). An alternative approach, called pathway analysis, functional enrichment analysis, or gene set analysis [9], which focuses on directly determining predefined gene sets or classes that are significantly regulated, has received a great deal of attention. Gene set analysis (GSA) methods score groups of genes, and can identify genes that exhibit subtle changes at an individual level, but show concordant enrichment within a set [10].

In particular, Bayesian network (BN) models have gained popularity for the task of learning biological networks and pathways from microarray gene expression data [3, 11]. In gene network modeling studies using BNs, nodes generally represent the expression level of a gene, and edges represent the relationship between genes. BN models capture both linear and nonlinear interactions between sets of random variables, and handle stochastic events in a probabilistic framework accounting for noise. This results in the emphasis of only the strong relations in the observed data. BNs are, therefore, viable candidates for modeling gene regulation systems where stochastic effects and large amounts of noise are expected. Furthermore, BNs are able to focus on local interactions where each node is directly affected by a relatively small number of nodes, and interactions defined by a BN can be related to causal inference [3]. These properties are similarly observed in biological networks, justifying the use of BNs in exploring pathways in the setting of gene interaction networks using HTBD.

One of the aims of this dissertation is to develop a new algorithm of pathway analysis, which uses a graph theoretic approach and BN theory to model biological pathways, and evaluate whether a pathway successfully describes the underlying HTBD by scoring the fitness of the network [12]. Previously described GSEA [10] or Gene Ontology [13] based methods do not take into account the connectiveness of the analyzed gene lists. There have been methods proposed to take into account the GO graph topology [14], overlap between GO categories in the GO hierarchy [15], or modeling interactions between GO categories [16] in assessing the significance of enrichment of a GO term based on experimental data. However, none of these methods take into account the network or structure defining the relation between the genes in each category.

Our simulations using synthetic data demonstrated robustness of the Bayesian Pathway Analysis (BPA) approach. We tested the proposed method on human microarray data regarding Renal Cell Carcinoma (RCC) and compared our results with gene set enrichment analysis. BPA was able to find broader and more specific pathways related to RCC.

In addition to the experimental gene expression data, a range of other data types can be used as prior knowledge to enhance the reconstruction of gene networks. Due to technological advances in sequencing, microarray, proteomics and related fields, biological and clinical data have been produced at an enormous rate. A list of 1380 biological databases in categories such as nucleotide, RNA, protein sequence, pathway, organelle, proteomic has been reported [17].

BNs have a number of features which make them attractive candidates for combining prior knowledge and data, dealing with uncertainty, avoiding overfitting a model to training data, and learning from incomplete datasets. BNs handle stochastic events in a probabilistic framework accounting for noise, which results in emphasizing only the strong relations in the observed data. Furthermore, BNs are able to focus on local interactions where each node is directly affected by a relatively small number of nodes [3], and interactions defined by a BN can be related to causal inference [18].

The aforementioned properties are similarly observed in biological networks, justifying the use of BNs in exploring pathways in the setting of gene interaction networks using gene expression data. Learning algorithms for both the structure and parameters of BNs have been developed [19]. Most of the research on BNs has focused on directed acyclic graphs (DAGs) and static systems with discrete variables, and/or linear Gaussian models. Friedman *et al.* used BNs to generate a causal model of the yeast cell cycle data, using either a model with discretized expression levels (e.g. Boolean, or underexpressed/normal/overexpressed), or a linear Gaussian model [3]. The latter treats the expression level of a gene as being normally distributed around a mean which is a linear sum of inputs. Therefore, rather than the true causal relationships, the results may represent co-regulation of genes. Friedman *et al.* [20] introduced a method to sample network structures from the posterior distribution with the Markov chain Monte Carlo (MCMC) model. Most BN structure learning algorithms are based on heuristic search techniques utilizing maximum likelihood or marginal likelihood because of the infeasible computational complexity. However, structure learning with likelihood approximation may lead to a false model. Informative structure priors generated from existing biological information can improve the learning process to get better models

that describe the underlying gene interactions.

One of the aim of this dissertation is to develop a framework that incorporates multiple sources of prior knowledge, regardless of its type, into BN learning. The meaning of prior knowledge in our context is the enumeration of pairwise interactions of genes from biological information sources, and is the use of this information in BN modeling of gene expression domain.

External knowledge types utilized were obtained from interaction databases such as BioGrid and Reactome and consisted of protein-protein, protein-DNA/RNA, and gene interactions. First, a Bayesian Network Prior (BNP) model was created to predict if two genes interact by employing parameter learning using known gene interactions. The resulting model was incorporated into the greedy search heuristic learning algorithms to learn networks from HTBD, and interacting genes were represented in the form of a gene network. In this process of network generation, the BN model deducing gene interactions from external knowledge were used to calculate the probability of candidate networks to enhance the structure learning task. Our simulations on both synthetic and real data sets showed that the proposed framework can successfully enhance identification of the true network, and be used in building biologically plausible networks.

1.4 Overview of the Dissertation

Chapter 1 includes the motivationsi, the problem statement, the aims of the study and the overview of the dissertation.

Chapter 2 presents background information on Computational Systems Biology. First, the rationale for systems biology is presented. Its basic principles are explained, including definition, tasks, and objectives of systems biology, computational systems biology techniques and its impact on current genomic studies. This chapter includes a short background on fundamentals of molecular biology, consisting of description

of DNA, RNA, proteins and their processing. Microarray technology is explained. Chapter 2 gives definition, applications and types of gene networks, and computational models in Systems Biology to model gene networks including co-expression networks, Gaussian graphical models, dependency networks, and Bayesian networks.

Chapter 3 describes the theory of Bayesian Networks. A short mathematical basics of the theory is given, and many aspects and concepts of Bayesian networks are explained including static and dynamic Bayesian networks, local probability distribution, conditional independence, and Markov equivalence. This chapter includes detailed description of score based structure learning of Bayesian networks, as this dissertation is mainly based on learning structure of gene networks, using Bayesian network models. Scoring schemes are presented, and model search and selection methods including a short description of search heuristics are explained.

Chapter 4 includes details of the Bayesian Pathway Analysis method, proposed in this dissertation for the first time, that models biological pathways as Bayesian networks, and identifies pathways that best explain given high throughput biological data by scoring fitness of each network. Overall methodology and results are given in detail.

Chapter 5 presents a framework to incorporate multiple sources of prior knowledge, regardless of its type, into Bayesian network learning to rigorously harness, and use the existing wide range of biological information. This chapter starts with the description of the complete methodology including the proposed Bayesian Network Prior model, a novel structure prior function, and the description of integration of these improvements into the Bayesian network structure learning algorithms. This chapter includes simulations and the results of the proposed algorithm, and models on both synthetic and real data in the context of Bayesian networks.

Chapter 6 includes discussion, conclusions and advancements in revealing gene networks, based on the novel methods proposed in this dissertation.

2. COMPUTATIONAL SYSTEMS BIOLOGY

2.1 Basic Principles

Living organisms are complex and heterogeneous biological systems built on the structural and functional units called cells. A living organism may be composed of a single cell, as in unicellular organisms, or many cells, as in multi-cellular organisms. The cell is accepted to be the smallest functional unit of life. A biological system such as mammalians has several levels of organization: At the lowest level, an organism is, chemically, composed mostly of the elements carbon, hydrogen, oxygen, and nitrogen. These elements are combined into organic compounds such as carbohydrates, lipids (fats and oils), DNA, RNA, proteins, nucleic acids, and vitamins. Biological compounds form organelles in a cell. Cells are organized into tissues. Tissues are arranged to form organs, which are grouped to form an organism.

The study of biological systems cannot be limited to simply listing organizational levels and their parts (such as proteins, genes, cells, etc.). Although an exhaustive list of all the parts of a system may give a vague impression, it does not necessarily help one to understand how the system functions. Biological systems are dynamic and their parts operate vastly on different temporal scales, from microseconds to years, and spatial scales, from nanometers to meters. This complexity makes it extremely challenging to understand how even the simplest organism functions. However, a holistic view of biological systems can demonstrate how these parts are assembled together, and how they interact with each other and with the surrounding environment. In other words, a system-level understanding is required. This is the objective of Systems Biology.

Systems Biology has gained prominence in recent years due to several advancements: biological knowledge with the prospect of utilization in biotechnology and health care has been improved; high-throughput experimental techniques for making measurements of the biological quantities have become widely available; classical mathematical

modeling of biological processes has been an active field of research; computer power for simulation of complex systems has improved tremendously; storage and retrieval capability in large databases and data mining techniques have been developed; Internet has become the medium for the widespread availability of multiple sources of knowledge.

Process of a systems biology approach means investigating the components of cellular networks and their interactions, applying experimental high-throughput and whole-genome techniques, and integrating computational and theoretical methods with experimental efforts to understand the underlying biological phenomena.

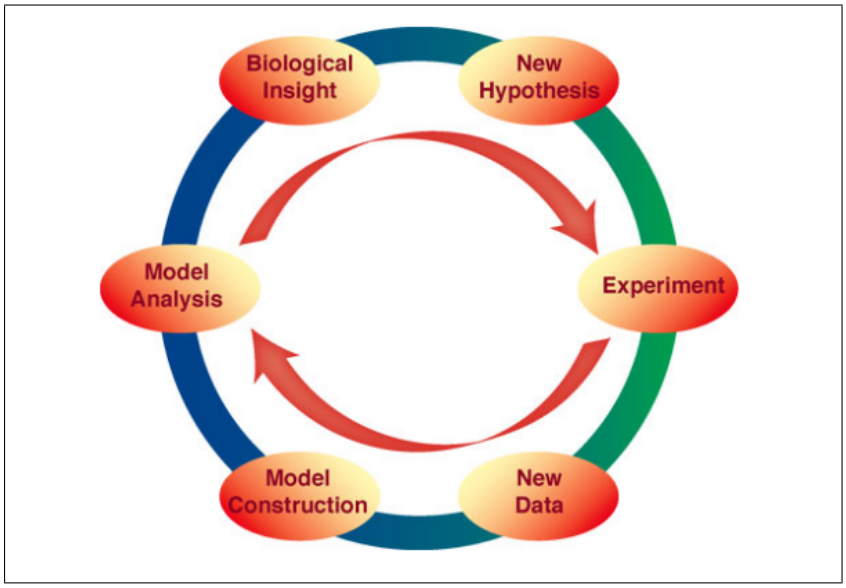

Figure 2.1 Process of Systems Biology.

Computational Systems Biology attempts to undertake the task of integration of genomics, proteomics, and actually all the emerging "omic" disciplines, with the aim of understanding biological systems. The genome is the full complement of genetic information. Biological systems contain two main types of programmatic information: genes which encode the proteins through the intermediary of RNA, and regulatory networks which specify how these genes are expressed in time and space. People have introduced other "omes" in analogy to the genome: the transcriptome is the entirety of RNA transcripts which are produced by a cell; the proteome is the full complement of proteins; the metabolome is the full complement of metabolites, small molecules involved in metabolism; the interactome is the complete set of molecular interactions.

The microarray technology and more recently the high-throughput sequencing technology makes it possible to measure the entire expressed transcriptome in a tissue or cell culture. The mass spectrometry and the NMR technology allows high-throughput quantification of metabolites. These technologies provide us with the opportunity to examine the cell, and allow us to develop and compare Systems Biology models of cellular function.

Many aspects of Systems Biology are relevant to Computer Science. Computational Systems Biology is the application of formal (i.e. mathematical or computational) modeling to help understand biological function, dynamics, and interactions. Computational Systems Biology models link different components of the system, and are therefore often based on networks. The biological function emerges from the collective behavior of components linked by networks. By building models and studying their properties, we can gain important insights into some fundamental biological processes. Numerical computation is required to accurately simulate models, and statistical machine learning is useful in learning model parameters and scoring alternative models[21].

2.2 Molecular Biology Background

A major breakthrough in Life Sciences is the sequencing of the genomes of several species, including the first draft of the human genome in 2000 [22]. The genome is the entirety of an organism's hereditary information, and it carries the instructions for making the proteins, and other molecules that cells are built from. The genome defines the structure and function of the cell. It includes both the coding sequences of the DNA (deoxyribonucleic acid) and the non-coding sequences of the DNA.

Complexity of a cell with respect to its structure and function are mainly embodied in and regulated by three biological sequences: DNA, RNA (ribonucleic acid) and protein. DNA is the hereditary material in multicellular organisms such as human [23]. Most DNA is contained within chromosomes in the cell nucleus, which is called

nuclear DNA, but a small amount of DNA can also be found in the mitochondria, which is called mitochondrial DNA.

DNA is a long sequential molecule which is made up of a chain of four types of chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a nucleotide. Nucleotides are arranged in two long strands that form a spiral called a double helix. DNA is the carrier of genes and other regulatory information. DNA has a direction such that the information contained in a sequence on one strand flows from one end (called the 5' end) to the other (3' end). This direction is reversed on the opposite strand. Genetic information can be encoded on both strands.

Genes are segments of the DNA sequence that encode the instructions for making a gene product (protein or RNA) through a process called gene expression. Genes contain regulatory coding sequences that either increases or decreases its transcription rate. In eukaryotes, coding sequences (exons) are interlaced with non-coding sequences (introns). The DNA is a template for making ribonucleic acids(RNAs) through a process called transcription. The RNA copies the genetic information of a gene by transcription [24]. RNA is a molecule formed from a chain of bases similar to DNA except that the base T is replaced by U (Uracil). Unlike DNA, RNA molecules are formed from a single strand which folds in on itself to form a complex structure. When an RNA molecule is transcribed from DNA, it is synthesized as a chain of bases complementary to the DNA sequence template (A$\rightarrow$U, T$\rightarrow$A, G$\rightarrow$C and C$\rightarrow$G).

In some cases, the RNA molecule itself is the final gene product, after some modifications, but more often the RNA is an intermediate for a protein product. The functions of genes are implemented via proteins which are linear polymers composed of 20 different types of amino acids. Proteins play a central role in virtually all aspects of cell structure and function. The sequence and function of a protein is defined by the sequence of a corresponding gene in nature, while the expression strength, place, and

time of the protein are regulated by a set of other genes.

Enzymes are proteins which catalyze chemical reactions. Proteins called transcription factors bind with DNA to regulate the transcription of genes. Other proteins are the component parts of large complexes which act as molecular machines.

The genetic information between genes and proteins are linked by mRNA (messenger Ribonucleic Acid). Protein-coding genes are transcribed to form the messenger RNA (mRNA). The mRNA is transported out of the cell nucleus into the cytoplasm of the cell. In the cytoplasm, the mRNA is bound by ribosomes which read the mRNA in triplet codons (nucleotide sequence) during translation. Transfer RNA (tRNA) is brought into the ribosome-mRNA complex, and matches the codon in the mRNA to the anti-codon in the tRNA, hence adding the correct amino acid in the sequence encoded by the gene. The mapping from nucleotide triplets to amino acids is known as the genetic code. The amino acids are linked into a growing peptide chain, and begin to fold into the correct formation. The folding continues until the protein chains are released from the ribosome as a mature protein.

Another important process is the DNA replication where the DNA copies itself (replicates) when cells divide so that each daughter cell has a copy of the same DNA as the parent.

2.3 Gene Expression and Regulation

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. Gene expression (activity) is the most fundamental level at which the genotype gives rise to the phenotype. The genetic code stored in the DNA is interpreted by gene expression, and the properties of the expression give rise to the organism's phenotype such as shape and color. Such phenotypes are often expressed by the synthesis of proteins that control the organism's shape, or that act as enzymes catalyzing specific metabolic pathways characterizing the organism.

Nearly every cell in a multi-cellular organism contains its complete genome. However, expression (activity) of genes in cells with different function within a multi-cellular organism is normally not the same. The function and differentiation of a cell could be explained by the expression levels of the genes. The expression level of a gene in a cell at a certain point in time is the amount of transcribed RNA encoded by the gene at that time point.

There are a wide range of mechanisms that are used by cells to increase or decrease the production of specific gene products (protein or RNA), and this process is called gene regulation. The expression of a gene is controlled by other genes. Gene expression is regulated both temporally and spatially [25]. 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. A gene has different expression patterns at different times. For example, the expression patterns of the zebrafish globin genes are different at different stages of the development [26]. There is also the spatial control of gene expression. Although cells from the same organism have identical genomes, cells in different parts of an organism may have different gene expression patterns due to the various functions they fulfill. Therefore, the regulation of gene expression is an essential part of life. There are two types of regulations: up-regulation, and down-regulation. Up-regulation is a process that occurs within a cell triggered by a signal originating internal or external to the cell, which results in increased expression of one or more genes and the resultant proteins encoded by those genes. On the converse, down-regulation is a process resulting in decreased gene and corresponding protein expression.

2.4 Microarrays

Monitoring the expression levels of all the genes in the genome of an organism can be done by microarrays. By using DNA microarrays, researchers are now able to measure the abundance of thousands of mRNA targets simultaneously [27, 28], providing a genomic viewpoint of gene expression. The amount of mRNA can be used

to understand about the activity of certain genes in certain circumstances. Although less precise than traditional low-throughput methods such as Northern blot and real-time PCR, the information gained from measuring the expression of thousands of genes simultaneously is considered significant, particularly in exploratory phases of research. More recently, high-throughput next generation sequencing is replacing microarrays as the method of choice for mRNA quantification and many other applications.

Microarray technology is based on DNA hybridization in which a DNA strand binds to its unique complementary strand. A set of known sequences called probes are fixed to a solid surface, and are placed in interaction with a set of fluorescently tagged unknown sequences called targets. Most microarray types use probes consisting of single-stranded DNA sequences, either derived from mRNAs via reverse transcription, or synthesized based on known mRNA sequences. After hybridization, the fluorescently lit spots indicate the identity of the targets, and the intensity of the fluorescence signal is in correlation with the quantitative amount of each target. Typically, green is used to label the reference samples, representing the baseline level of expression, and red is used to label the target sample in which the cells were treated with some condition of interest. Due to biological variation and a multi-step experimental protocol, these data are very noisy.

A genome wide measurement of transcription is called an expression profile, and provides us with a complete list of genes whose transcription levels are affected by the condition. From a biological viewpoint, what is measured is how the gene expression of each gene changes to perform complex coordinated tasks in adaptation to a changing environment.

The microarray technology propelled functional genomics, a discipline that strives to identify the role of genes in cellular processes, into the spotlight because it allowed functional analysis of genome-wide differential RNA expression between different samples, states, and cell types to gain insights into molecular mechanisms that regulate cell fate, development, and disease progression. Microarray data is used to generate a profile of gene expression, which serves as a determinant of protein levels, and therefore

cellular function between biological samples. A single experiment can provide information on the expression of thousands of genes, virtually the entire human genome, to compare expression patterns between any two states. Microarray experiments can indicate which genes are up or down regulated between samples from normal and diseased tissue, or two samples in absence and presence or a certain stimuli. It is easy to see why this technology might be appealing for understanding complex biological systems as well as drug discovery, disease diagnosis, and novel gene identification.

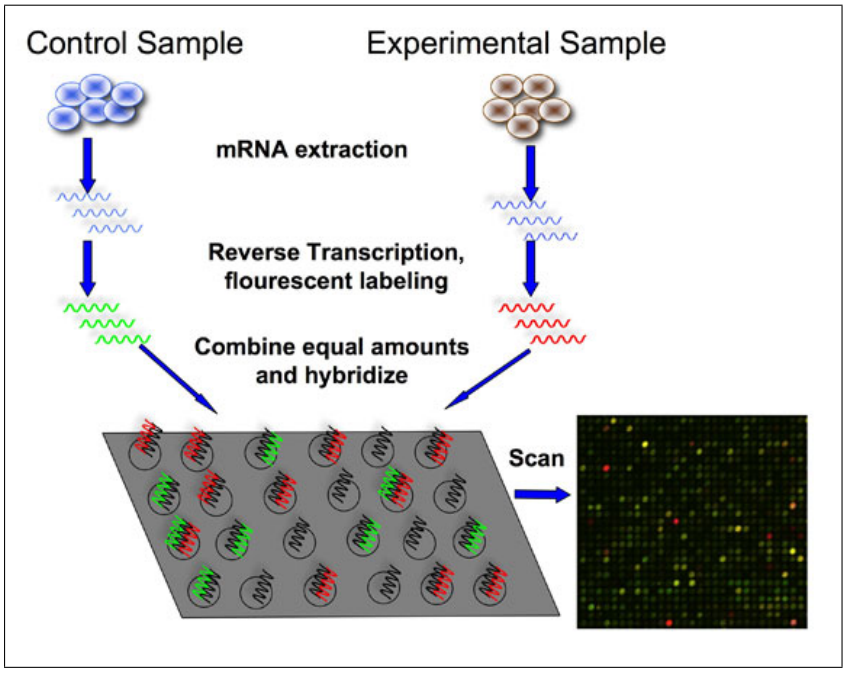

Figure 2.2 An image of a microarray. Each spot represents a different gene.

There are two common microarray platforms for investigating gene expression: complementary DNA (cDNA) [27] and oligonucleotide microarrays [28]. These platforms differ in experimental protocols, lengths of probes, and number of tissues measured per array implying challenges in the integration and comparison of data sets from different platforms.

The oligonucleotides are synthesized directly onto the surface using a combination of semiconductor-based photolithography and light-directed chemical synthesis. One of the main proponents of oligonucleotide arrays approach is Affymetrix, whose GeneChip arrays consist of small glass plates with thousands of oligonucleotide DNA probes, which are short stretches of nucleotides, 25-mers in Affymetrix' case, attached

to their surface. Very large numbers of mRNAs can be probed at the same time. However, manufacturing and reading the chips require expensive equipment. Current chips have over 650,000 different probes, with several probes and controls for each mRNA.

cDNA Microarrays provides a simpler solution to mRNA measurement. The microarrays are glass slides on which full-length cDNAs have been deposited by high-speed robotic printing. They are cheaper to manufacture and easy to read, but require handling a large number of cDNAs, which makes them less scalable.

Microarray measurements are carried out as differential hybridizations to minimize errors that originate from cDNA spotting variability. mRNA samples from two different sources, such as control and drug-treated cases, labeled with two different fluorescent dyes, are passed over the array at the same time. The fluorescence signal from each mRNA population is evaluated independently, and then used to calculate the expression ratio.

There are two main types of gene expression microarray data: static and time series microarray data. In static expression experiments, a snapshot of the expression of genes in different samples is measured while in the time series expression experiments, a temporal process is measured.

The scans of a microarray have to be transformed into values representing the gene expression rates in order to make quantitative analysis possible. The scans usually contain a lot of noise, and specialized image processing methods are used to reduce this. Background estimation and finding the optimal spot regions are a few examples. The resulting intensity values are transformed into well distributed gene expression values, by logarithmic ($\log_2$) transformation for example, making statistical analysis easier. The data has to be normalized to correct for systematical differences between the conditions in which the microarrays are hybridized. The result is a data matrix representing the relative gene expression values associating each gene (row) and condition (column).

2.5 Gene Networks

The complexity of a living cell is due to the cooperative activity of many genes and their products. This activity is often coordinated by the organization of the genome into regulatory modules, or sets of co-regulated genes that share a common function [29]. Cells collect vast amounts of information about the environment, process the information, and make complex decisions about how to respond and direct new phenotypes. The functions that enable these sophisticated behaviors are programmed by networks, and assemblies of interacting genes and proteins. The global gene expression pattern is therefore the result of the collective behavior of individual regulatory pathways. Thus, understanding a gene network as a whole is essential, and learning gene networks is an important central topic in the post genomic research [30, 31]. A widely used method to represent gene regulation is to draw network diagrams where genes connect to other genes as if they directly affect each other. Such gene networks are phenomenological models because they do not represent explicitly the proteins and metabolites that mediate those interactions. A gene network is then a projection of the whole biochemical network onto a space where the only observables are gene transcripts (mRNA), but where the influence of the remaining biochemical system is felt implicitly [32].

In systems biology, gene networks are of considerable interest. In general, a gene network is a graph in which vertices correspond to genes or gene products, and edges correspond to molecular interactions. Hence, a gene network represents a map of causal molecular interactions which in turn may allow us to elucidate the organism's observable characteristics. The interactions between genes and gene products can be represented as gene networks, or more particularly as transcriptional regulatory, signaling, metabolic, or protein networks. Networks in systems biology serve several purposes: they are capable of representing complex interrelations among genes or other components of a biological system; networks form a mathematical representation, which can be interpreted as a model; a network represents a data structure that can be utilized in data analysis to extract biological information by application of computational and statistical methods.

There are several applications and advantages to studying gene networks [32, 30] as follows. Gene networks provide a large-scale, coarse-grained view of the physiological state of an organism at the mRNA level. Gene networks describe a large number of interactions in a succinct way. They also present the dynamic properties of the gene regulatory system. It is an important step to uncover the complete biochemical network of the cell. Knowledge about gene networks might provide valuable clues for the therapeutics of complex diseases. As most phenotypes are the result of the collective response of a group of genes, gene networks help explain how complex traits arise, and which groups of genes are responsible for them. Gene networks are well suited for comparative genomics. Comparing gene networks from different genomes helps with the understanding of evolution.

A protein synthesized from a gene can serve as a transcription factor (TF) for another gene, as an enzyme catalyzing a metabolic reaction, or as a component of a signal transduction pathway. Apart from DNA transcription regulation, gene expression may be controlled during RNA processing and transport, RNA translation, and the post-translational modification of proteins. Therefore, gene regulatory networks (GRNs) involve interactions between DNA, RNA, proteins and other molecules. A comprehensive way to understand this complexity may consist of using functional association network (FAN) models. In these networks, the edges of the corresponding graph do not represent chemical interactions, but functional influences of one gene on the other. Specifically, networks for DNA transcription regulation via TFs are called Transcriptional regulatory (TR) networks. They are directed graphs, and vertices may correspond either to transcription units together with their protein products or to the regulated genes. Edges in these networks represent the TR interactions imposed by transcription factors (TFs).

In order to understand the function of a cell's metabolism, a special kind of a biological network can be defined from a collection of biochemical reactions. Such networks are called metabolic networks. A metabolic network consists of nodes corresponding either to enzymes or metabolites, and an edge indicates that the two nodes are found in the same biochemical reaction.

Protein-Protein Interaction (PPI) Networks, also called protein networks, represent the binding among proteins by, for example, forming complexes. The first approach to obtain such networks was based on two-hybrid studies. Recently, high-throughput technologies using affinity purification techniques followed by mass spectrometry have been employed to infer genome-scale protein interactions. They are undirected networks.

2.6 Gene Network Learning Techniques

The success of genome sequencing projects has led to the identification of almost all the genes responsible for the biological complexity of several organisms. The next important task is to assign a function to each of these genes. Genes and gene products do not work in isolation; rather, they are connected in highly structured networks of information flow through the cell. The learning of such gene networks from scratch using computational and statistical methods, and experimental data has been an important research topic.

The biological meaning of a network component depends on the type of data analyzed. Mostly, the network components are genes, since the primary data for inference is microarray data, and the network is a gene regulatory network. However, the methods are general, and can also be applied to protein data. In probabilistic models, we treat each component of the network as a random variable. The dataset consists of measurements that represent realizations of the random variables. Network components are identified with nodes in a graph. The goal is to find an edge set representing the dependency structure of the network components. We call the graph $G = (V, E)$ the structure of the cellular network. Depending on the model, G can be directed or undirected, cyclic or acyclic. If the graph has directed edges and no cycles, G is called a directed acyclic graph (DAG).

Biological processes result from the concerted action of interacting molecules. This general observation suggests a simple idea, which has already motivated the first

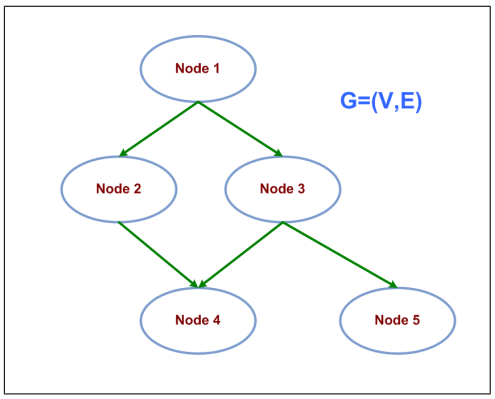

Figure 2.3 Structure of a graph. A graph is made up of vertices or nodes and lines called edges that connect them.

approaches to clustering expression profiles, and is still widely used in functional genomics: if two genes show similar expression profiles, they are likely to follow the same regulatory regime. Namely, co-expression suggests co-regulation. Co-expression networks are constructed by computing a similarity score for each pair of genes. If the similarity score is above a certain threshold, the gene pair gets connected in the graph, if not, it remains unconnected. Several similarity measures have been proposed, the most simple of which is correlation.

Stuart *et al.* [33] suggest that networks of co-expressed genes provide a widely applicable framework to elucidate gene function on a global scale. They identified pairs of genes that are co-expressed over 3000 DNA microarrays from humans, flies, worms, and yeast and found over 22,000 such co-expression relationships, each of which has been conserved across evolution. They argue that such conservation implies that genes are functionally related. Many of these relationships provide strong evidence for the involvement of new genes in core biological functions such as the cell cycle and protein expression. They experimentally confirmed the predictions implied by some of these links, and identified cell proliferation functions for several genes. By assembling these links into a gene-co-expression network, they found several components that were animal-specific as well as interrelationships between newly evolved and ancient modules.

Even high similarity of gene expression tells us little about the underlying bi-

ological mechanisms. Co-expression networks include regulatory relationships, but we cannot distinguish direct dependencies from indirect ones based on the similarity of expression patterns. To this end, use of Gaussian Graphical Models (GGM) is proposed. A GGM is an undirected graph on vertices. Each vertex (i.e. node) corresponds with a random variable. The edge set of a GGM is defined by non-zero partial correlations.

To estimate a GGM from data, we need to know which elements of the precision matrix are zero. Precision matrix shows correlation after correcting for the influence of all other genes. Full conditional relationships can only be accurately estimated if the number of samples is relatively large compared to the number of variables (number of genes). However, the number of genes to be analyzed almost always exceeds the number of distinct expression measurements in genomic applications of graphical models. Therefore, one must either improve the estimators of partial correlations, or resort to a simpler model. The basic idea in all of these approaches is that biological data are high-dimensional but sparse in the sense that only a small number of genes will regulate one specific gene of interest.

Full conditional independence models are closely related to a class of graphical models called dependency networks [34]. Dependency networks are built using sparse regression models to regress each gene onto the remaining genes. The genes, which predict the state of a certain gene, are connected to it by directed edges in the graph. In general, dependency networks may be inconsistent, i.e. the local regression models may not consistently specify a joint distribution over all genes. Thus, the resulting model is only an approximation of the true full conditional model. Still, dependency networks are widely used because of their flexibility and the computational advantage compared to structure learning in full conditional independence models.

Bayesian Networks model each variable with a conditional probability function dependent on a subset of other variables. Their stochastic nature makes them excellent candidates for modeling gene regulation systems where stochastic effects and data sets with large amounts of noise are expected. The logical next step is to ask for independencies of all orders. In the resulting graph, two vertices are connected if

no subset of the other variables can explain the correlation. This includes testing marginal, first order, and full conditional independencies. The graph encoding the above independence statements for all pairs of nodes is still undirected. It can be shown that knowing independences of all orders gives an even higher resolved representation of the correlation structure. The collection of independence statements already implies directions of some of the edges in the graph [35, 36]. The resulting directed probabilistic model is called a Bayesian network.

3. THEORY of BAYESIAN NETWORKS

A Bayesian Network (BN) is a compact graphical representation of the joint probability distribution over a set of random variables. The graph, G, of a BN consists of a set of N nodes (variables), $X_1, ..., X_N$, and a set of directed edges between these nodes. If there is a directed edge pointing from node X_i to node X_j, symbolically $X_i \rightarrow X_j$, then X_i is called a parent (node) of X_j, and X_j is called a child (node) of X_i. The graph structure of a static Bayesian network is defined to be a directed acyclic graph (DAG), that is, a directed graph in which no node can be its own descendant. Bayesian networks are the representation of conditional independency assumptions among variables, using directed graphs. In a Bayesian network context, the DAG is named as the structure, and the values in the conditional probability distributions are called the parameters. A directed graph consists of nodes, each representing a random variable, and directed edges representing dependencies among the variables (nodes). More generally, for random variables $X_1, ..., X_N$, the directed graph, G, implies a set of conditional independency relations. Conditional on the graph, the distributional form of the joint probability distribution, $P(X_1, ..., X_N|G)$, has to be chosen such that these stochastic independencies, encoded in the graph topology, G, are conserved in the probabilistic model. The probabilistic model has to be chosen such that an observed sample of realizations of the variables, $X_1, ..., X_N$, can be explained best by graphs that encode the true independencies among the variables.

Following the Bayes' theorem, the posterior probability of a graph, G, given the data, $\boldsymbol{D}$, is defined as follows:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{3.1}$$

where $P(D|G)$ is the marginal likelihood, and $P(G)$ is the prior probability of the graph, G. The posterior probability, $P(G|D)$, quantifies how much a graph is explained by the observed data, $\boldsymbol{D}$. The probability $P(D)$ in Eq. 3.1 serves as a normalization constant

that the expression $P(D)$ does not depend on the graph, and is defined as follows

$$P(D) = \sum_{G^*} P(D|G^*)P(G^*) \tag{3.2}$$

where the sum is over all valid directed graphs.

This definition of $P(D)$ ensures that Eq. 3.1 is a probability distribution over graphs; in particular Eq. 3.2 ensures the normalization:

$$\sum_{G^*} P(G^*|D) = 1 \tag{3.3}$$

The marginal likelihood, $P(D|G)$, quantifies how likely the observed data are conditional on the graph, G. Assuming that the true independencies among the variables are actually inferable from the data, $\boldsymbol{D}$, high marginal likelihoods can only be reached by those graphs that imply, or approximate these true relationships. The graph prior distribution, $P(G)$, is used as a weighting factor for each graph. These weights do not depend on the data, D, and can be used to include external knowledge, which may be available from previous studies or other external sources. The greater the product of the marginal likelihood, $P(D|G)$, and the graph prior probability, $P(G)$, the more plausible (likely) is the graph, G, from a Bayesian perspective.

Recalling that $P(D)$ is a normalization constant, it can be seen from Eq. 3.1 that the posterior probability of a graph, $P(G|D)$, is proportional to the marginal likelihood, $P(D|G)$, times the graph prior distribution, $P(G)$:

$$P(G|D) \propto P(D|G)P(G) \tag{3.4}$$

The graph prior distribution, $P(G)$, can be used to incorporate biological prior knowledge, or in the absence of real prior knowledge about the regulatory network, G, a uniform distribution may be assumed for $P(G)$.

If the data set consists of independent realizations of the variables, static Bayesian network methodology is applied, and all valid graphs have to be directed and acyclic.

If the variables have been measured over time, dynamic Bayesian network (DBN) methodology is required, and all directed graphs are valid, independently of whether they are acyclic or not.

3.1 Static Bayesian Network

A static Bayesian network is a graphical representation of the independency structure between the components of a random vector X, where $X = (X_1, ..., X_N)$, and n is the number of network components. The individual random variables are associated with the vertices (i.e. nodes) of a directed acyclic graph (DAG) G, which describes the dependency structure. Each node is described by a local probability distribution and the joint distribution P(X) over all nodes $X_1, ..., X_n$ factors as

$$P(X) = \prod_i P(X_i|Pa(X_i), \theta_i) \tag{3.5}$$

where θ_i denotes the parameterization of the local distribution, and $Pa(X_i)$ is the vector of parent states denoting the activity levels of a gene's regulators. The DAG structure implies an ordering of the variables known as the Markov condition that a node is conditionally independent of all its non-descendants given its parents. The factorization of the joint distribution is the key property of Bayesian networks.

3.2 Local Probability Distributions

Bayesian network models differ with respect to assumptions about the local probability distributions $P(X_i|Pa(X_i), \theta_i)$ attached to each node $v \in V$. There are two types of parametric local probability distributions used in practice, which are multinomial distributions for discrete nodes and Gaussian distributions (normal distributions) for continuous nodes. A discrete node with discrete parents follows a multinomial distribution parameterized by a set of probability vectors, one for each parent configuration. A continuous node with continuous parents follows a Gaussian distribution,

where the mean is a linear combination of parent states.

3.3 Conditional Independence in Directed Graphs

The way to read off the independence statements from Bayesian networks is given by the definition of d-separation [35]. The three archetypical situations of d-separation (chain, fork, and collider) can be seen in Figure 3.1. In a chain X $\rightarrow$ Y $\rightarrow$ Z, the middle node Y blocks the information flow between X and Z and thus it holds that X is conditionally independent of Z given Y (X $\perp$ Z | Y). In a fork, where X and Z are both regulated by Y, knowing the state of the regulator renders the regulatees conditionally independent and thus again X $\perp$ Z | Y. The last case is more surprising: if X and Z are independent regulators with a common target Y, then the state of Y gives us information about X and Z. For example, imagine that Y is only expressed if only one of its regulators is active, then seeing Y expressed and X active implies Z being inactive. Thus, in the collider X $\rightarrow$ Y $\leftarrow$ Z the middle node Y unblocks the path between X and Z and thus X $\not\perp$ Z | Y.

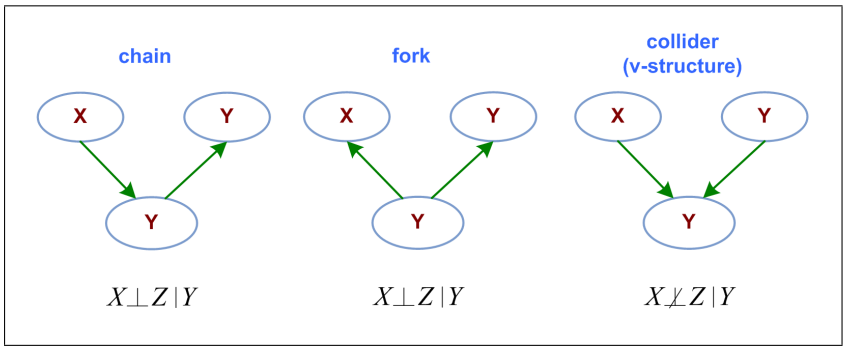

Figure 3.1 Conditional independence in directed graphs.

3.4 Markov Equivalence

Many Bayesian networks may encode the same conditional independencies, and they are called Markov equivalent. The set of all DAGs can be partitioned into Markov equivalence classes. Each class can be represented by a PDAG (partially directed

acyclic graph) called an essential graph or pattern. That is, all equivalent networks share the same underlying graph skeleton but may differ in the direction of edges that are not part of a collider (also called a v-structure) [37]. Markov equivalence poses a theoretical limit on structure learning from data: even with infinitely many samples, we cannot resolve the structures in an equivalence class. In biological terms this means that even if we find two genes to be related it may not be clear which one is the regulator and which one is the regulatee.

3.5 Dynamic Bayesian Networks

Different from static Bayesian networks, Dynamic Bayesian networks are used to model temporal relationships among variables. Namely, a Bayesian network only represents the probabilistic relationships among a set of variables at some point in time. It does not represent how the value of some variable may be related to its value, and the values of other variables at previous points in time. In many problems, however, the ability to model temporal relationships is very important. More formally, Dynamic Bayesian networks (DBNs) can be applied if the random vector, $X = (X_1, ..., X_N)^T$ with T time slices and N number of nodes, has been measured over time,. All interactions between nodes are then subject to a time delay, τ , where τ is called the order of the DBN model. An edge from X_j to X_n, symbolically $X_j \rightarrow X_n$, in a first order DBN, $\tau = 1$, indicates that the realization of X_n at time point t is conditionally dependent on the realization of X_j at time point $t - 1$.

As for static Bayesian networks π_n denotes the parent set of $X_n (n = 1, ..., N)$, and there is a one-to-one mapping between the graph, G, and the system of parent sets, $\pi_1, ..., \pi_N$. Because of the time delay, τ, of interactions, DBNs are based on a bipartite graph structure between two time steps t and $t + 1 (t = 2, ..., m)$ so that the acyclicity constraint, which is fundamental for the factorization in static Bayesian networks, is guaranteed to be satisfied. An example of The bipartite graph structure of DBNs is illustrated graphically in Figure 3.2, where the prior and transition network are shown. It can be seen that while the prior network is simply a general Bayesian network, the

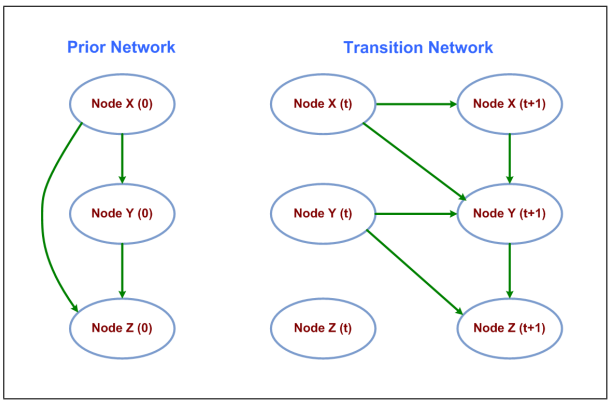

Figure 3.2 The bipartite graph structure of DBNs.

transition network has a slightly different structure to it. In this, there are two layers of nodes, and arcs from the first layer only go to the second. In addition, no arcs go from the second layer to the first. For the purposes of performing inference, or simply reasoning about them, DBNs can be expanded out into a single network.

Murphy *et al.* [38] provide an overview of different DBN variants and their learning algorithms, and how these relate to various gene expression models. In particular, they point out the similarity of learning DBNs with discrete variables and unknown structure, to Boolean network reverse engineering algorithms such as REVEAL [39].

3.6 Score Based Structure Learning

There are two very different approaches to structure learning: constraint-based, and search-and-score. In the constraint-based approach, we start with a fully connected graph, and remove edges if certain conditional independencies are measured in the data. This has the disadvantage that repeated independence tests lose statistical power [40].

In the more popular search-and-score approach, we perform a search through the space of possible DAGs, and either return a point estimate, the best DAG found, or return a set of the models found, an approximation to the Bayesian posterior. Learning the structure of a Bayesian network can be considered a specific example of the

general problem of selecting a probabilistic model that explains a given set of data. In computational systems biology applications, the network structure is mostly learnt by score based techniques [35, 36]. In the following, we review the maximum likelihood scores and the Bayesian scores that evaluate the model fit to data. Once the score is defined, model selection is posed as an optimization problem over the discrete space of possible model structures. Additional topics include avoiding overfitting and encoding prior information.

3.6.1 Maximum Likelihood

A straightforward idea for model selection is to choose the DAG G, which allows the best fit to the data $\boldsymbol{D}$. The best fit for a given DAG G is determined by maximizing the likelihood$P(D|G,\theta)$ as a function of θ, the parameters of the local probability distributions. A score for DAG G is then given by

$$score_{ML}(G) = \max_{\theta} P(D|G,\theta) \tag{3.6}$$

Unfortunately, the likelihood is not an appropriate score to decide between models since it tends to overfit the data. Richer models with more edges will have a better likelihood than simpler ones, since the additional parameters allow a better fit to the data. A standard solution to this problem is to penalize the maximum likelihood (ML) score according to the model complexity. An often used example of this general strategy is scoring with the Bayesian information criterion.

3.6.2 Bayesian Information Criterion (BIC)

The BIC score [41] is a regularized maximum likelihood estimate, which controls overfitting by penalizing the maximal likelihood of the model with respect to the number of model parameters. It is defined as

$$score_{BIC}(G) = \max_{\theta} P(D|G,\theta) - \frac{d}{2}\log N \tag{3.7}$$

where d is the number of parameters and the factor $\log N$ scales the penalty with respect to the likelihood. The BIC score can also be used to learn Bayesian networks with missing values or hidden variables. The likelihood has then to be maximized via the Expectation-Maximization (EM) algorithm.

3.6.3 Bayesian Scores

In most cases a full Bayesian approach is preferred over ML or BIC. In Bayesian structure learning we evaluate the posterior probability of the model topology G given the data $\boldsymbol{D}$ as:

$$score_{Bayes}(G) = P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{3.8}$$

The denominator $P(D)$ is an average of data likelihoods over all possible models. This normalizing constant is the same for all models, and thus we do not need compute it to decide between competing models. The two main terms to consider in the Bayesian score are the prior over model structures, $P(G)$, and the marginal likelihood $P(D|G)$.

3.6.4 Marginal Likelihood

The marginal likelihood $P(D|G)$ is the key component of Bayesian scoring metrics. It equals the full model likelihood averaged over parameters of local probability distributions (LPD), that is,

$$\int_{\Theta} P(D|G,\theta)P(\theta|G)d\Theta \tag{3.9}$$

Marginalization is the reason why the LPD parameters θ do not enter the definition of the posterior above; they have been integrated out. It is important to note that the LPD parameters were not maximized as it would be done in a maximum likelihood estimate or in a BIC score. Averaging instead of maximizing prevents the Bayesian score from overfitting. Computation of the marginal likelihood depends on the choice

of local probability distributions and local priors in the Bayesian network model. To compute the marginal likelihood analytically, the prior $P(\theta|G)$ must fit to the likelihood $P(D|G,\theta)$. Statistically, this fit is called "conjugacy". A prior distribution is called conjugate to a likelihood, if the posterior is of the same distributional form as the prior [42]. If no conjugate prior is available, the marginal likelihood has to be approximated.

The marginal likelihood for discrete Bayesian networks was first computed by Cooper *et al.* [43], and is further discussed by Heckerman *et al.* [44]. The conjugate prior for the multinomial distribution is the Dirichlet prior [42]. Assuming independence of the prior for each node and each parent configuration, the score decomposes into independent contributions for each family of nodes. Corresponding results exist for Gaussian networks using a Normal-Wishart prior [45]. The marginal likelihood again decomposes into node-wise contributions. Conjugate analysis and analytic results are possible using normal-gamma priors for each leaf node [46, 47].

For a given BN model, the probability of observing data is [19]:

$$P(D|G) = \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij}+M_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk}+s_{ijk})}{\Gamma(a_{ijk})} \tag{3.10}$$

where N is the number of nodes, q_i is the number of different states of node's parents, and r_i is the set of values a node can take on. N_{ij} is the sum of corresponding Dirichlet distribution hyper-parameters a_{ijk}. M_{ij} is the number of times that the parents of node i take on configuration j in the dataset. Of these M_{ij} cases, s_{ijk} is the total number of times in the sample that node i is observed to have value k when its parents take on configuration j. The equation above is used as a score metric and named the Bayesian Scoring Criterion (BSC) [44].

Given the gamma function $\Gamma(x) = \int_0^\infty t^{x-1}e^{-x}\,dt = (x-1)!$ note that $\frac{\Gamma(x+1)}{\Gamma(x)} = x$. The beta density function(ρ) with parameters $a, b \in \Re^+$ and $N = a + b$ is defined as

$$\rho(f) = \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)} f^{a-1}(1-f)^{b-1}, 0 \le f \le 1 \tag{3.11}$$

We refer to this function as $\beta(f; a, b)$, and a random variable F with this density

function is said to have a beta distribution. One can show that

$$\int_0^1 f^a(1-f)^b df = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)} \tag{3.12}$$

which yields $\mathrm{E}[\beta(f;a,b)] = \frac{a}{N}$. Suppose F has a beta distribution $\beta(f;a,b)$, and X is a random variable with two values (1 and 2) such that $P(X=1|f) = f$, then $P(X=1) = \mathrm{E}[f] = \frac{a}{N}$. In a BN setting (without loss of generality, consider a binary BN), we view F as the "driving function" for node X, assign a prior set of parameters (a,b) for X and update the count for a and b, if there is some observed data.

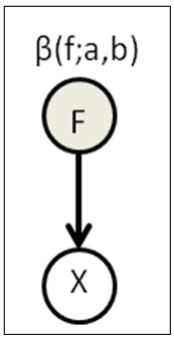

Figure 3.3 Single node binary Bayesian Network.

Consider the single node binary BN in Figure 3.3. Let's consider $X=1$ to be heads and $X=2$ to be tails. If initially say, $a=b=3$, then we say probability of observing a heads is $3/6 = 1/2$. Assume we use a biased coin, and one now observe data $d = 11112211111$ with 9 number of 1s and 2 number of 2s. Then the distribution function is updated as $a = 3+9 = 12$, and $b = 3+2 = 5$ and we say the probability of observing a heads is $12/17$. If one has a valid reason to bias the initial configuration of a and b, this can be reflected in the prior distribution definition. For the above example, one could choose a larger value for a compared to b, initially.

Now consider binomial data $d = (x_1, x_2, ..., x_M)$ with parameter F following $\beta(f;a,b)$ and $N = a+b$. Assume we have s number of 1s and t number of 2s in d. One now can calculate $P(d) = \int_0^1 P(d|f)\rho(f)d(f)$ and

$$P(d) = \frac{\Gamma(N)}{\Gamma(N+M)}\frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(a)\Gamma(b)} \tag{3.13}$$

For example, let $a=b=1$ and assume we observe $d=(1,2)$, i.e. one heads and one

tails. Then

$$P(d) = \frac{\Gamma(2)}{\Gamma(2+2)} \frac{\Gamma(1+1)\Gamma(1+1)}{\Gamma(1)\Gamma(1)} = \frac{1}{6} \tag{3.14}$$

Note that if a random variable X_2 in a BN has a parent X_1, then we would have two "driving functions" for X_2, one for each instance of X_1 (whether X_1 is 1 or 2). Similarly, if a node in a BN has 2 parents, we would have four beta functions, one for each configuration of the values the node's parents assumes $(11, 12, 21, 22)$, and so on.

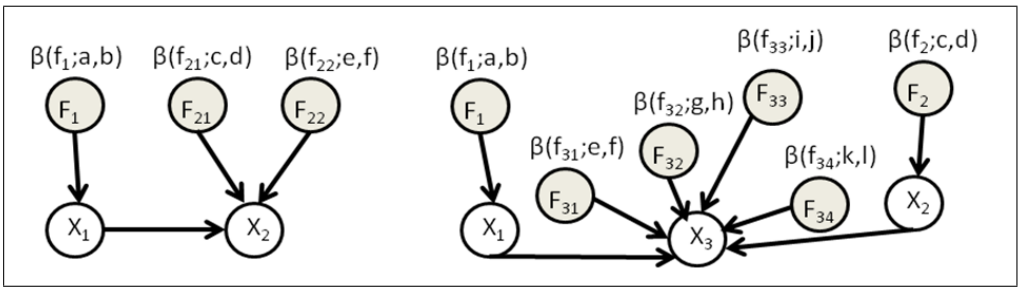

Figure 3.4 Driving functions of various parent configurations of BNs.

3.6.5 Bayesian Dirichlet Equivalent (BDe) Score

Bayesian Dirichlet equivalent (BDe) scoring scheme uses Dirichlet functions driving each node, which is a generalization of the beta distribution. Based on the definitions listed here, first, let's review the BDe score definition: Hyper-parameters a_{ijk} in Eq. 3.10 of BSC can be determined using the equivalent sample size method (i.e. sum of the initial Dirichlet parameters used at each node have the same total), in which case the score is called the Bayesian Dirichlet Equivalent (BDe) [19].

Parameters used in Eq. 3.10 are best explained by an example. Consider the BN in Figure 3.5, where nodes can take on values 1, 2, or 3 and follow Dirichlet distributions.

Note that X_3 has two parents and since each node can take on 3 values, X_3's parents can take on 9 (3x3) different configurations. Following equivalent sample size method, sum of the initial Dirichlet hyper-parameters, a_{ijk}, driving each node has the

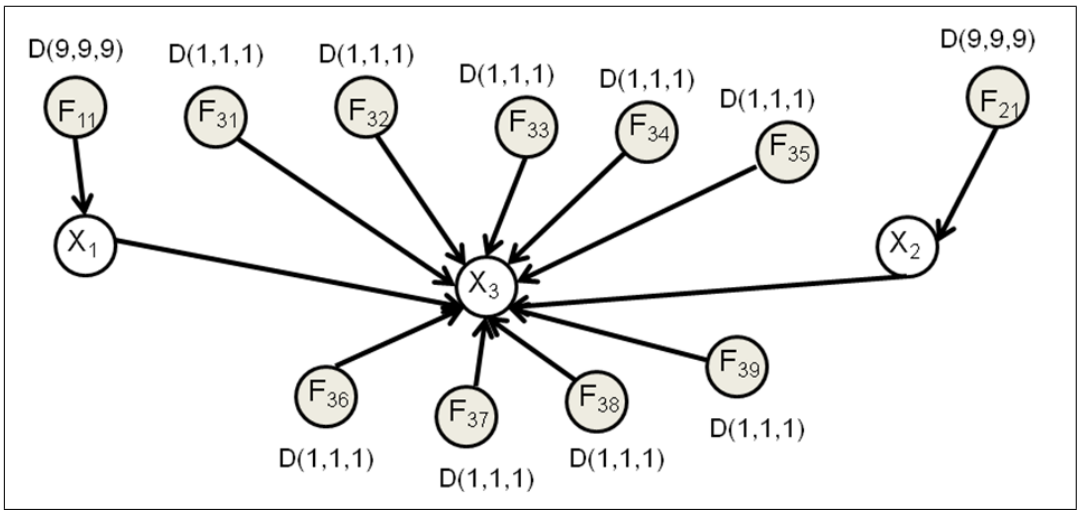

Figure 3.5 Configuration of a BN with nodes which take on values 1, 2, or 3.

same total, 27. Hence,

$$a_{111} = a_{112} = a_{113} = a_{211} = a_{212} = a_{213} = 3 \text{ , and}$$

$$a_{311} = a_{312} = a_{313} = 1(X_3\text{'s parents take on configuration1}, i.e. X_1 = 1, X_2 = 1)$$

$$a_{321} = a_{322} = a_{323} = 1(X_3\text{'s parents take on configuration2}, i.e. X_1 = 1, X_2 = 2)$$

$$a_{331} = a_{332} = a_{333} = 1(X_3\text{'s parents take on configuration3}, i.e. X_1 = 1, X_2 = 3)$$

$$a_{341} = a_{342} = a_{343} = 1(X_3\text{'s parents take on configuration4}, i.e. X_1 = 2, X_2 = 1)$$

$$a_{351} = a_{352} = a_{353} = 1(X_3\text{'s parents take on configuration5}, i.e. X_1 = 2, X_2 = 2)$$

$$a_{361} = a_{362} = a_{363} = 1(X_3\text{'s parents take on configuration6}, i.e. X_1 = 2, X_2 = 3)$$

$$a_{371} = a_{372} = a_{373} = 1(X_3\text{'s parents take on configuration7}, i.e. X_1 = 3, X_2 = 1)$$

$$a_{381} = a_{382} = a_{383} = 1(X_3\text{'s parents take on configuration8}, i.e. X_1 = 3, X_2 = 2)$$

$$a_{391} = a_{392} = a_{393} = 1(X_3\text{'s parents take on configuration9}, i.e. X_1 = 3, X_2 = 3)$$

Now, let's consider a sample input data and focus on node 3:

Observation	X_1	X_2	X_3
1	3	1	2
2	3	1	1
3	1	2	1
4	2	1	3
5	2	2	1
6	1	3	2
7	1	3	3
8	3	3	2
9	3	2	3
10	2	3	1

$$\text{Note that} N_{31} = N_{32} = N_{33} = N_{34} = N_{35} = N_{36} = N_{37} = N_{38} = N_{39} = 3$$

Considering observed data,

$$M_{31} = 0, M_{32} = 1, M_{33} = 2, M_{34} = 1, M_{35} = 1, M_{36} = 1, M_{37} = 2, M_{38} = 1, M_{39} = 1$$

That is, for example, $M_{37} = 2$ means 3rd node's (X_3's) parents assumed configuration 7 ($X_1 = 3, X_2 = 1$) in 2 instances. Now breaking these M_{ij} cases into s_{ijk}'s for node 3, we have:

$$s_{311} = 0 s_{312} = 0 s_{313} = 0$$
$$s_{321} = 1 s_{322} = 0 s_{323} = 0$$
$$s_{331} = 0 s_{332} = 1 s_{333} = 1$$
$$s_{341} = 0 s_{342} = 0 s_{343} = 1$$
$$s_{351} = 1 s_{352} = 0 s_{353} = 0$$
$$s_{361} = 1 s_{362} = 0 s_{363} = 0$$
$$s_{371} = 1 s_{372} = 1 s_{373} = 0$$
$$s_{381} = 0 s_{382} = 0 s_{383} = 1$$
$$s_{391} = 0 s_{392} = 1 s_{393} = 0$$

That is, for example, 7th row in this table means when X_3's parents assumed configuration 7, X_3 assumed the value "1" and "2" once ($s_{371} = 1, s_{372} = 1$), and the value "3" zero times ($s_{373} = 0$).

3.6.6 Model Selection and Assessment

To search for the DAG with the highest score is mathematically trivial: compute the score for every possible DAG and choose the one that achieves the highest value. What makes exhaustive search computationally infeasible in almost all applications is the huge number of DAGs. The number of DAGs on n edges is

$$G(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} G(n-k) \tag{3.15}$$

where n is the number of nodes [48]. The number of DAGs increases super-exponentially; the first few values are shown below:

n	G(n)
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503
7	1.1 x 10^9
8	7.8 x 10^{11}
9	1.2 x 10^{15}

The exhaustive search approach to structure learning is a method to enumerate all possible DAGs, and score each one. In practice, one can't enumerate all possible DAGs for $N > 5$ with the current computational power, but one can evaluate any reasonably-sized set of hypotheses in this way. Therefore, we must use heuristic strategies to find high-scoring Bayesian networks, without enumerating all possible DAGs.

3.6.7 Defining the Search Space

First, we need to decide how to describe the models of interest. This defines the model space, in which we search for models describing the data well. To apply search heuristics, we have to equip the search space with a neighborhood relation, that is, operators to move from one point of the search space to the next one. The most simple search space results from defining a neighborhood relation on the DAGs. Two DAGs are neighbors if they differ by one edge, which is either missing in one of them or directed the other way round. Madigan *et al.* [49] and Chickering *et al.* [47] restrict the search space to Markov equivalence classes of DAGs, which uniquely describes a joint distribution. Thus, no time is lost in evaluating DAG models that are equivalent anyway. Friedman *et al.*[20] search over orders of nodes rather than over network structures. They argue that the space of orders is smaller and more regular than the space of structures, and has a much smoother posterior landscape.

3.6.8 Search Heuristics

Score based algorithms assign a score to each candidate Bayesian network, and try to maximize it with some heuristic search algorithm. Greedy search algorithms are a common choice, but almost any kind of search procedure can be used. Score based algorithms on the other hand are simply applications of various general purpose heuristic search algorithms, such as hill climbing, TABU search, simulated annealing and various genetic algorithms. Most of the search algorithms can be applied to all search spaces, even though they are usually applied to DAGs. They return a single best network. A simple and fast, but still powerful method is the hill-climbing algorithm. First, a point in the search space is chosen to start from, e.g. a random graph or the empty graph. The posterior probability for all graphs in the neighborhood of the current graph are computed, and the graph with highest score is selected. This iteration is repeated until no graph in the neighborhood has a larger score than the current graph. This procedure finds a local maximum of the Bayesian scoring metric. Various other optimization techniques, such as iterated hill-climbing try to overcome

this problem. Iterated hill climbing makes local search until a model with a local maximum score is found. The structure is randomly perturbed, and the process is repeat for some manageable number of iterations. The K2-algorithm [43] is a variant of the greedy search, which assumes that the order of nodes is known. Several approaches have been suggested to speed up the model search. The sparse candidate algorithm [50] restricts the number of possible parents for each node by searching for pairs of nodes which are correlated. The optimal reinsertion algorithm, introduced by Moore and Wong [51] is a search-and-score algorithm that works as follows: at each step, a target node is chosen; all edges entering or leaving the target are deleted; the optimal combination of in-edges and out-edges is found; the node is re-inserted with these edges. The optimal reinsertion may be combined with the sparse candidate method. Pena *et al.* [52] propose an algorithm in which Bayesian networks grow starting from a target gene of interest. Parents and children of the given target genes are iteratively added to the Bayesian network. The algorithm stops after a predefined number of steps and thus, intuitively, highlights the surrounding area of the seed gene without having to compute the complete Bayesian network over all genes. Friedman [53, 54] introduces the structural EM algorithm to learn Bayesian networks in the presence of missing values or hidden variables. It is an extension of the Expectation- Maximization (EM) algorithm that performs structure search inside the EM procedure, and shows improvements in terms of speed and accuracy.

4. BAYESIAN PATHWAY ANALYSIS

Most current approaches to high throughput biological data (HTBD) analysis either perform individual gene/protein analysis or, gene/protein set enrichment analysis for a list of biologically relevant molecules. Bayesian Networks (BNs) capture linear and nonlinear interactions, handle stochastic events accounting for noise, and focus on local interactions, which can be related to causal inference. Here we describe for the first time an algorithm, called Bayesian Pathway Analysis (BPA), that models biological pathways as BNs, and identifies pathways that best explain given HTBD by scoring fitness of each network.

The proposed method (BPA) considers the topology via which genes interact with each other when analyzing a group of genes [12]. Using pathway information from global databases, we model each biological pathway as a BN after merging repeating entries and, if necessary, solving for cyclicity while preserving the dependencies entailed by the original pathway. We consider the resulting BN, which is a graphical representation of gene interactions rendered by the given pathway, with non-informal, uniform belief priors. We quantify the degree to which observed experimental data fits this BN using Bayesian Dirichlet equivalent (BDe) score calculation where the BN is updated with input data during score calculation. We assess statistical significance for the score of each pathway by testing it against data sets generated by applying randomization via bootstrapping. Results are evaluated in forms of nominal p-values and False Discovery Rate (FDR) values correcting for multiple hypotheses testing. Overall workflow used in BPA is depicted in Figure 4.1.

Renal Cell Carcinoma (RCC) is the sixth leading cause of cancer deaths in the United States and has no established biomarker for early detection or follow-up [55, 56]. Most common histological subtypes of RCC are clear-cell RCC (cRCC) and papillary RCC (pRCC) generally related to deficiencies in von Hippel-Lindau, rapamycin complex 1 kinase (mTOR) and fumarate hydratase [57]. Treatment for metastatic

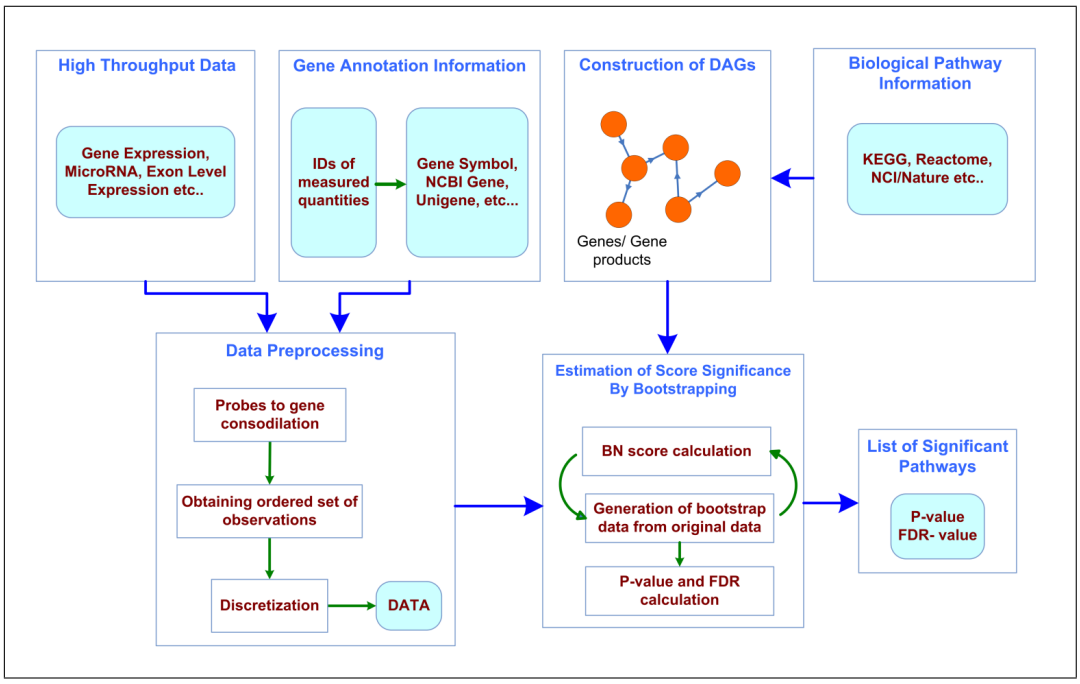

Figure 4.1 Layout of the BPA approach.

RCC includes rather unspecific application of cytokine therapies (e.g. interferon-alfa, interleukin-2) and more targeted use of receptor tyrosine kinase inhibitors (Sorafenib and Sunitinib), mTOR inhibitors (Everolimus and Temsirolimus), and monoclonal anti VEGF antibody therapy with Bevacizumab [57, 56]. These therapies offer acceptable response rates (30-40%) but are not beneficial in overall survival. Therefore a more detailed analysis of molecular pathways underlying RCC is essential. In previous studies, transcriptional profiling of various RCC subtypes were analyzed, and a predictive proteomic signature that distinguishes between interleukin-2 therapy responders and non-responders were obtained [58, 55]. Others have also used gene expression and proteomic approaches to gain insight into the molecular pathways governing RCC [59]-[60]. In addition to cRCC and pRCC, we used BPA to analyze the rare RCC subtype chromophobe RCC (chRCC) and other renal malignancies such as transitional cell cancers of the renal pelvis (TCC) and Wilms' tumors (WT) or benign renal tumors such as oncocytomas (OC).

4.1 Pathway Information Retrieval

Biochemical network data of pathways were retrieved from KEGG [61], NCI/Nature Pathway Interaction Database [62], Reactome [63], and HumanCyc [64] representing molecular interaction and reaction networks for metabolism, genetic information processing, environmental information processing, cellular processes, and diseases.

4.2 Construction of Directed Acyclic Graphs

A BN is a compact graphical representation of the joint probability distribution over a set of random variables in the form of a DAG where nodes represent random variables. The DAG encodes assertions of conditional independence, which are generally represented as a set of conditional probability tables (CPTs).

When modeling pathways as BNs, we first merge repeating entries, for example Smad2/3 is present at several locations in the KEGG TGFβ pathway, see Figure 4.2, as a single node in the DAG while conserving edge relations. Cyclic paths are eliminated using Spirtes' method [65]. In this procedure, graph representation of structural equation models (SEM) are converted to collapsed acyclic graphs such that d-separations in the collapsed graph entails the same independency relations defined by the model [35]. Cyclegroups (set of all cycles sharing at least one node) are found using Tarjan's algorithm [66]. For a given BN, all d-separations are conditional independencies, and every conditional independency implied by the BN is identified by d-separations [19]. Therefore, our way of solving cyclicity preserves distributional features explained by the pathway after it has been converted to a DAG.

4.3 Microarray Data Preprocessing and Discretization

BPA assumes normalized data as input. First, IDs used in the array platform corresponding to a given node in the pathway representation are pooled, and one representative signal value per node is calculated using the one-step Tukey's biweight algorithm [67]. Currently, BPA addresses experimental designs consisting of two groups of samples (e.g. cancer vs. normal) though generalization of this framework to

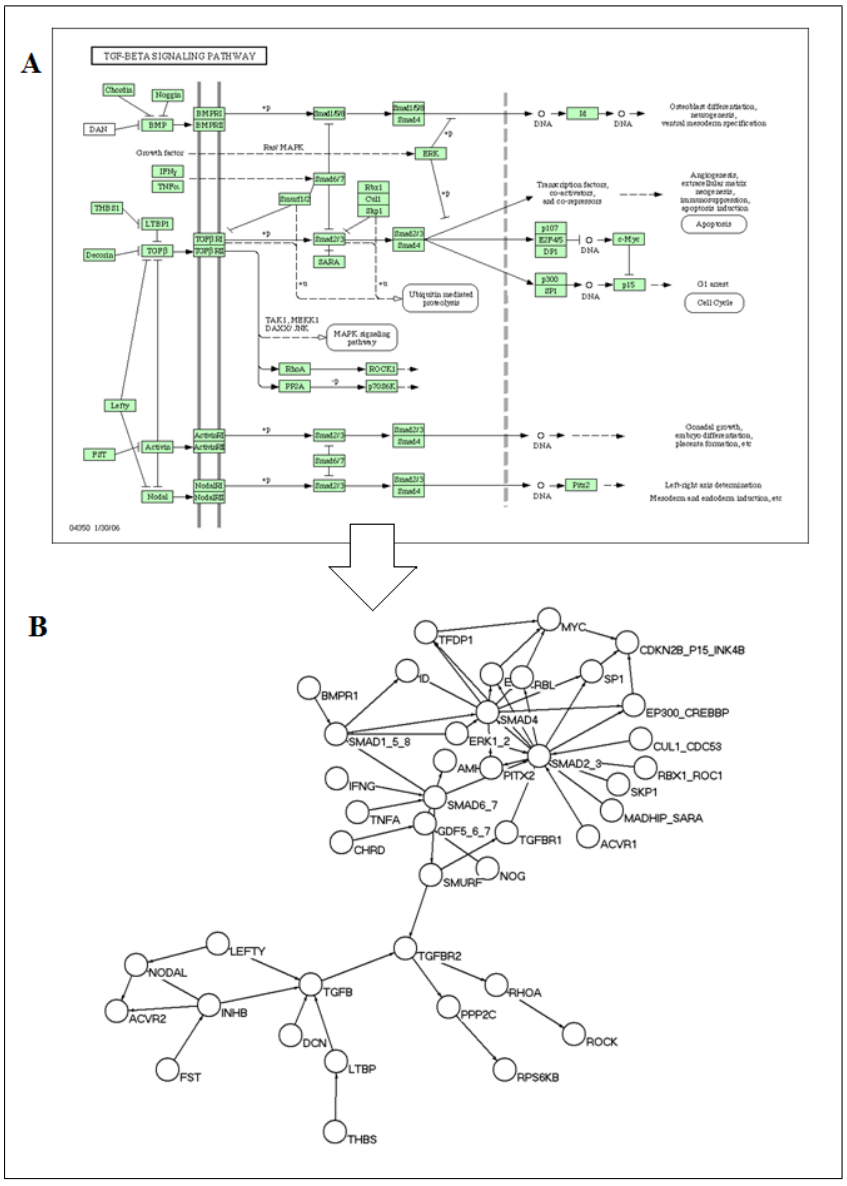

Figure 4.2 Construction of directed acyclic graphs. (A) TGF-β Signaling Pathway as retrieved from the KEGG database. (B) DAG produced from TGF-β pathway map retrieved from the KEGG database. Each node is identified by gene symbol (e.g. DCN for Decorin).

multiple groups is straightforward. For a given BN (converted from a pathway), we obtain observed fold changes for genes in this BN (pathway) by pairwise comparisons of samples in each group. This approach provides a distribution of fold change values and a reasonable data set size used to score each BN. Let SG_1 and SG_2 represent two groups of samples in the data set with C_1 and C_2 samples in each group, respectively. Let g_{i1j} and g_{i2k} be the expression values of the i^th node in the pathway in j^{th} and k^{th} samples in the sample groups SG_1 and SG_2, respectively, where $1 \leq j \leq C_1$ and $1 \leq k \leq C_2$. Let X_i, $1 \leq i \leq N$, represent the random variable for the i^{th} node in a BN with N nodes. An ordered set of observations, O, for the data set is obtained by pairwise comparison of all samples in sample groups SG_1 and SG_2. The l^{th} element of O, o_l, corresponds to comparison of j^{th} and k^{th} samples in the sample groups SG_1 and SG_2 such that $l = (j-1)xC2 + k$, where $1 \leq j \leq C_1$ and $1 \leq k \leq C_2$. Thus the cardinality of O is C_1xC_2. Each o_l is a vector with dimension equaling the number of nodes in the pathway, N, such that the i^{th} element of o_l, o_{li}, equals g_{i2k}/g_{i1j}, where j and k are related to l as described above. The data matrix $\boldsymbol{D}$, with elements d_{li} is obtained from O such that d_{li} equals 1 if $o_{li} < 0.5$ or $o_{li} > 2$ (i.e. a gene is dysregulated) and 2 otherwise. To this end, we have converted each pathway into a BN, where nodes of the BN represent nodes in the pathway, and node random variables are identified by discretized FC values. Nodes of BNs are assumed to follow Dirichlet distribution, and are initialized using the equivalent sample size method for prior beliefs [19]. The matrix $\boldsymbol{D}$, which consists of N columns and C_1xC_2 rows, is sequentially evaluated row by row, where each row is used to update the Dirichlet distribution parameters used at the nodes of BN. Upon conclusion of evaluation of $\boldsymbol{D}$, the score for the BN is calculated.

4.4 Bayesian Score Metric

Following discretization, the nodes in the BN model represent discrete random variables with a multinomial distribution. We use Bayesian Dirichlet equivalent (BDe) scoring scheme, which uses Dirichlet functions driving each node, which is a generalization of the beta distribution. The Dirichlet distribution is chosen as the conjugate prior of the multinomial distribution. Details of the Bayesian Dirichlet equivalent

(BDe) scoring scheme can be found in Chapter 2.

4.5 Estimation of Score Significance by Randomization via Bootstrapping

At this point, we have BNs converted from pathways, a data matrix $\boldsymbol{D}$ for each BN representing observed, discretized fold change values for genes (nodes) represented in the BN, and BDe score calculated for the BN using the observed $\boldsymbol{D}$ matrix. We assess statistical significance of the BDe score, S_n, calculated for n^{th} BN by using randomization via bootstrapping. We use a data generating process, and estimate a distribution of the score S given the null hypothesis that the scores are result of pure chance. For a one-tailed test with a rejection region in the upper tail, the bootstrap p-value for S_n, $P(S_n)$, is estimated by the proportion of randomized samples that yield a score greater than S_n. If we have $\boldsymbol{R}$ randomized data sets, then

$$P(S_n) = \frac{1}{R}\sum_{k=1}^{R} I(S_k > S_n) \tag{4.1}$$

where I is the indicator function yielding 1 if the Bayesian score is better than the original network score and 0 otherwise, and S_k is the score of the BN using k^{th} randomized data set. As $\boldsymbol{R}$ goes to infinity, the estimated p-value will tend to the ideal p-value, and the error in estimation will be kept minimal [68, 69].

The process for generating the randomized samples is as follows: Suppose dataset $\boldsymbol{D}$ is composed of M cases for a total of N genes and can be considered as an MxN matrix where l^{th} row $d_l = [d_{l1}, d_{l2}, ..., d_{lN}]$, $1 \leq l \leq M$, and d_{li} is the value of the i^{th} node (gene) in the l^{th} instance of input data. For each node X_i, we sample with replacement M instances from the i^{th} column, $[d_{1i}, d_{2i}, ..., d_{Mi}]^T$, of the original data matrix $\boldsymbol{D}$, and obtain the newly formed column of the bootstrapped data matrix D_k. The BDe score for this new data matrix is calculated, and the whole process is repeated B times. The approach we adopt here has previously been described, and applied to phylogeny reconstruction using molecular sequences [70, 68]. Bootstrap alone, which

is generally used to establish confidence, would not be fitting to assess significance in the current setting. Therefore, we provide randomization via bootstrapping, which provides an approximation of the null distribution. When scoring a BN, the rows of $\boldsymbol{D}$, which hold information reflecting the dependency relation between nodes of BN, are considered sequentially in order to update the parameters of each node on the BN. We randomize rows of $\boldsymbol{D}$ by changing the structure of columns of $\boldsymbol{D}$ via sampling with replacement each column of $\boldsymbol{D}$ separately. Querying each pathway database that holds few hundreds of networks generates a multiple hypothesis testing problem (utilized KEGG database contributes over 200 pathways). We address this issue by calculating FDR using the Benjamini-Hochberg procedure applied on p-values calculated for each pathway [71].

The overall complexity of the BPA is $O(N^2 + E^2 + RC^2G)$ where N is the number of nodes, E is the number of edges, G is the number of genes, C is the number of samples, and R is the number of bootstrap data sets. Modeling of pathways as DAGs is quadratic in N and E for Spirtes' algorithm and linear in N and E for Tarjan's algorithm. Data discretization is quadratic in C and linear in G. BDe score calculation is $O(RN^2G)$. Modeling pathways as DAGs is done off line, which does not lead to the computational time of a single analysis.

4.6 Creation of Simulated BNs

We created synthetic BNs for testing. Two of them are the well known Alarm and Asia BNs. The Alarm BN is designed to identify anesthesia problems with 37 nodes of 2, 3, or 4 states [72], and the 8 node binary Asia BN is designed to calculate the probability of a patient having tuberculosis [73]. In order to create synthetic BNs, we first calculated following parameters of for pathways listed in the KEGG database [61]:

These numbers represented the properties of typical biological networks. We then created 8 synthetic BNs (in addition to widely used Asia and Alarm BNs) following

	Mean	Std. Dev.
# of Nodes	24.292	21.466
# of Edges	25.307	34.148
Max Degree	5.714	6.189
Average Degree	2.058	2.211
Density	0.180	0.264

parametric distribution found in KEGG. Synthetic BNs have (avg. $\pm$ st. dev.) 25.38 ± 13.87 nodes, 24.38 ± 13.87 links, and 1.93 ± 0.08 average degree reflecting a spectrum of typical biological networks.

4.7 Identification of Data Fitting to Network

We tested our method to assign significance to BDe scores on 8 synthetic binary BNs of different sizes and the well known Alarm and Asia BNs. For each BN, we used two data sets to calculate BDe scores and their significance: one that follows the underlying CPT and one that does not. For a given BN, we randomly fixed Dirichlet hyper-parameters for each node and generated data that follow this CPT using the Bayes Net Toolbox (BNT) for Matlab [40]. CPTs calculated from this data are considered ideal as they are based on data following the fixed CPT for a given BN. For inconsistent CPTs, we chose Dirichlet hyper-parameters to be equal for each node, and obtained data using BNT. Therefore, the underlying CPTs calculated from this randomly generated data are considered to be the nonideal CPT as they are not based on data reflecting dependency structure implied by the given BN. We used data sets with a size of 1000, the bootstrap test count was chosen to be 2000, and an equivalent sample size of 1 for Dirichlet hyper-parameters was used during BDe score calculation. The results summarized in Table 4.1 show that when the data following underlying CPTs are used, p-values indicate strong significance, and are very close to zero ($p < 5x10^{-4}$). Conversely, when data generated using CPTs not consistent with independencies entailed by the BNs are used, p-values are severely deteriorated. Therefore, datasets produced from inconsistent CPTs are quickly detected by BPA.

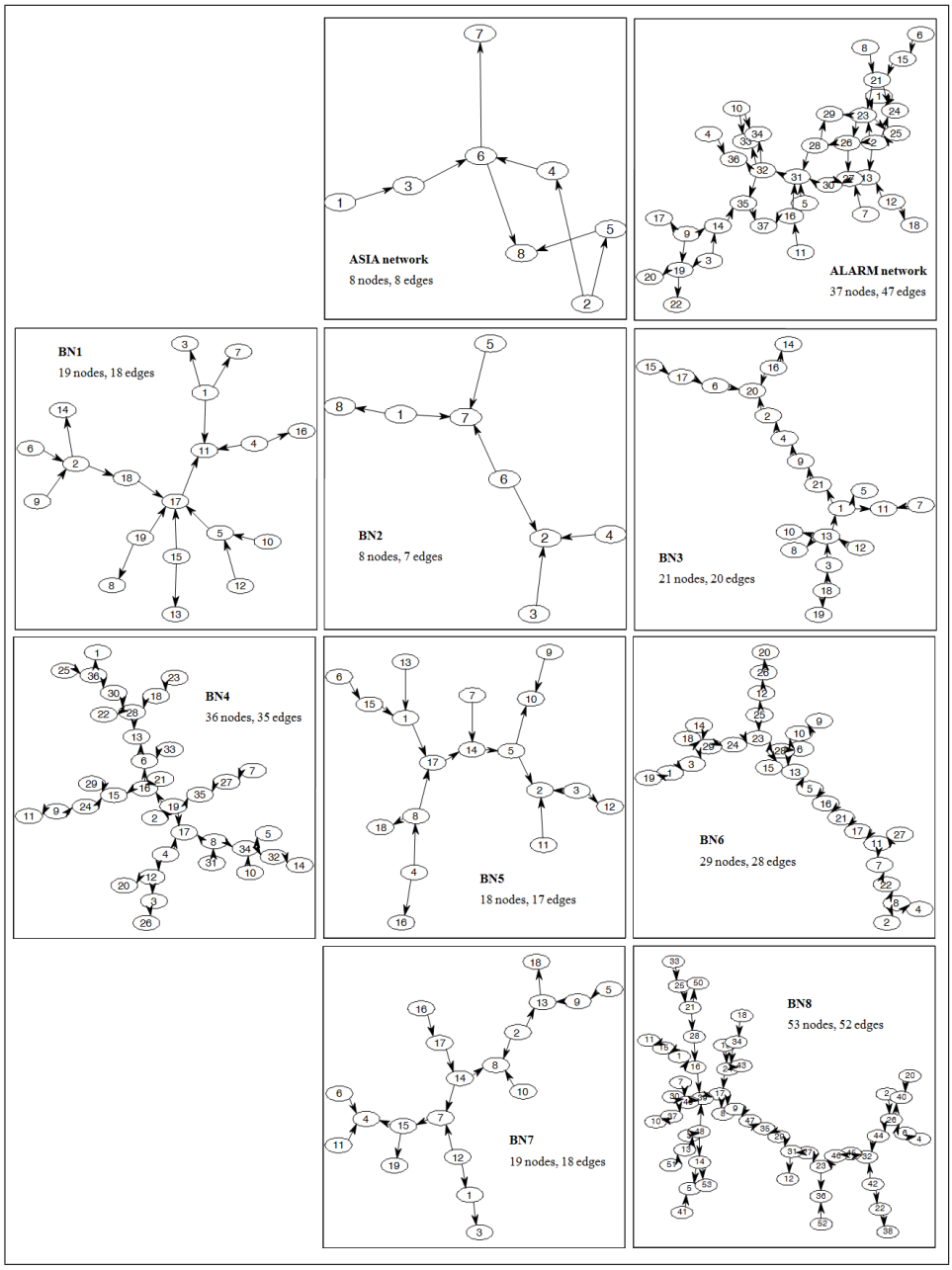

Figure 4.3 Graphs of simulated BNs.

4.8 Sample Size

In real life microarray experiments, the number of samples rarely exceeds 100 due to technical and financial limitations rendering limited number of observations to assess change in expression of a given gene. In order to see the effect of this limitation on BPA, we tested BNs in Table 4.1 with varying sizes of data sets (using CPTs that follow underlying BN structure: data set size 20-200, and CPTs that are non-ideal for

Table 4.1
Scores and p-values of scores for synthetic, Alarm, and Asia BNs.

		Data following CPT		Data inconsistent with CPT	
BN Name	**# of nodes**	**Score**	**p-value**	**Score**	**p-value**
Alarm	37	-9,955	$< 5x10^{-4}$	-22,600	0.56
Asia	8	-2,221	$< 5x10^{-4}$	-2,926	0.54
BN1	19	-9,344	$< 5x10^{-4}$	-10,213	0.62
BN2	8	-3,569	$< 5x10^{-4}$	-3,874	0.54
BN3	21	-10,844	$< 5x10^{-4}$	-12,763	0.55
BN4	36	-20,074	$< 5x10^{-4}$	-21,746	0.59
BN5	18	-9,607	$< 5x10^{-4}$	-10,245	0.50
BN6	29	-15,859	$< 5x10^{-4}$	-17,122	0.64
BN7	19	-9,804	$< 5x10^{-4}$	-10,996	0.65
BN8	53	-29,937	$< 5x10^{-4}$	-32,262	0.67

the underlying BN structure: data set size 20-300) and calculated the significance of corresponding BDe scores. Results are shown in Figure 4.4. For each data set size 50 runs have been performed. The average p-value of the runs, each obtained using 1000 bootstrapped samples, and associated standard errors are shown in Figure 4.2. In case of ideal CPTs, p-values start to get lower after a small increase in the sample size with highest attainable significance ($p<10^{-3}$) at sizes larger than 140. This is a dataset size that can be generated by the proposed method in an experimental setting where one has 12 samples in each of the two groups (BPA would generate $12x12 = 144$ observations for each BN. In case of non-ideal CPTs, p-values remain high regardless of the data set size (see also Table 4.1). These results suggest that BPA can successfully be used with datasets commonly seen in real experimental settings.

4.9 Change in Pathway Structure

Biological pathways may be incomplete as some of the nodes and/or edges for a given cascade of events may not have been identified yet. In order to test for the effect

of missing edges and nodes on the significance of BDe scores calculated by BPA, we systematically removed all possible k edge combinations, $1 \leq k \leq 5$, for BNs listed in Table 4.1. For each k, we calculated the average p-value obtained by removing different combinations using a data set size of 140 (following the underlying CPT) with 1000 bootstraps. We repeated the same procedure by removing all combinations of nodes. Results are shown in Figure 4.5. In both cases, all BNs maintain significant results despite removal of up to 5 edges and/or nodes, except for BN2 (8 nodes, 7 edges), BN5 (18 nodes, 17 edges), and Asia BN (8 nodes, 8 edges), which have smallest number of nodes and edges among the 10 synthetic networks. Note that the effect of node removal is more severe as when a node is removed so are all the edges connected to

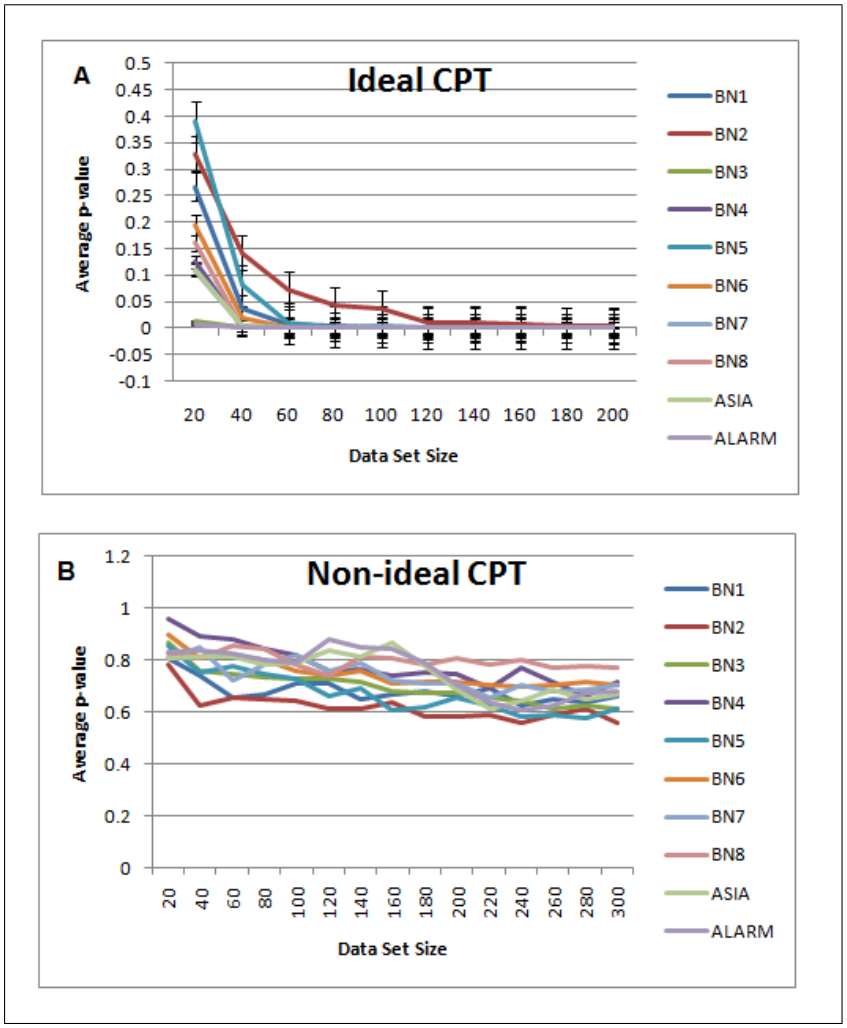

Figure 4.4 BPA performance with ideal and non-ideal CPTs for BNs listed in Table 4.1: (A) data follow underlying CPTs (B) data do not follow underlying CPTs. In each case average p-values of 50 runs have been calculated. Data set sizes are 20-200 in (A), 20-300 in (B) to depict better resolution, plateau, and real life settings, respectively.

it. Robustness to node/edge removal is possibly due to the factorized scoring metric and the BNs ability to focus on local interactions where each node is directly affected by a relatively small number of parent nodes and interactions. The synthetic networks tested follow average node/edge distribution in typical biological pathways and the removal of up to 5 nodes/edges is likely to be very high compared to pathway error instances seen in real biological pathways, which makes application of BPA for pathway analysis possible.

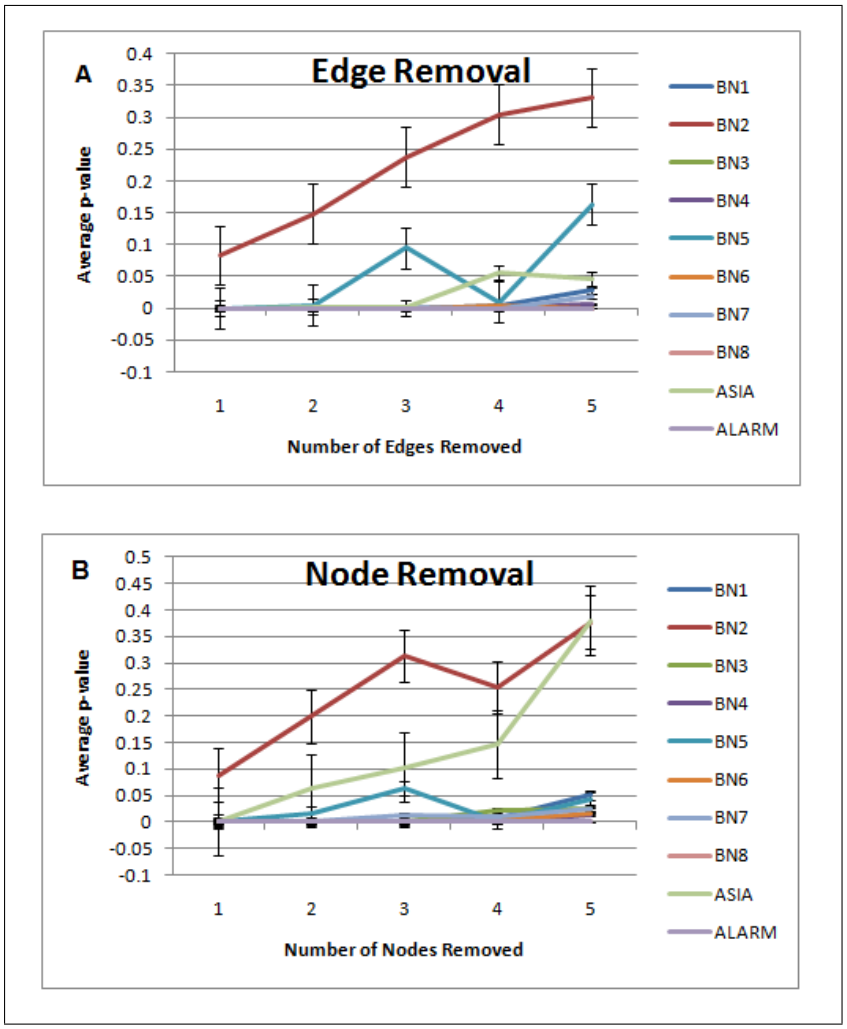

Figure 4.5 BPA performance with changing in network structure for BNs listed in Table 4.1: (A) progressive removal of edges in BNs (B) progressive removal of nodes in BNs In each case average p-values of 50 runs have been calculated. Data set sizes are 140 in (A) and (B) to depict better resolution, plateau, and real life settings, respectively.

4.10 Application to Synthetic Data Sets

We first looked at whether our method of solving cyclicty could create large cliques and inversely affect the BPA's overall performance. We generated 20 synthetic directed graphs containing cyclic paths following a SEM [74] using TETRAD IV [75]. The corresponding acylic collapsed versions of synthetic cyclic graphs show about 2.7-2.8 times the increase in number of nodes, number of edges, max degree, average degree, and density of the networks, on average. The BPA was run on a 1000 size data set with 1000 bootstraps, and resulted in significant p-values (lowest attainable) in all cases when we generated data that follow SEMs. These results suggest that our method of handling cyclicity may generate large cliques, however, the BPA is not adversely affected by the proposed method of generating DAGs from biological pathways (for details see [12]).

We then compared performance of the BPA with GSEA v2 [76] and a model-based approach, GlobalTest [77] using Bioconductor v2.7, GlobalTest v5.4.0 package on synthetic data sets approximating real microarray data. We generated synthetic transcriptional regulatory networks, and produced simulated gene expression data with noise using SynTReN v1.12 [78]. We created 60 synthetic networks (58/60 have cycles) with sizes ranging from 2 to 200. Details of the networks' parameters are included in the [12]. We randomly selected 25 out of 60 pathways to be active and SynTReN generated corresponding expression data sets for 20 test and 20 normal samples with 2249 synthetic genes adding a 4% noise level. For all three methods, we used 1,000 bootstraps, and chose a nominal p-value and FDR cut-off values of 0.05 and 0.25, respectively. We assessed accuracy (if a network -or corresponding gene set- is correctly called active/inactive) of the three algorithms for 10 simulated data sets, and provide the results in Table 4.2.

BPA was tested for fold change (FC) cut-off values (CO) of 2 and 3, and discretization levels of 2 (i.e. if a gene's FC is above CO or below 1/CO, that is i.e. a gene is dysregulated, we insert a 1 in the observation data matrix $\boldsymbol{D}$, otherwise we insert a 2) and 3 (i.e. we insert a 1, 2, or 3 in the observation data matrix $\boldsymbol{D}$, if a

Table 4.2
Average ± std. dev. of fraction of pathways accurately called active or inactive by BPA, GSEA and GlobalTest (GT).

BPA 2 Level		**BPA 3 Level**		**GSEA**	**GT**
FC 2	**FC 3**	**FC 2**	**FC 3**		
0.825 ± 0.047	0.838 ± 0.042	0.825 ± 0.047	0.838 ± 0.042	0.583 ± 0.001	0.400 ± 0.024

gene's FC is above CO, below 1/CO, between CO and 1/CO -inclusive-, respectively). These results suggest that BPA outperforms both GSEA and GlobalTest, and there is no significant change in using two or three levels of discretization in BPA with a slight improvement in performance when a fold change cut-off of 3 is used. We used 2-level discretization with a FC cut-off of 2 when applying BPA to real data sets. A 2-level discretization seems more natural as we do not keep activator/repressor information when modeling biological pathways as DAGs, and an FC cut-off of 2 allows BPA to capture subtle changes.

4.11 Application to Real RCC Data Set

We applied BPA on real RCC data sets in order to identify the underlying molecular mechanisms of the disease (Table 4.3). In each experiment, every cancer subtype was individually compared to the normal samples generating 16 data sets in total. BNs corresponding to biological pathways are scored using BDe with an equivalent sample size of 1 for Dirichlet hyper-parameters, and a selected subset of those that remain significant after 1000 bootstraps are shown in Table 4.4. CPU times for BPA is 30 ± 10 mins. (avg ± std. dev.) for the 16 analyzed datasets, where the time given is for the complete analysis of a single data set. Running times range from 17 to 57 minutes. Analysis was performed on an Intel Core 2 Duo CPU E6550 2.33 Ghz processor with Windows XP 32 bit OS. We also analyzed the 16 data sets using GSEA and GlobalTest. In all three methods, the p-value and FDR cut-off values were chosen

to be 0.05 and 0.25, respectively. In order to avoid the problem of pathway alignment that would arise if multiple pathway sources were used, we limited our analysis to the KEGG pathway database for this exemplary case. In case of GSEA and GlobalTest, we used MSigDB v2.5 (group CP under C2). The complete list of significant pathways for each method and a comparative analysis are included in the [12].

Table 4.3
Data sets used in BPA analysis of malignancies in kidney.

Data set name	**Number and types of samples**	**GEO #**
Lenburg et al.	17 (8 N, 9 cRCC)	GSE 781
Jones et al.	2 (23 N, 32 cRCC, 11 pRCC, 6 chRCC, 12 OC, 8 TCC)	GSE 15641
Furge et al. and Yang et al.	47 (12 N, 35 pRCC)	GSE 7023 and GSE 2748
Gumz et al.	20 (10 N 10 cRCC)	GSE 6344
Kort et al.	79 (12 N, 10 cRCC, 17 pRCC, 6 chRCC, 7 OC, 27 WT)	GSE 11024
Koeman et al.	32 (12 N, 10 chRCC, 10 OC)	GSE 8271
Wang et al.	22 (12 N, 10 cRCC)	GSE 14762

Most of the pathways deemed significant by BPA agree with those found in literature using genomic and proteomic approaches. For example arginine and proline metabolism, citrate cycle (TCA cycle), purine metabolism, fatty acid metabolism, pyruvate metabolism, glycolysis/gluconeogenesis, valine, leucine and isoleucine degradation pathways have been shown to be important in RCC analyzed using a proteomic approach [79]. On the other hand, significant pathways found in different subtypes show notable agreement among data sets analyzed. Using BPA, we found 25 pathways significant in at least half of the data sets; this number was only 9 for GSEA (see also [12]). In BPA, on average 10.6 data sets were found significant by each of 25 pathways (for a total data set occurrence of 265), while GSEA's average was 9.3 significant data sets per pathway (for a total data set occurrence of 84). When we considered path-

ways deemed significant for at least one data set, we found 129 pathways discovered by BPA yielding 571 data set occurrences in total and 121 pathways discovered by GSEA resulting in 390 data set occurrences. BPA was able to find 63% of pathways discovered by GSEA. Overall, these results indicate that BPA found a greater pathway base related and specific to RCC as compared to GSEA. We believe this enhancement in performance is due to the ability of BPA to take into account connectedness of genes that make up a pathway whereas in GSEA analysis such genes are only considered as a list, and no topological information is incorporated into the analysis. GlobalTest results indicated 199 significant pathways (out of 206) for 2974 data set occurrences yielding 14.95 average data sets (out of 16) deemed significant for each pathway. We include complete results of GlobalTest in [12] as pathway selection with this method showed little specificity (97% of tested pathways found significant for 90% of the data sets). Similar behavior for GlobalTest have been observed previously by other studies potentially due to distributional assumptions (that regression coefficients for the genes come from the same normal distribution) and errors in empirical covariance estimates made by this approach leading to high false positive rates especially in cases with small sample sizes [80, 81]. Furthermore, GlobalTest loses significant power when a given gene list contains correlated genes, which holds true for genes in a given pathway [82].

Out of 12 pathways shown in Table 4.4 that were shown to be related to RCC using an experimental proteomic approach, BPA found 117 data set occurrences for which these pathways were significant (61% of possible $12x16 = 192$ data set occurrences) while GSEA could only identify 50 (26%) dataset occurrences.

BPA was also able to yield high consensus among pathways found significant for a given RCC subtype. In Figure 4.6, we show the overlap of pathways found significant for cRCC subtype in four different data sets. More than 70% of pathways found in four data sets are shared by at least two data sets, while eight pathways were common to all. Among these eight pathways, six of them (except for nicotine and folate pathways) have been shown to be activated in RCC based on a proteomic approach [79].

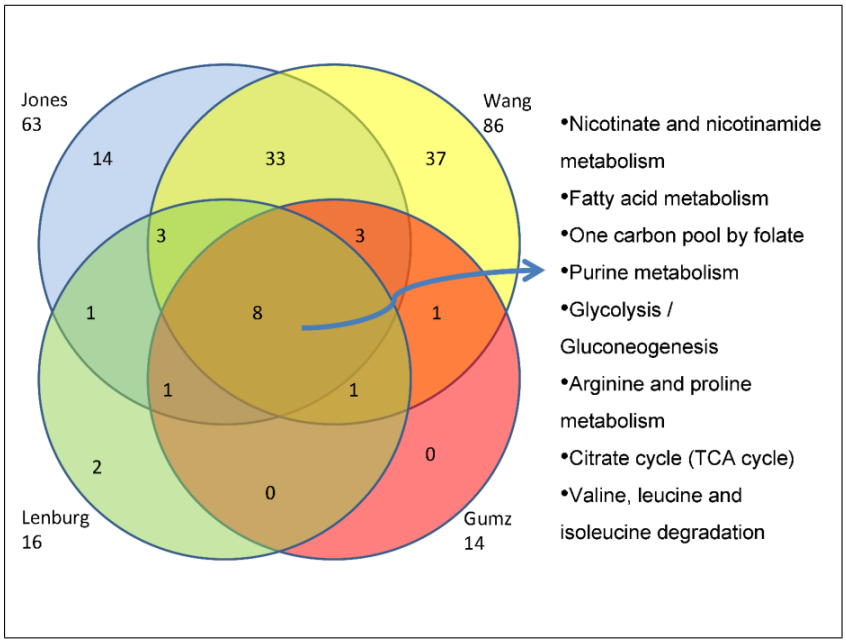

Figure 4.6 Venn diagram depicting pathways shared by BPA analysis of Jones, Lenburg, Gumz, and Wang cRCC data sets. Eight pathways at the intersection of all four analyses are indicated.

Results summarized in Table 4.4 put forth molecular mechanisms that are not only subtype specific but also commonly seen in different RCC tumors. The complexity/relevance of some of these pathways can be exemplified by the activation of the insulin signaling pathway through the activation of the insulin growth factor receptor-1 that activates the PI3K/Akt signaling pathway. PI3Ks catalyze the conversion of phosphatidylinositol bisphosphate (PIP) 2 to PIP 3 (inositol phosphate metabolism). PIP 3 acts as a second messenger to activate Akt. Akt mediates the activation of mTOR that is responsible for its effects on cell growth. In addition to activating the PI3K/Akt/mTOR pathway, IGFR-1 also activates the Ras/MAPK/Raf/MEK/ERK mitogenic signaling pathway (MAPK signaling pathway). Subsequently this leads to an activation of the cell cycle transition through stimulation of cyclin D1 (Cell cycle) and to increased cell proliferation. These prominent cellular pathways have been the objective for most of the targeted therapies now used in metastatic RCC. mTOR inhibitors act on PI3K/Akt/mTOR pathway leading to inhibition of protein synthesis and cell cycle arrest, while some receptor tyrosine kinase inhibitors (Sorafenib) also affect Raf, blocking the MAPK mitogenic signaling pathway. In addition, both approaches inhibit angiogenesis which has a strong impact in RCC tumorigenesis and

progression. Interestingly, further studies are underway analyzing possible composite or sequential therapies blocking these pathways for the identification of the optimal therapeutic approach. However, different targets within other pathways described here may lead to additional successful results, and should be explored further. According to our analysis one of these novel pathways could be the glyoxylate and dicarboxylate metabolism, which has been associated with lung cancer but not with RCC [83]. Indeed, metabolism related pathways have been shown to play a role in RCC progression, and would therefore be reasonable targets for further in depth analysis and in vitro testing.

Table 4.4
Selected significantly regulated pathways (p<0.05, FDR<0.25; *: BPA, §: GSEA). Boldface pathways are shown to be important in RCC using an experimental proteomic approach. (c: cRCC; p: pRCC; ch: chRCC; O: OC; T: TCC; W: WT)

Pathway/DataSet	Lenburg	Jones	Yang	Gumz	Kort	Koeman	Wang
Alanine and aspartate metabolism	c§	c*§ p* ch* O* T§	p*	c*	W*		c*§
Arachidonic acid metabolism	c§	c* p* ch*§ O* T*	p*		O* W*§	O*	c*
Arginine and proline metabolism	c*§	c* p*§ ch* O* T*§	p*	c*	p* W§		c*§
Cell cycle		c*§ p* ch* O* T*§	p*		p* W*§	O*	c*§
Citrate cycle (TCA cycle)	c*§	c*§ ch* O* T*		c*	O* W*	O*	c*
Drug metabolism - cytochrome P450		c* p* ch* O* T*			W*		
ECMreceptor interaction		c* p* ch* T*	p*		p* W*		c*
Fatty acid metabolism	c*§	c*§ p*§ ch* O* T*§	p*§	c*§	c*p*§O* W*	ch* O*	c*§
Focal adhesion		c* p* ch* O T*	p*		p* W*	O*	c*
Galactose metabolism		c* ch*			W*		c*
Glutamate metabolism	c§	c*§ p*§ch* O* T*	p*		p* W*	ch*	c*
Glycolysis / Gluconeogenesis	c*§	c*§ p* ch* O* T*	p*	c*§	c* p*O*W*§	ch* O*	c*
Glycosphingolipid biosynthesis		c* ch* O* T*	p*		W*	O*	c*
Inositol phosphate metabolism		c*§ p*§ch* O* T*					c§
Insulin signaling pathway		c* p*§ ch* O* T*			W*		c*
MAPK signaling pathway		c* p*§ ch* T*			W*		c*
Metab. of xenobiotics by cyt. P450	c§	c§ p* ch*§ O§		c§	c* O§ W*§	O§	c*§
Natural killer cell mediated cytotox.		c*§ p* ch* T*	p*		p* W*		c*§
Nicotinate and nicotinamide met.	c*§	c* p* ch*§ O* T*		c*	W*§	ch*	c*
Nitrogen metabolism	c§	c*§ p*§ ch* O* T*§	p*	c§	p* W*§		c*§
One carbon pool by folate	c*§	c* p* ch* O*§ T*	p*	c*	p* W*		c*
p53 signaling pathway		c§ ch*	p§		W§		c§
Pentose phosphate pathway		c* p* ch* O* T*	p*		p* W*		c*
Propanoate metabolism	c§	c*§ p§ ch* T§		c§	W*§		c§
Purine metabolism	c*	c* p* ch* O* T*	p*	c*	c*p*ch*O*W*	ch* O*	c*
Pyruvate metabolism	c*§	c*§ ch* O* T*§	p*		ch* O* W*§	ch* O*	c*
Pyrimidine metabolism		c* p* ch* O* T*	p*		p* W*§		c*
Retinol metabolism	c*	c* p* ch* O* T*	p*		p* W*§	ch* O*	c*
Urea cycle metabolism of amino g.	c*§	c*§ p§ ch*§ O§ T*§	p§		p§ ch§ W*	O§	c§
Valine leucine and isoleucine degr.	c*§	c*§ p*§ ch* O* T*§	p*	c*§	c*p*ch*O*W*§	ch* O*	c*§

5. LEARNING GENE INTERACTION NETWORKS USING EXTERNAL BIOLOGICAL KNOWLEDGE

We present a framework to incorporate multiple sources of prior knowledge, regardless of its type, into Bayesian network learning. The meaning of prior knowledge in our context is the enumeration of pairwise interactions of genes from biological information sources and the use of this information in Bayesian Network modeling. The proposed method is fully automatic, and does not use likelihood approximations when finding the optimal network that explains observed experimental data. We propose a novel framework that uses BN infrastructure itself to incorporate external biological knowledge, when learning networks. This infrastructure yields gene interaction information for pairs of genes, which can be used as informative priors in structure learning. We use this prior information to calculate the probability of a candidate graph G, and optimize the true model in the network learning process.

5.1 Methodology

Our methodology is depicted in Figure 5.1. A BN model for prior knowledge (called as BNP) is developed, using biological database information, to make inferences about interactions between gene pairs. The model is instantiated each time with the given gene expression correlation input to infer whether the gene pair is related or not, represented by a prediction value between 0 and 1. A prior knowledge matrix is populated with prediction values of all combinations of gene pairs. Using a proposed energy formula and informative prior formula, the prior knowledge is utilized in learning network structure with the greedy search algorithm.

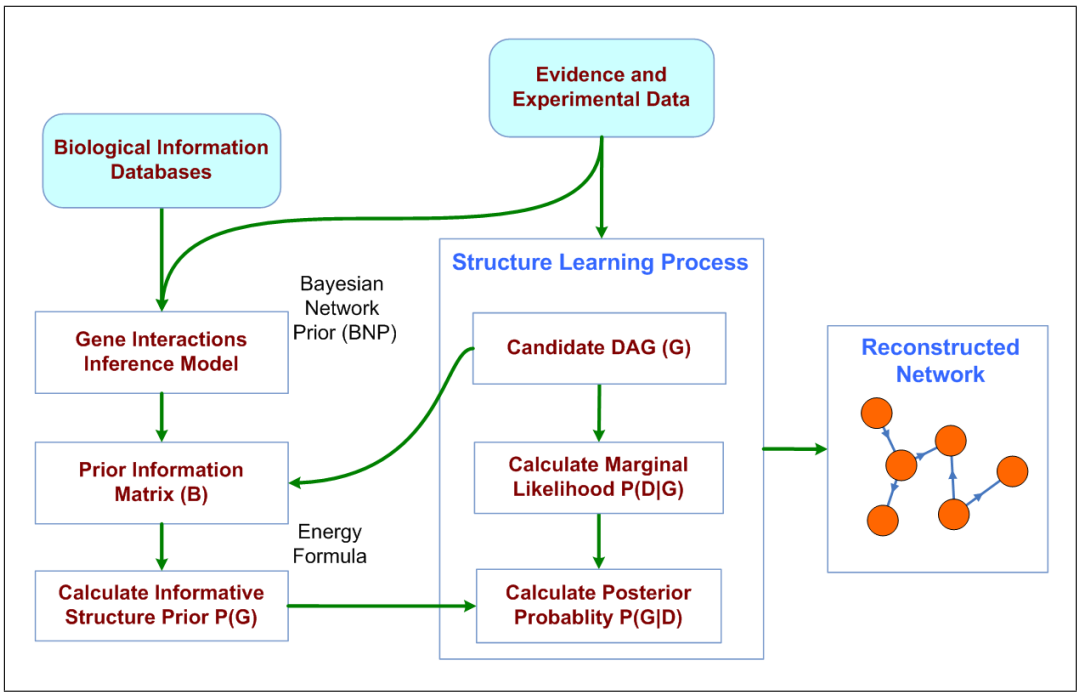

Figure 5.1 Layout of prior knowledge incorporation for gene networks

5.2 Scoring Function for Bayesian Network Models

The task of network inference (i.e. structure learning) is to make inferences regarding the graph G (i.e. DAG) that best explains the data. According to the Bayes' theorem, the posterior probability of the DAG G:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{5.1}$$

where $P(D|G)$ is the marginal likelihood of data, $P(D)$ is the probability of the data, $P(G)$ is the structure prior (or network prior) probability of the graph (DAG) G over all possible graphs, and $P(G|D)$ is the posterior probability of DAG G. Assuming a constant value for $P(D) = c$ is reasonable as $\boldsymbol{D}$ is observed. Use of uniform (flat) priors is common that $P(G)$s are assumed equal for lack of prior knowledge on Gs. The impact of the structure prior $P(G)$ is in penalizing over complex models. Ignoring the contribution of $P(G)$ may cause failure in differentiating between DAGs that are in the same Markov Equivalence set, in which several DAGs may support the same conditional probability distribution.

For discrete BNs, most of the learning tasks are performed by calculating $P(D|G)$ with Bayesian Dirichlet Equivalent (BDe) scoring function and by assuming uniform (flat) prior structure for all possible candidate DAGs [44]. The posterior probability of graph given the multinomial data is expressed as

$$P(G|D) = Score_{BDe} \cong P(D|G) \tag{5.2}$$

5.3 Network Learning Using Greedy Search with Informative Structure Priors

When the number of nodes N is not small, to find the best DAG by exhaustively considering all DAGs is computationally unfeasible as the number of DAGs increases super-exponentially in N. One way to handle this is to use a heuristic algorithm to search for G that maximizes $P(G|D)$. We chose the greedy search algorithm for its simplicity and integrated within it the score function that uses our informative structure model. The algorithm proceeds as follows. We start with an initial DAG, usually a DAG with no edges. At each step of the search, of all those DAGs in the neighborhood of our current DAG, we greedily choose the one that maximizes $P(G|D)$, which uses prior knowledge formula to calculate structure priors. We halt when no operation increases this score.

5.4 Previous Work on Informative Structure Priors

The use of uninformative flat structure priors may lead to a false model which is unable to describe the real network. In reality, the true parameter that needs to be optimized is $P(G|D)$. It is hypothesized that in case of learning gene interaction networks from high throughput biological data, $P(G|D)$ can be calculated using a model for $P(G)$.

Heckerman *et al.* has proposed a model of informative priors as follows [44]:

$$P(G) = c\kappa^{\delta} \tag{5.3}$$

where c is a constant, κ is an integer which has to be pre-defined, δ is the symmetrical difference between prior knowledge matrix $\boldsymbol{B}$ and the adjacency matrix of the candidate graph G. The symmetrical difference of the two sets is the union ($\cup$) of the two sets, minus ($\ddot{\cdot}\cdot$) their intersection ($\cap$) and would be expressed as the following:

$$\delta = \sum_{i}^{N} \left| \pi_i^B \cup \pi_i^G \ddot{\cdot}\cdot \pi_i^B \cap \pi_i^G \right| \tag{5.4}$$

where π_i is the parent set of node i, and N is the number of nodes.

Imoto *et al.* proposed to use Gibbs distribution for modeling of the structure priors of networks [6] as the following :

$$P(G) = Z^{-1} e^{-\beta E(G)} \tag{5.5}$$

and

$$Z = \sum_{G \in \rho} e^{-\beta E(G)} \tag{5.6}$$

where β is a hypermarameter, E is the energy function, Z is the partition function, and ρ is the set of all possible network structures. The computation of Z is intractable because it requires the sum over all possible networks, where the number of networks increases super-exponentially with the number of nodes. They describe $\boldsymbol{U}_{ij}$ as the interaction energy of the edge from gene i to gene j and assumes $\boldsymbol{U}_{ij}$ to be categorized into I values, $H_1, ..., H_I$ based on prior biological knowledge. For example, if one knows gene i regulates gene j, we set $\boldsymbol{U}_{ij} = H1$. However, if it is not known whether gene i regulates gene j or not, it is assigned that $\boldsymbol{U}_{ij} = H2$ and $0 < H_1 < H_2$. The total energy of the network G would then be defined as

$$E(G) = \sum_{i,j \in G} U_{ij} \tag{5.7}$$

They used Bayesian Network and Nonparametric Regression Criterion (BNRC) scoring method to learn structures in addition to an optimization algorithm to estimate structure, structure prior, and its parameters β, H_1, and H_2.

5.5 A Novel Model for Informative Structure Priors

In the calculation of informative structure priors, we propose a method that borrows ideas from both Imoto's and Heckerman's approaches [6, 44]. However, proposed method does not use categorized prior knowledge, but assigns probabilities to each candidate edge.

Let $\boldsymbol{B}$ be the prior information matrix, where $\boldsymbol{B}(i,j) = P(X_{ij})$, the probability of gene i and j interact based on external knowledge. Let A_G denote the adjacency matrix of the candidate graph G. We define the matrix $\boldsymbol{U}$ such that $\boldsymbol{U}(i,j) = 1 - [\boldsymbol{B}(i,j)A_G(i,j)]$, the element by element multiplication of $\boldsymbol{B}$ and A_G. Note that if there exists no edge from i to j in G, $\boldsymbol{U}(i,j) = 1$; and if there is an edge from i to j in G, $\boldsymbol{U}(i,j)$ is inversely proportional to our prior belief on the existence of the edge. The total energy of G is defined as:

$$E(G) = \sum_{i,j} U_{ij}/N^2 \tag{5.8}$$

where N is the number of nodes in G. We note that using categorical values for $\boldsymbol{U}_{ij}$ does not reflect the continuous case of having different probabilities for the presence of an edge between two nodes. $\boldsymbol{U}_{ij}$ can be defined without setting hard probability cutoffs (i.e. true or false edge) for the existence of an edge. Therefore, we prefer to assign values from previously evaluated prior probabilities $P(X_{ij})$ to the matrix $\boldsymbol{B}$ representing prior knowledge.

Informative structure prior is formulated as follows:

$$P(G) = C.e^{-\beta E(G)} \tag{5.9}$$

where C is a scaling constant. C may be assigned the total sum of all $P(G)$ calculated for each candidate DAG structure during the structure learning task. $P(G)$ is bounded by $E(G)$ since $E(G)$ has a scaling factor (N^2) as its denominator. The hyperparameter β is a fixed value used to scale $P(G)$. The hyperparameter β can be marginalized out from the equation in the interval of $[\beta_L, \beta_H]$ as follows:

$$P(G) = C.\frac{1}{\beta_H - \beta_L}\int_{\beta_L}^{\beta_H} e^{\beta E(G)} d\beta \tag{5.10}$$

For further use, the integral is calculated for a range of $E(G)$ and stored in a lookup table.

Ignoring the constant C, which does not affect relative comparison during scoring of graphs in structure learning, the posterior probability of graph given data with informative structure priors becomes:

$$P(G) = C.\frac{1}{\beta_H - \beta_L}\int_{\beta_L}^{\beta_H} e^{\beta E(G)} d\beta * Score_{BDe}(G, D) \tag{5.11}$$

5.6 Sensitivity Analysis of Prior Parameters

We tested the sensitivity of the proposed method to the hyperparameter β for a range of $\Delta\beta$ values of $[\beta_L, \beta_H]$ from 0.1 to 20 by performing receiver operating characteristic (ROC) curve analysis. The area under the curve (AUC) of top DAGs were calculated using posterior probability $P(G|D)$ with informative priors, and marginal likelihood $P(D|G)$ scores with flat priors for the Sprinkler BN shown in Figure 5.3. For each $\Delta\beta$ value, the scoring is repeated by generating new data sizes of 10, 20, 50, and 100. We plot the mean AUC values obtained versus the parameter interval values as shown in Figure 5.2. For $\Delta\beta$ of 1 and greater, there is a slight change in the mean AUCs as expected because the hyperparameter β is averaged out in the prior formula.

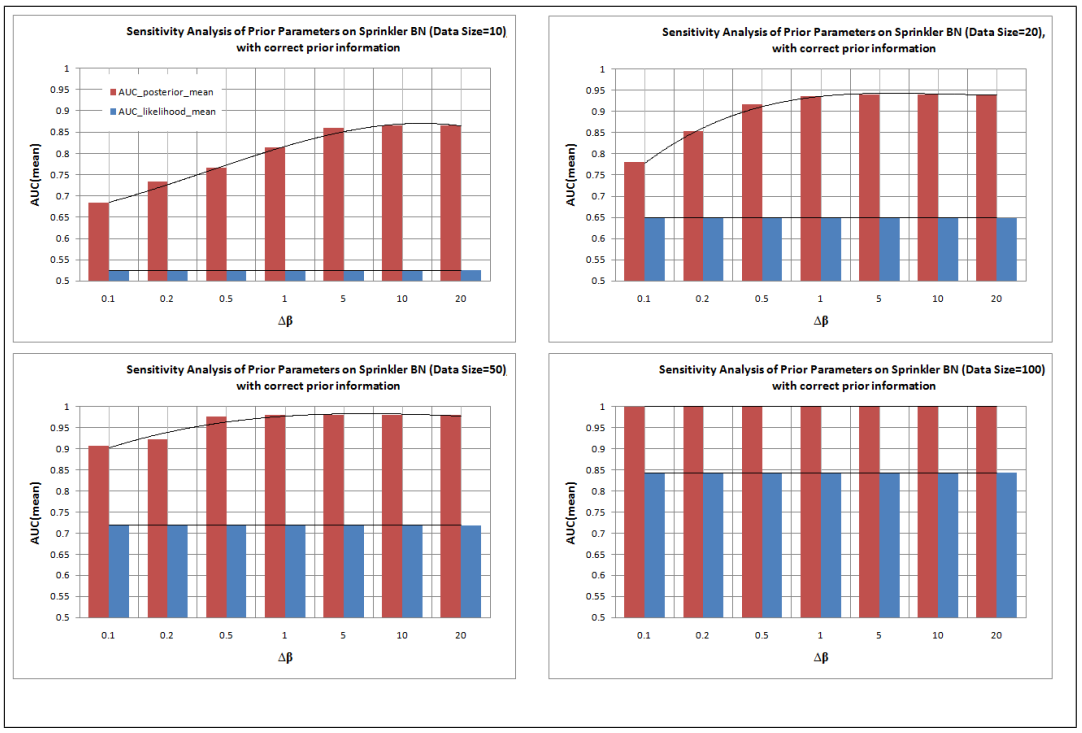

Figure 5.2 Sensitivity analysis of prior parameters on Sprinkler BN

5.7 Test on Prior Formula With Simulated Data

We tested the incorporation of $P(G)$ on the ubiquitous Sprinkler BN shown in Figure 5.3. Sprinkler BN is a binary network that shows CPTs for the events of the weather being cloudy, raining, grass being wet and the sprinkler being on.

We generated data that follows the model for a data set size of 100 and 1000, and randomly selected a DAG to represent prior knowledge seen in Figure 5.3. We performed scoring each of the 543 possible 4-node DAGs in a brute force approach without using a heuristic search algorithm. We calculated $P(D|G)$ according to Eq. 5.1 and $P(G|D)$ according to Eq. 5.2 with fixed $P(D)$ (i.e. data is given). $P(G)$ is calculated according to the proposed method in Eq. 5.10. Logarithm of the respective probabilities (scores) are shown in Table 5.1 for data set sizes of 1000 and 100.

We show top 10 DAGs with the highest $P(G|D)$ scores and make two observations: the true DAG, which had the DAG # 504, comes out at the top when $P(G|D)$

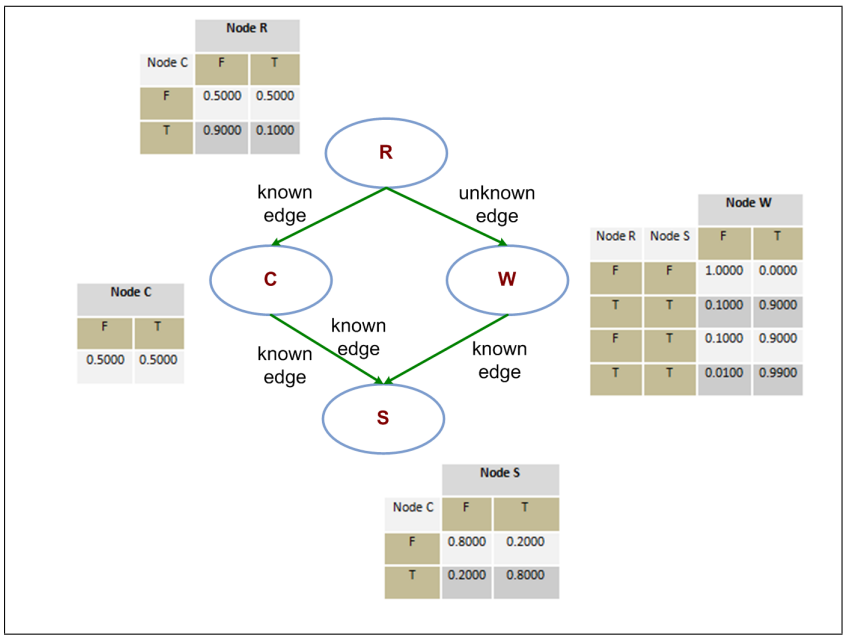

Figure 5.3 Sprinkler BN with corresponding CPTs and exemplary prior knowledge.

is considered although the log likelihood of this DAG, $P(D|G)$, is not the highest. Maximizing the log likelihood overfits, and results in ranking other DAGs at top.

By incorporating $P(G)$, we may differentiate between DAGs in the same Markov Equivalence Class (e.g. DAGs #504, 491, and 503 OR DAGs #506 and 518) using $P(G|D)$ although these DAGs render the same $P(D|G)$ scores.

We then generated 100 random 5-node BNs along with their CPTs. Data sets of size 100 were generated with BNT and both likelihood, and proposed scores were calculated for the DAGs. In applying the proposed approach, we distorted the prior matrix so that it did not always represent the true adjacency matrix. For a given DAG, we changed the real edge probabilities in the prior matrix with a fixed value between 0.7 and 1.0. This value was randomly chosen for each DAG. If no edge was present in the true DAG, this was reflected with a probability value of 0 in the prior knowledge matrix. Again a brute force method was used in that all 29,281 possible 5-node DAGs were created, and scored using both methods. In Figure 5.4, we show the percent rank of the true DAG for both methods with changing distortion levels. Percent rank is calculated as (Rank of the true DAG's score/Number of all DAGs)*100%. In all the

Table 5.1
Top 10 DAGs using data that follows CPTs described by the Sprinkler BN.

Data Set Size = 1000			**Data Set Size = 100**		
DAG #	**P(D|G)**	**P(G|D)**	**DAG #**	**P(D|G)**	**P(G|D)**
504	-1952.83	-2018.97	504	-215.242	-222.532
491	-1952.83	-2082.79	491	-215.242	-229.566
503	-1952.83	-2082.79	503	-215.242	-229.566
506	-1949.39	-2091.12	502	-222.226	-229.608
518	-1949.39	-2092.23	506	-214.584	-233.28
533	-1964.02	-2096.06	518	-214.584	-233.75
430	-2057.4	-2127.08	431	-219.48	-234.086
432	-1995.77	-2129.95	505	-219.48	-234.086
502	-2063.72	-2132.27	430	-227.136	-234.829
431	-2021.04	-2155.53	484	-222.226	-236.864

simulations, the proposed method ranked the true DAG higher than it was ranked using marginal likelihood scoring.

5.8 In-depth Study on Informative Structure Prior Function

We further analyzed the performance of the posterior probability scoring with informative priors against likelihood scoring with flat priors on the 4-node Sprinkler network and a randomly selected 5-node network (Figure 5.7) in detail. In the application of the proposed method, we distorted the prior knowledge matrix by assigning the same probability values, a, to the edges and the same probability values, b, to the entries with no edges using all combinations for a and b in the range [0,1] using 0.1 increments. The generated data set sizes were 10, 20, 50, and 100 and the process was repeated 10 times for each pair of (a, b) and for each data set size. The percent ranks of top DAGs achieved using posterior probability $P(G|D)$ with informative prior and

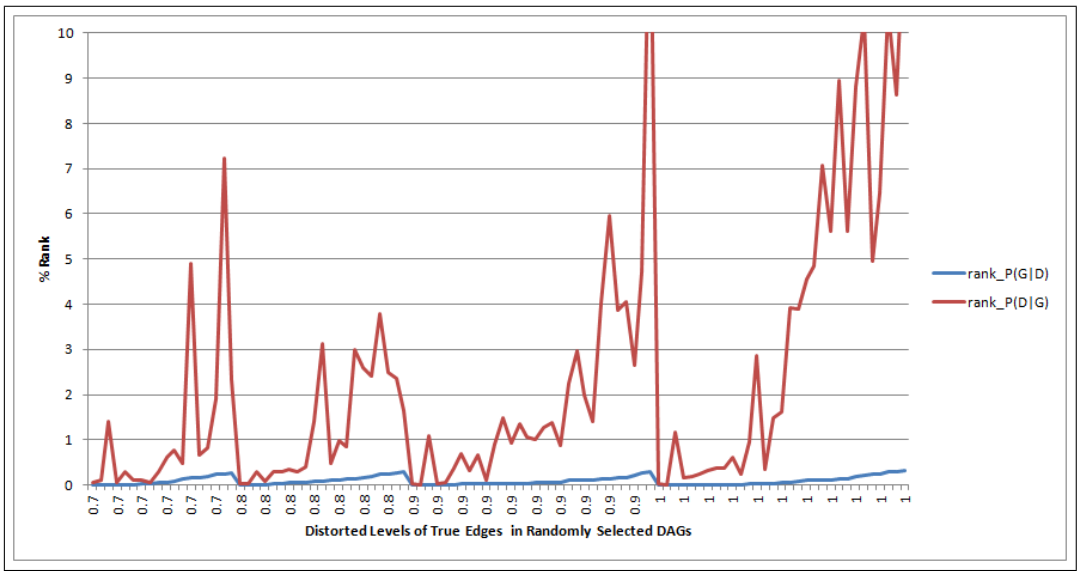

Figure 5.4 Comparison of $P(G|D)$ with informative prior function and $P(D|G)$ scoring on Random 5-Node BNs using a data size of 50.

likelihood $P(D|G)$ scores are shown in Figure 5.5. We also calculated the area under the ROC curve (AUC) values as seen in Figure 5.6. Y-axis represents probability values in the range of [0-1], and assigned to non-existing edges. X-axis represents probability values in the range of [0-1], and assigned to true edges in the graph. Note that heat maps of likelihood scoring do not encode probability ranges. Each heatmap is composed of percent ranks of 1210 repeats. Each color in the pixel encodes the average value of the percent ranks achieved for 10 generated data sets. Lower left quadrant represents prior knowledge that the true edge probability is in the range of [0.6-1], and the false edge probability is in the range of [0-0.4]. In this region, the overall mean percent rank of the posterior probability scores was about 3.51 for data size of 10, 2.51 for data size of 20, and 1.91 for data size of 50, whereas the likelihood scoring yielded 15.25 mean percent rank for data size of 10 and 5.39 for data size of 20, 1.05 for data size of 50. The upper right quadrant represents the region where all existing edges are set to probability values in the range of [0-0.4], and each non-existing edge are given probability values in the range of [0.6-1]. The results indicate that the incorrect prior knowledge is punished by the proposed informative prior model, even for small datasets. As the data size increased, both posterior probability score rankings and likelihood rankings resulted in high values.

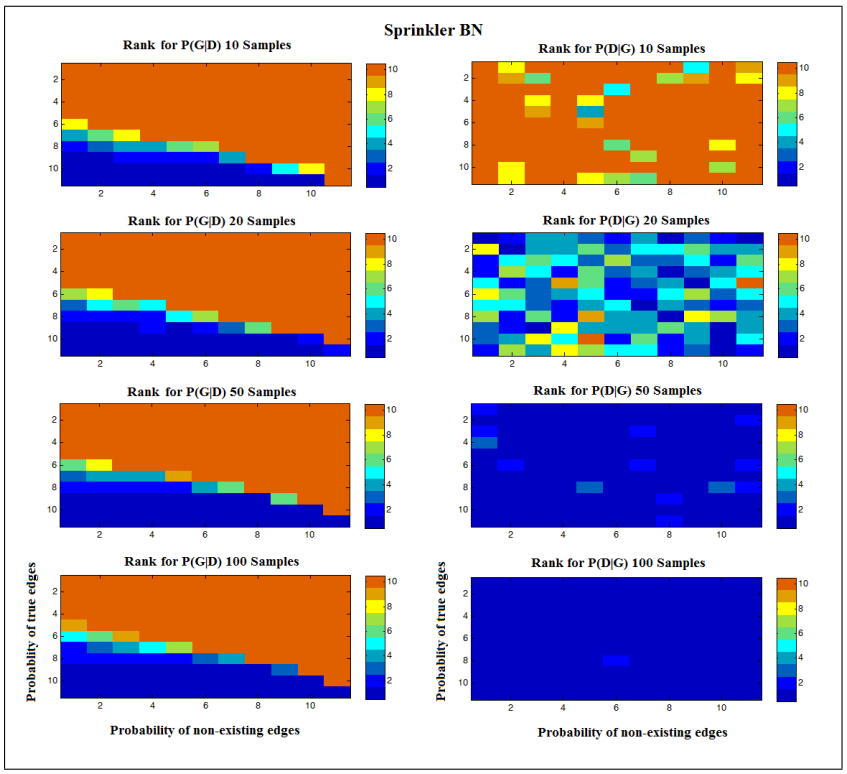

Figure 5.5 The heatmap to show rankings for learnt BNs for Sprinkler network according to posterior probability $P(G|D)$ with informative priors and likelihood $P(D|G)$ scores.

In the lower left quadrant, the overall mean AUC of the posterior probability scoring was about 0.71 for data size of 10, 0.75 for data size of 20, and 0.79 for data size of 50, whereas the likelihood scoring yielded 0.14 mean AUC for data size of 10, 0.6 for data size of 20, and 0.75 for data size of 50. As the data size increased, AUCs for both scoring models increased as expected. In the lower left quadrant, the overall mean AUC of the posterior probability scoring was over 80%. If the true edges are indicated in the prior matrix with high accuracy, then the proposed method performs quite well in finding the DAG under investigation. For example, in the Sprinkler BN, when the true edges are correctly represented with a 1 in the prior matrix, AUC remains at 100% even the false edge probabilities are as high as 0.9. For a fixed true edge probability of 0.9, the average AUC is around 92% when the false edge probability ranges from 0 to 0.9. These results indicate that incorrect prior knowledge is punished by our informative prior model severely, and the proposed system is more robust to false positives than it is to false negatives in the prior matrix. On the other hand, we rarely find high AUC values for the likelihood scoring compared to the performance of the proposed method

in the lower left quadrant.

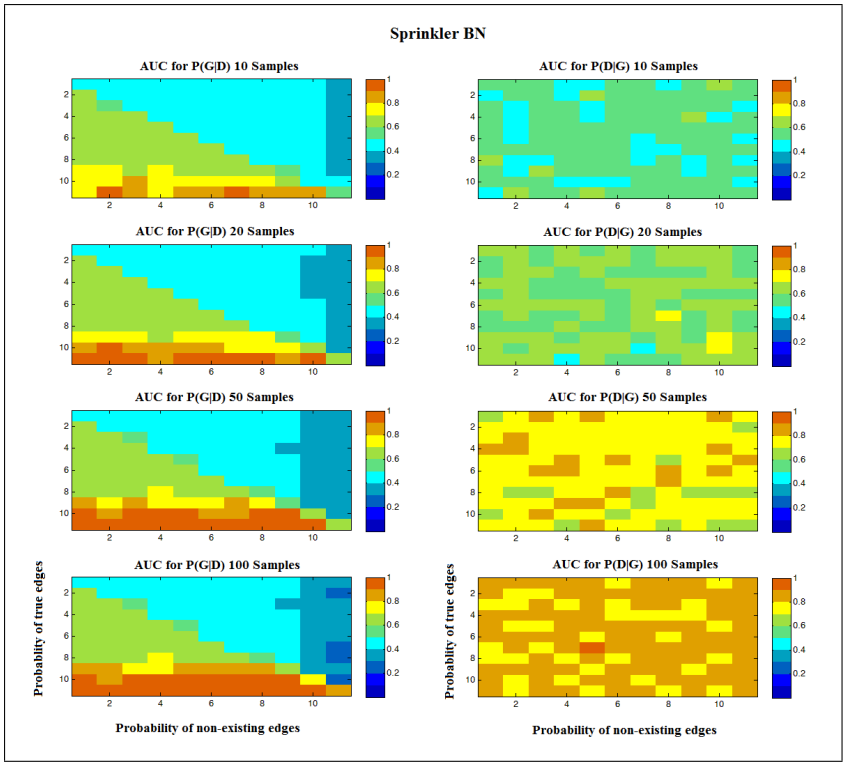

Figure 5.6 The heatmap to show mean AUC for learnt BNs for Sprinkler network according to posterior probability $P(G|D)$ with informative priors and likelihood $P(D|G)$ scores.

We also showed that a 5-node BN network with randomly assigned CPTs can be better approximated by using prior knowledge. The selected DAG structure and randomly generated CPTs are shown in Figure 5.7.

The overall mean ranking of the posterior probability scoring in the lower left quadrant (correct edge probability in the range of [0.6-1] and incorrect edge probability in the range of [0-0.4]) was about 0.375 for data size of 10, 0.24 for data size of 20, 0.22 for data size of 50, and 0.19 for data size of 100, whereas likelihood scoring yielded 24.63 mean ranking for data size of 10 and 10.84 for data size of 20, 4.67 for data size of 50, 2.28 for data size of 100.

These results indicate that as the network size grows, the likelihood score does not yield good results even for large data sizes as the search space increases super-exponentially. For a 4-node BN (like the Sprinkler BN), there are 543 possible DAGs

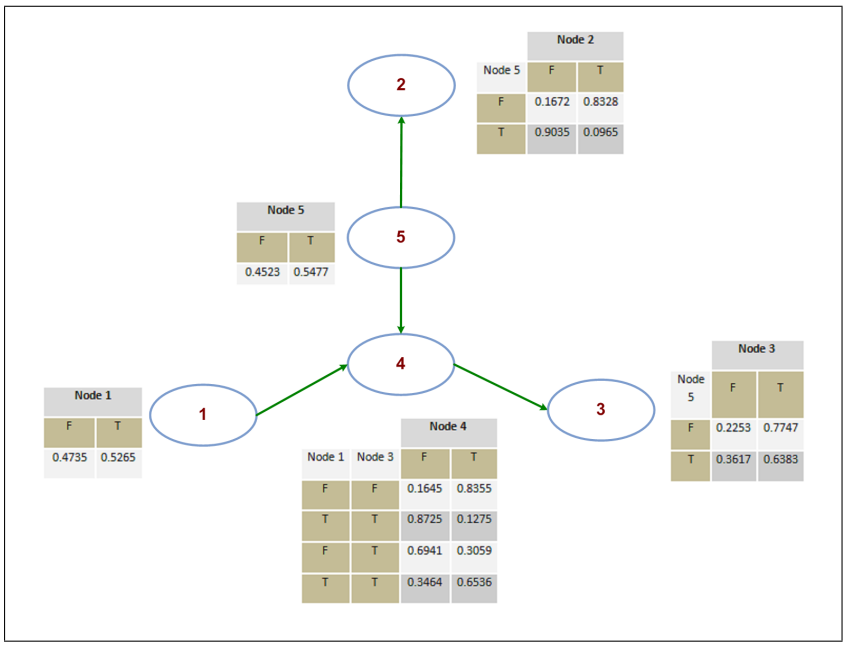

Figure 5.7 Randomly selected 5-node BN with randomly generated CPTs.

whereas for a 5-node BN there are 29,281 possible DAGs. For example, a data size of 50 for the 4-node network yields better mean AUC (i.e. 0.79) than that of the 5-node network (i.e. 0.5).

As seen in Figure 5.9, posterior probability scoring with the incorporation of prior knowledge yields higher AUCs. In the lower left quadrant, the overall mean AUC of the posterior probability scoring was about 0.64 for data size of 10, 0.65 for data size of 20, 0.64 for data size of 50, and 0.6635 for data size of 100, whereas likelihood scoring yielded 0.55 mean AUC for data size of 10, 0.57 for data size of 20, 0.50 for data size of 50 and 0.64 for data size of 100. As the data size increased, AUCs for both scoring models increased as expected. Nevertheless, the posterior probability scoring model for data size of 100 was able reach to 0.75 mean AUC to find the true graph structure for correct edge probability and extreme [0-0.9] incorrect edge probability range where the likelihood scoring had 0.64 mean AUC.

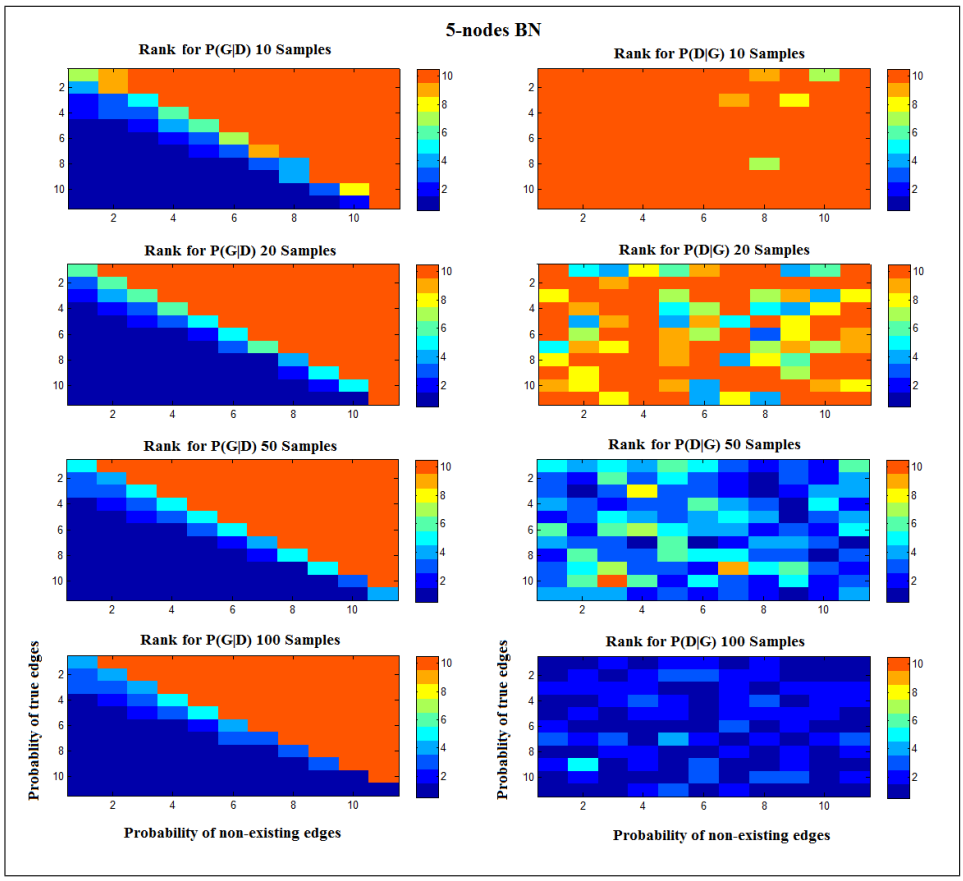

Figure 5.8 The heatmap to show rankings for learnt BNs for 5-node network according to posterior probability $P(G|D)$ with informative priors and likelihood $P(D|G)$ scores.

5.9 Prior Knowledge Inference Model

In this work, we present a model where existing external knowledge is used to determine a BN that can be utilized to deduce if any two genes interact with each other. Our goal is to use as much existing information as possible, in an intelligent way, to come up with a gene interaction network, which can further be used to identify the underlying interactome given high throughput biological data.

Previously, Troyanskaya *et al.* [84] proposed a Bayesian Framework for combining various data sources for gene function prediction. In this method a Naive Bayesian model was constructed. The parameters (CPTs) of the model were determined by experts. Then, a separate network was instantiated for each gene pair by initializing bottom level nodes with evidence. After that, the probability of functional relationship

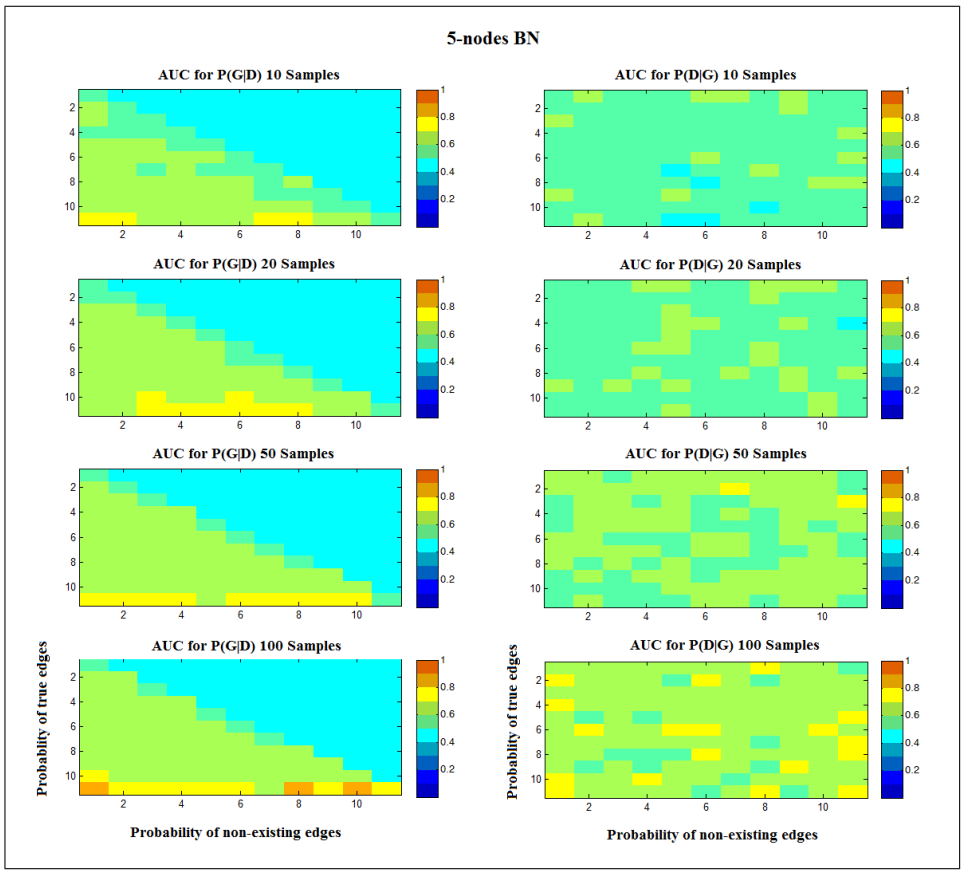

Figure 5.9 The heatmap to show mean AUC for learnt BNs for learnt BNs for 5-node network according to posterior probability $P(G|D)$ with informative priors and likelihood $P(D|G)$ scores.

between two genes was updated. The model is designed for functional prediction, not for gene interaction network learning.

Here, we describe a novel prior knowledge inference model that automatically learns parameters of the nodes used in a 20-node BN that predicts if two genes interact using external biological knowledge. To this end, pairwise gene relations for Homo Sapiens for different experimental methods have been extracted from pathway databases, a set of microarray experiments and a protein interaction database.

5.9.1 Data Preparation

Microarray co-expression relations were obtained using data from two different sources:

1-) The first dataset aims to provide a gene atlas for the human genes, and examines 79 normal human tissues with 158 samples [85]. Probe cell intensity (CEL) [85] files were acquired from the NCBI GEO database (GSE 1133). The platform used for this experiment is Affymetrix Human Genome U133A Array.

2-) The second dataset came from the Reference Database for Gene Expression Analysis (RefExA) that represents 70 normal human tissue samples (http://www.lsbm.org) run on the Affymetrix Human Genome U133A and U133B Arrays.

Using Affymetrix Expression Console v1.1, normalized gene expressios (at probe set level) signals were obtained using the MAS5 method. Probes which had absence call in all samples were omitted from further analysis. Pairwise correlations among probes were calculated using centered Pearson correlations. Data corresponding to a correlation value greater than 0.98 were used. A total of 71,617 gene relations were obtained.

KEGG [86], NCI/NATURE [62], Reactome [63] databases were utilized to obtain the interaction information based on biological pathways. 3,258 pairwise gene relations that exist in at least two of the three pathway databases were used for further analysis. Finally, 35,600 non-redundant pairwise gene interactions were obtained from the BioGrid database [87]. These interactions were based on 17 evidence types that are observed in different experimental assays.

In the end, we obtained 60,950 pair wise gene interactions by merging all three sources. In Table 5.2, we list the evidence types with descriptions. Approximately, 10,000 of these interactions were based on more than one type of evidence. A Gene Interaction (GI) node is appended to this evidence matrix (where rows represent gene pairs and columns represent evidence types) with a "true" value, if there were at least two evidence types implying interaction. BNP was built by learning both structure and parameters using greedy hill climbing.

5.9.2 Model Creation

The BNP model was trained, and tested by using a 5-fold cross validation approach, using the "bnlearn" R package [88]. In this approach, labeled dataset is randomized, and 80% of the data is used to train the model, and the remaining 20% of the data is used to test the model. Success rate of the model with respect to the data labels (positive and negative class) is calculated as classification error, which is the percentage of non-matching real and predicted values of the GI node. This procedure was repeated 5 times and the average error values were calculated. The goal of this exercise was to see if the proposed BNP model could identify gene interactions deduced from existing external knowledge.

The value of the GI node was inferred with the given evidences in the test data. For each pair in the test data, BNP was instantiated with the corresponding evidence vector, using the Loopy Belief Propagation inference algorithm [89]. If the inference value was greater than 0.5, as the GI node was taken to be 'true' (positive class); otherwise the GI node was set as 'false' (negative class). At the end of the 5-fold cross-validation, BNP rendered a classification error of 0.105 ± 0.003. In other words, BNP exhibits an accuracy of around 90%, when estimating if two genes interact given external biological knowledge.

Following the cross-validation, the BNP model was learnt using the complete evidence data. The strength of the probabilistic relationships expressed by the edges of BNP was measured using Friedman *et al.*'s bootstrap method [50] with 1000 repeats. We used model averaging to build a Consensus DAG for BNP, containing only the significant edges with a significance threshold of 0.413. The threshold for significant edges was determined using the method of Nagarajan *et al.*[90]. The Consensus DAG for BNP is shown in Figure 5.10.

Using BNP, one can now calculate the $P(G)$ for a given DAG. To this end, given a candidate DAG based on a HTBD, we can score the fitness of this DAG using $P(G|D)$, the true model, instead of $P(D|G)$, by incorporating $P(G)$.

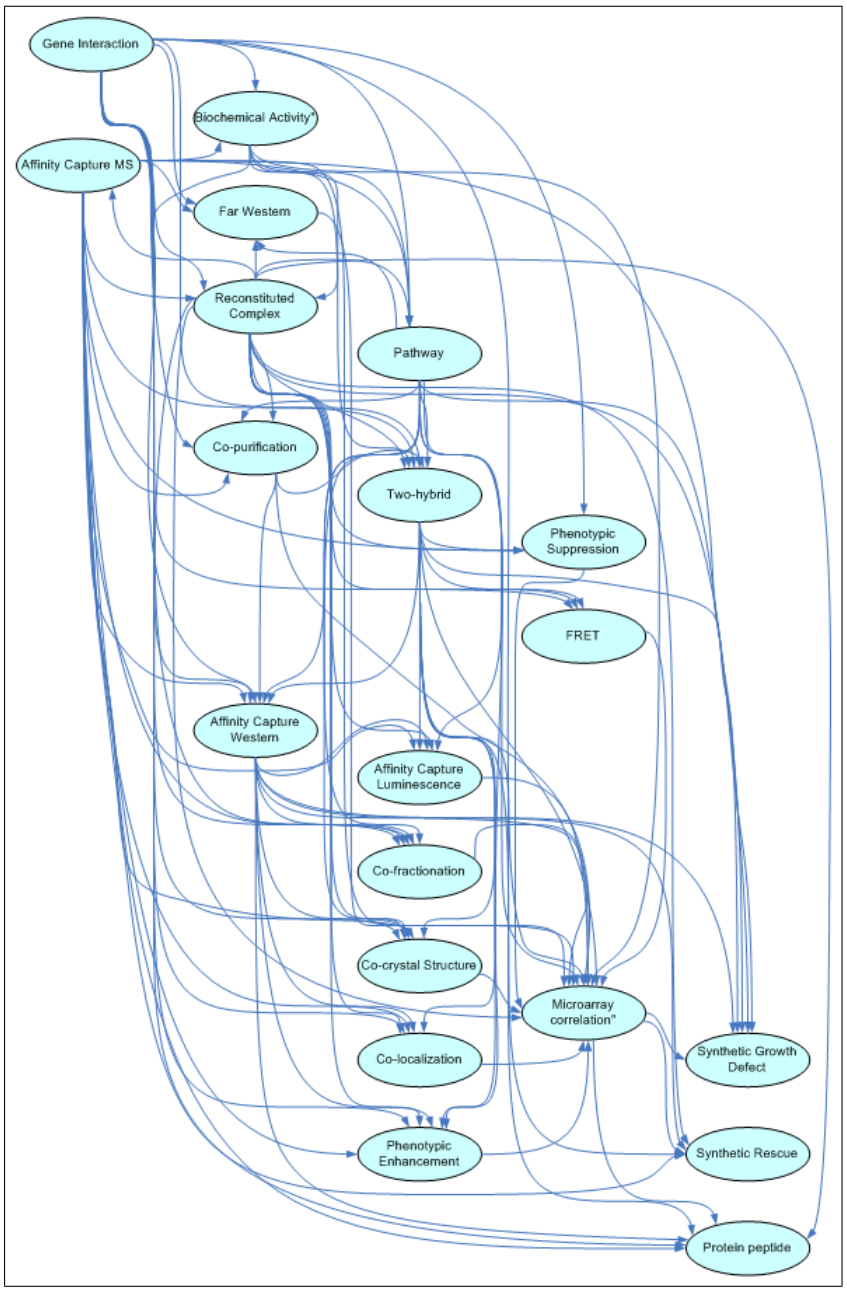

Figure 5.10 Consensus DAG of BNP.

5.10 Greedy Search Using Informative Priors

Informative prior formula evaluated in the previous sections was integrated into the greedy search algorithm to learn Bayesian networks. A set of DAGs were generated from KEGG pathways, using random CPTs fitting to the DAGs. Data of size 50 were generated, fitting to each BN. The cyclic pathways were converted to DAGs, using the method described in Chapter 4. The input pathways and their graph properties are

listed in Table 5.3.

The original DAGs implied from the pathways were used to obtain distorted prior matrices. In this case, distortion was introduced by adding Gaussian noise to the true DAG's adjacency matrix A_T to obtain the prior matrix $\boldsymbol{B}$. The distortion rate was calculated using $d = Fro(A_T - B)/Fro(A_T)$, where $Fro(.)$ represents the Frobenius norm [91]. The distortion rate was set to be in the [0.0 - 0.3] range, and this range was covered in 0.05 increments rendering 7 discrete rates. For each pathway and distortion rate, the synthetic data generation, distortion, and structure learning (both using the proposed method based on information priors and the likelihood based standard methods) steps were repeated five times. In Figure 5.11, we represent the average AUC values as a function of the introduced distortion rates. For all iterations, learnt DAGs with informative priors had higher AUCs (between 0.9 and 1) compared to the AUCs (between 0.5 and 0.6) for DAGs learnt with flat priors. The proposed method showed less variation in its performance measure compared to the standard methods. As the distortion level was increased, the difference between the mean AUC of DAGs learnt with informative prior and flat prior had a tendency to decrease.

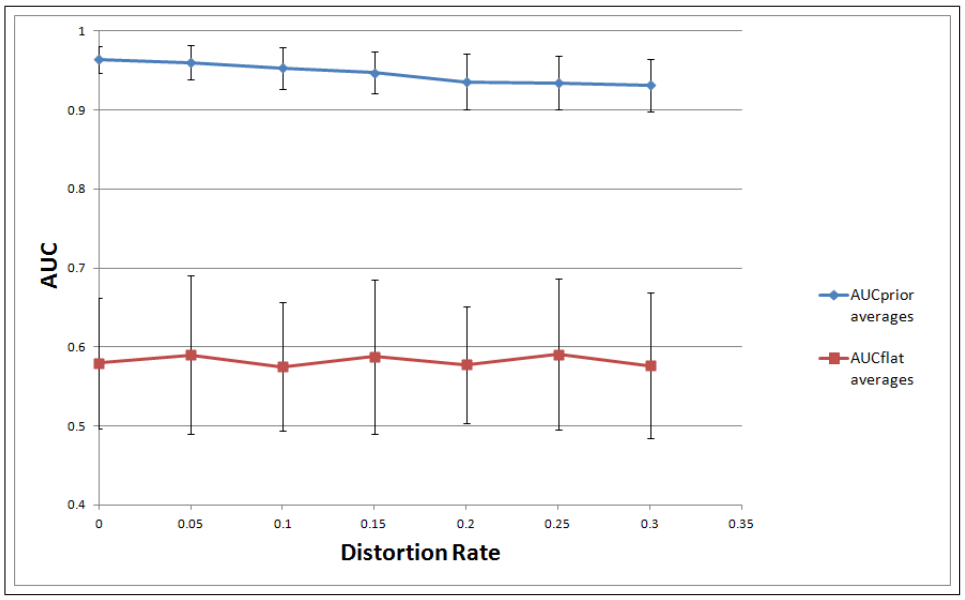

Figure 5.11 Overall AUC Plot for Several KEGG Pathways.

We used the same 23 KEGG pathways used in the previous step to generate simulated gene expression data. SynTReN v1.12 was used to generate the signal levels

for the genes in each of the 23 pathways with 10 control and 10 test samples and 10% background noise [78]. The input data for structure learning was obtained as described in Chapter 4. Briefly, columns represent genes in the pathway and rows represent observations. Each row (observation) is obtained by the fold change values of the genes between one pair of control and test samples. The input matrix, which consists of 100 observations in this case and reflects the distribution of fold change values between the two class of samples, was discretized into 3-levels using k-means clustering [92]. The inferred DAGs using prior knowledge (proposed method), and uninformative uniform prior (flat prior, standard methods) were compared to the original pathway structures using AUC values. This process was repeated 5 times for each pathway.

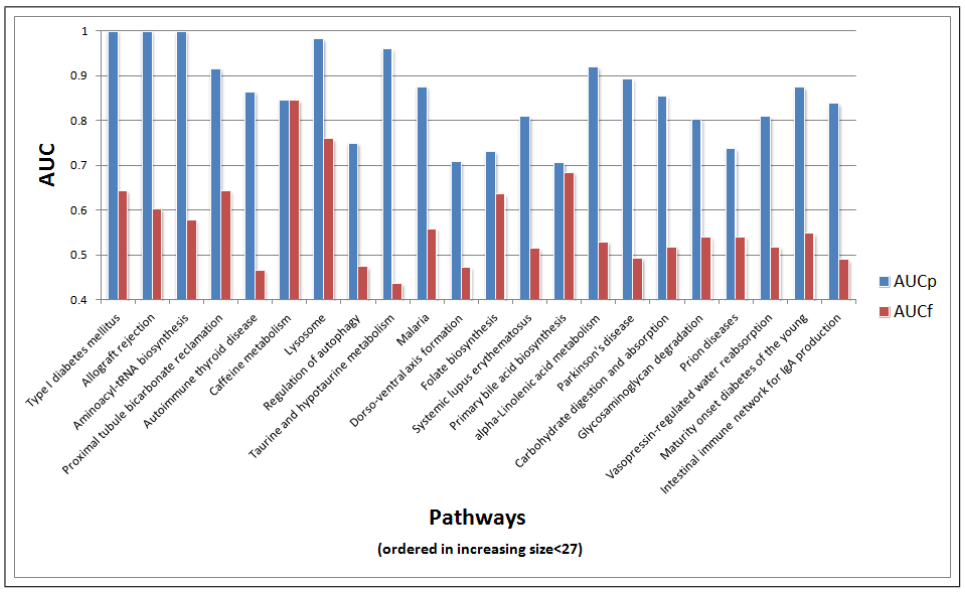

Figure 5.12 Comparison of AUC with informative prior vs. AUC with flat prior for KEGG pathway inference using synthetic high throughput data.

When the proposed method was employed, the BNP was instantiated for each gene pair in the given pathway to obtain the GI probability for the pair. These values made up the prior information matrix, $\boldsymbol{B}$. During the instantiation, the evidence vector used composed of existing evidence information for the gene pair in the databases, and the microarray correlation value calculated by the input gene expression data. This exemplifies the utility of the proposed method in which one can build interaction networks based on different evidence types originating from the performed experimental data. The BNP workflow then collates this observed information with the distilled

structure obtained from external knowledge bases to infer the GI probability for a pair of genes. The results for the AUC values between predicted and true DAGs for the 23 KEGG pathways using simulated gene expression data are shown in Figure 5.12. The proposed method dramatically surpassed classical structure learning methods where the AUC values for the DAGs found using the proposed method, on average, were 30% higher. Please note that "Caffeine metabolism" pathway gave rise to the same AUCs. This is probably BNP could not find enough evidence. The average AUC value for the proposed method was 86%. The improvement introduced by BNP shows the value of correctly incorporating existing external knowledge as reverse engineering gene interaction networks from noisy gene expression data is a difficult task.

5.11 Pathway Inference with Real Biological Data

We tested the proposed method using real gene expression data obtained from Renal Cell Cancer (RCC) and Normal samples as deposited in NCBI's GEO database with accession numbers GSE 11024 [93] and GSE 8271 [94]. Input data was obtained as described previously [12] and in Chapter 4. Briefly, MAS 5.0 normalized data was used and IDs in the array platform that correspond to a given node in a given pathway were pooled, and summarized as one representative signal value using one-step Tukey's biweight algorithm [67]. Observation matrix to be used in the structure learning process for a given pathway was obtained as explained in the previous subsection. We attempted at finding seven KEGG pathways shown to be important in RCC [12] using the expression values of the genes in these pathways from the two real RCC microarray data sets. The AUC values for the predicted and true pathways using the proposed method and likelihood scoring based methods are shown in Figure 5.13. In all seven cases, the proposed method found the underlying KEGG pathway with greater accuracy. The average AUC values for the proposed and existing methods were 89% and 57%, respectively.

The graph rendered from the KEGG database and the inferred DAG for "Glycosaminoglycan degradation" pathway with 19 nodes is seen in Figure 5.14

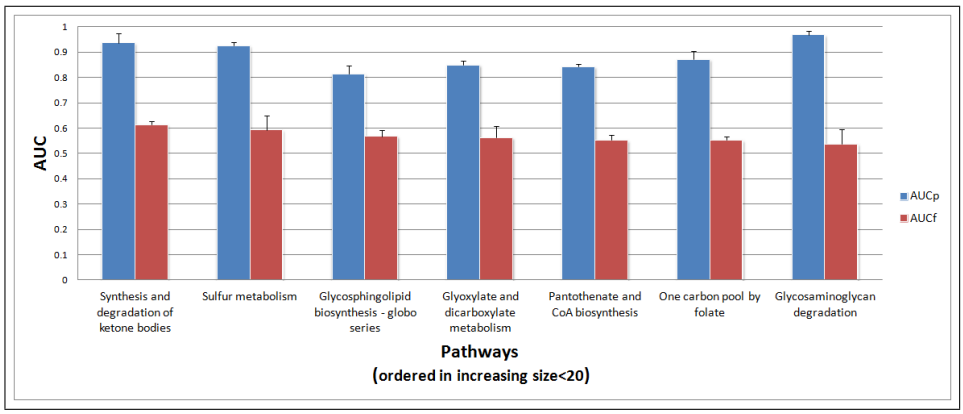

Figure 5.13 Comparison of AUC with informative prior vs. AUC with flat prior for pathway inference using real high-throughput biological data.

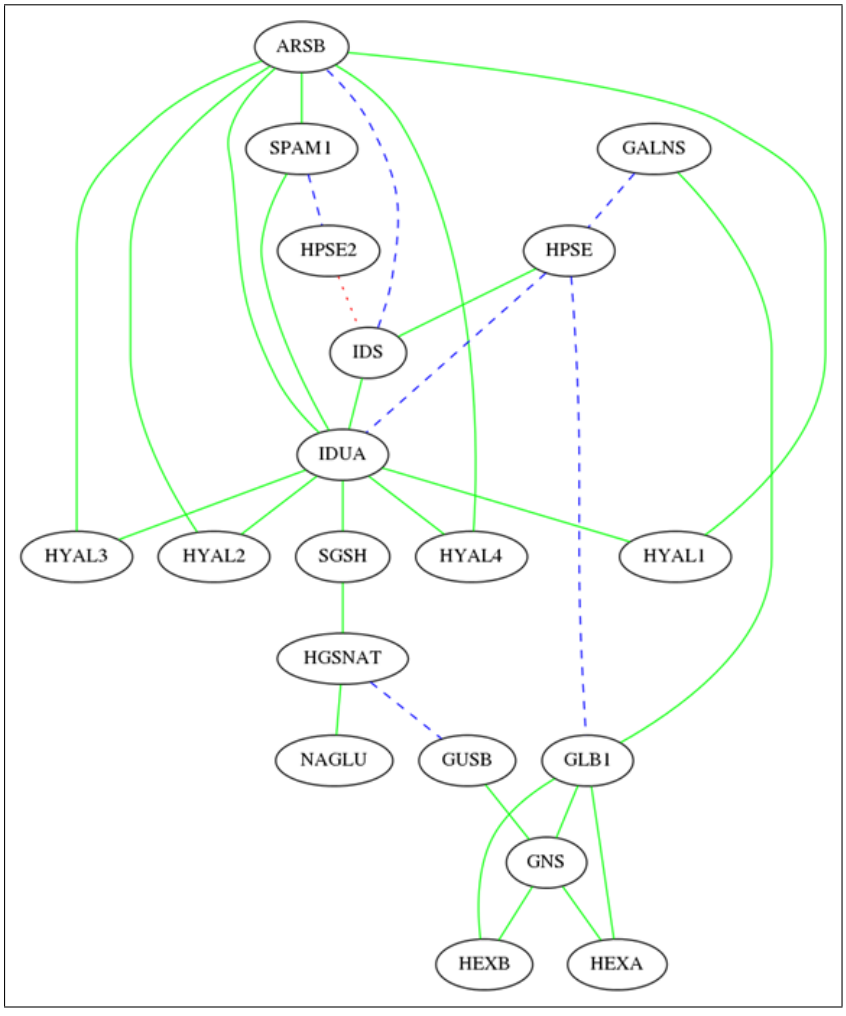

Figure 5.14 "Glycosaminoglycan degradation" Pathway. The green links are matching links between the KEGG pathway and the learnt DAG. Red dotted links are missing in the learnt DAG but exists in the KEGG pathway. Blue dotted links are inserted links that exist in the learnt DAG; the real pathway does not have these links.

Table 5.2
Evidence types used in building the Bayesian Network Prior (BNP).

Evidence Type	Description
Affinity Capture-MS	An interaction is inferred when a "bait" protein is affinity captured from cell extracts by either polyclonal antibody or epitope tag and the associated interaction partner is identified by mass spectrometric methods.
Biochemical Activity	An interaction is inferred from the biochemical effect of one protein upon another, for example, GTP-GDP exchange activity or phosphorylation of a substrate by a kinase. The "bait" protein executes the activity on the substrate "hit" protein.
Reconstituted Complex	An interaction is detected between purified proteins in vitro.
Pathway	An interaction is observed in at least two of the following three pathway databases: KEGG, NCI/NATURE, and Reactome.
Far Western	An interaction is detected between a protein immobilized on a membrane and a purified protein probe.
Co-purification	An interaction is inferred from the identification of two or more protein subunits in a purified protein complex, as obtained by classical biochemical fractionation or affinity purification and one or more additional fractionation steps.
Two-hybrid / TF Binding Site Localization	Bait protein expressed as a DNA binding domain (DBD) fusion and prey expressed as a transcriptional activation domain (TAD) fusion and interaction measured by reporter gene activation.
Phenotypic Suppression	A genetic interaction is inferred when mutation or over expression of one gene results in suppression of any phenotype (other than lethality/growth defect) associated with mutation or over expression of another gene.
FRET	An interaction is inferred when close proximity of interaction partners is detected by fluorescence resonance energy transfer between pairs of fluorophore-labeled molecules, such as occurs between CFP (donor) and YFP (acceptor) fusion proteins.
Affinity Capture-Western	An interaction is inferred when a bait protein affinity captured from cell extracts by either polyclonal antibody or epitope tag and the associated interaction partner identified by Western blot with a specific polyclonal antibody or second epitope tag. This category is also used if an interacting protein is visualized directly by dye stain or radioactivity. Note that this differs from any co-purification experiment involving affinity capture in that the co-purification experiment involves at least one extra purification step to get rid of potential contaminating proteins.
Co-localization	An interaction is inferred from co-localization of two proteins in the cell, including co-dependent association of proteins with promoter DNA in chromatin immunoprecipitation experiments.
Protein-peptide	An interaction is detected between a protein and a peptide derived from an interaction partner. This includes phage display experiments.
Co-crystal Structure	Interaction directly demonstrated at the atomic level by X-ray crystallography. Also used for NMR or Electron Microscopy (EM) structures. If a structure is demonstrated between 3 or more proteins, one is chosen as the bait and binary interactions are recorded between that protein and the others.
Affinity Capture-Luminescence	An interaction is inferred when a bait protein, tagged with luciferase, is enzymatically detected in immunoprecipitates of the prey protein as light emission. The prey protein is affinity captured from cell extracts by either polyclonal antibody or epitope tag.
Synthetic Growth Defect	A genetic interaction is inferred when mutations in separate genes, each of which alone causes a minimal phenotype, result in a significant growth defect under a given condition when combined in the same cell.
Phenotypic Enhancement	A genetic interaction is inferred when mutation or overexpression of one gene results in enhancement of any phenotype (other than lethality/growth defect) associated with mutation or over expression of another gene.
Co-fractionation	Interaction inferred from the presence of two or more protein subunits in a partially purified protein preparation. If co-fractionation is demonstrated between 3 or more proteins, one is chosen as the bait and binary interactions are recorded between that protein and the others.
Synthetic Rescue	A genetic interaction is inferred when mutations or deletions of one gene rescues the lethality or growth defect of a strain mutated or deleted for another gene.
Microarray Correlation	An interaction is inferred if the centered Pearson' s correlation is over 0.98.

Table 5.3
Several KEGG pathways and their graph properties.

Pathway	size (nodes)	order (edges)	density	max degree	average degree
D-Glutamine and D-glutamate metabolism	4	5	0.833333	3	2.5
Type I diabetes mellitus	5	3	0.3	2	1.2
Allograft rejection	7	4	0.190476	2	1.142857
Aminoacyl-tRNA biosynthesis	7	5	0.238095	3	1.428571
Autoimmune thyroid disease	7	4	0.190476	2	1.142857
Caffeine metabolism	7	6	0.285714	6	1.714286
Proximal tubule bicarbonate reclamation	7	6	0.285714	2	1.714286
Lysosome	8	7	0.25	7	1.75
Mineral absorption	8	6	0.214286	5	1.5
Regulation of autophagy	8	6	0.214286	5	1.5
Sulfur relay system	8	8	0.285714	5	2
Fat digestion and absorption	9	16	0.444444	5	3.555556
Taurine and hypotaurine metabolism	9	18	0.5	6	4
Malaria	11	7	0.127273	3	1.272727
Arrhythmogenic right ventricular cardiomyopathy (ARVC)	12	10	0.151515	4	1.666667
Dorso-ventral axis formation	12	14	0.212121	5	2.333333
Folate biosynthesis	13	11	0.141026	5	1.692308
Glycosphingolipid biosynthesis - globo series	13	66	0.846154	11	10.15385
Sulfur metabolism	13	24	0.307692	12	3.692308
Systemic lupus erythematosus	14	30	0.32967	8	4.285714
Glyoxylate and dicarboxylate metabolism	15	21	0.2	5	2.8
Primary bile acid biosynthesis	16	53	0.441667	13	6.625
Pantothenate and CoA biosynthesis	17	39	0.286765	8	4.588235

6. CONCLUSIONS

In this dissertation, we first describe a method that models biological pathways as BNs, and determines the fitness of given microarray data using the BDe score. The proposed method overcomes representation, mapping, data discretization, and cyclicity problems that arise in modeling pathways as BNs. We have chosen multinomial BNs with Dirichlet priors because 1) their posterior can be efficiently calculated in closed form; 2) they capture nonlinear interactions; 3) they render a plain model requiring less parameter adjustment. Moreover, algorithms scoring multinomial BNs have low time complexity. Alternative models such as linear Gaussian models, Gaussian process networks, or regression models are usually preferred in the task of structure learning. Linear Gaussian models and regression models in BN setting can only detect linear dependencies between child and parent variables. In Gaussian process BN models, Gaussian process priors are used as parametric families to model nonlinear relations. However, the problem then becomes one of selecting the best fitting covariance function and the number of its hyper-parameters in Gaussian process modeling, which requires various approximations and assumptions that may not be suitable in HTBD settings [95].

RCC represents a spectrum of genetically diverse epithelial tumors with a common derivation from the renal tubular epithelium and a variable clinical course. Approximately 30% of cases present with metastatic disease at initial diagnosis and 30% of initially organ confined cases develop metastases during later follow up. Since there are no reliable biomarkers available, patient management remains problematic despite improving understanding of the underlying molecular mechanisms. In particular, treatment of advanced RCC still poses a great challenge, as RCC is resistant to chemo- and radiation therapy and cytokine-based therapies offer only low clinical response rates with considerable toxicity. The advent of targeted therapy has brought exciting therapeutic options with promising clinical results, although the clinical benefit with respect to overall survival is only marginal. However, high-throughput technologies that an-

alyze the entire genome and proteome promise to elucidate the heterogeneity of this disease, and eventually enable a patient-tailored, individualized treatment. In contrast to the analysis of single genes, gene pathways enable us to see the context of complex interactions and to understand the biologic relevance of their expression. The plethora of pathways presented in this dissertation mirror complex biologic processes in kidney tumors and is often closely intertwined.

Overall, it is believed that the proposed approach, BPA, provides a unique perspective that merges Bayesian Network theory and HTBD analysis. Most BN models employed on HTBD use time series experimental designs in order to increase the size of the observed data. We have overcome this bottleneck, and provide a tool that can be used with most common experimental settings interpreting the results within the context of known biological pathways. Moreover, existing BN approaches on HTBD generally focus on building networks from input data, which makes these approaches applicable on a few dozens of genes due to the complexity of structure learning algorithms. Given the fact that high-throughput platforms generate data for tens of thousands of genes, the proposed approach makes use of relevant experimental information, and is applied to the complete data set within the context of known biological pathways. Our simulations on synthetic and real data sets show that BPA is able to successfully find molecular mechanisms that best describe underlying HTBD.

In this dissertation, we also developed a framework to incorporate multiple sources of prior knowledge, regardless of its type, into Bayesian network learning. In several studies the use of prior biological knowledge of the gene interaction network in conjunction with gene expression data has been suggested to improve the fidelity of network reconstruction. However, existing methods fail to rigorously harness and use the existing wide range of biological information.

A Bayesian Network Prior (BNP) model for assessing prior biological knowledge is developed, using biological database information, to make inferences about interactions between gene pairs. The model is instantiated each time with the given gene expression correlation input to infer whether the gene pair is related or not, repre-

sented by a prediction value between 0 and 1. A prior knowledge matrix is populated with prediction values for all combinations of gene pairs. Using a proposed energy and informative prior function, the prior knowledge is utilized in learning network structure with the Greedy Search algorithm in the BN framework. The goal on these applications were to construct gene networks from gene expression data and a list of genes of interest. We tested the sensitivity our prior model to its parameters. We analyzed the performance of the posterior probability scoring with informative priors against scoring with flat priors. Our BNP model incorporating selective evidence types rendered an accuracy of over 90% when estimating if two genes interact given external biological knowledge. This informative prior formula is integrated into the greedy search algorithm to learn Bayesian networks. It was shown that the proposed method was able to infer real pathways with high area under the curve (AUC) values, using both synthetic and real gene expression data.

6.1 Future Recommendations

The Bayesian Pathway Analysis described in this dissertation handles static gene expression data that is composed of two conditions. It could be useful to generalize it to handle multiple conditions. Another direction would be to extend the Bayesian Pathway Analysis to analyze time series gene expression data using DBNs. Additionally, during the study, it was observed that discretization step of the method is important and affects the strength and robustness of the approach. Therefore, it is necessary to research on optimal discretization techniques to be applied to the Bayesian Pathway Analysis.

In this dissertation, the integration of prior knowledge into Bayesian structure learning was accomplished via the Bayesian Network Prior and informative structure prior P(G) models. However, it is necessary to generalize the approach to include time series data. Therefore, further research is needed to investigate DBN learning algorithms such as REVEAL [39], and integrate BNP and P(G) methodology described in this dissertation into DBN to reveal biologically plausible gene networks derived from

time series data.

Bayesian structure learning algorithms and the improved algorithms described in this dissertation have certain limitations in terms of the size of the network to apply to. Any biological pathway may not work alone but function as a part of large atlas. Therefore, inferring large gene networks (atlas) from data would be an important research topic.

REFERENCES

1. Kitano, H., "Systems biology: a brief overview.," *Science*, Vol. 295, no. 5560, pp. 1662–1664, 2002.

2. Hecker, M., S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, "Gene regulatory network inference: data integration in dynamic models-a review," *BioSystems*, Vol. 96, no. 1, pp. 86–103, 2009.

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

4. Hartemink, A. J., D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Combining location and expression data for principled discovery of genetic regulatory network models," in *Proceedings of the Pacific Symposium on Biocomputing (PSB'02)*, pp. 437–449, Pacific Symposium on Biocomputing, 2002.

5. Tamada, Y., S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano, "Estimating gene networks from gene expression data by combining bayesian network model with promoter element detection," *Bioinformatics*, Vol. 19, no. suppl 2, pp. ii227–ii236, 2003.

6. Imoto, S., S. Kim, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano, "Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network," *Journal of Bioinformatics and Computational Biology*, Vol. 1, no. 02, pp. 231–252, 2003.

7. Werhli, A. V., and D. Husmeier, "Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge," *Stat Appl Genet Mol Biol*, Vol. 6, no. 1, p. 15, 2007.

8. Mukherjee, S., and T. P. Speed, "Network inference using informative priors," *Proceedings of the National Academy of Sciences*, Vol. 105, no. 38, pp. 14313–14318, 2008.

9. Nam, D., and S.-Y. Kim, "Gene-set approach for expression pattern analysis," *Briefings in Bioinformatics*, Vol. 9, no. 3, pp. 189–197, 2008.

10. Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, *et al.*, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, Vol. 102, no. 43, pp. 15545–15550, 2005.

11. Imoto, S., K. Sunyong, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano, "Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network.," in *Bioinformatics Conference, 2002. Proceedings. IEEE Computer Society*, pp. 219–227, IEEE, 2002.

12. Isci, S., C. Ozturk, J. Jones, and H. H. Otu, "Ipathway analysis of high-throughput biological data within a bayesian network framework," *Bioinformatics*, Vol. 27, no. 12, pp. 1667–1674, 2011.

13. Hosack, D. A., G. Dennis Jr, B. T. Sherman, H. C. Lane, R. A. Lempicki, *et al.*, "Identifying biological themes within lists of genes with ease," *Genome Biololgy*, Vol. 4, no. 10, p. R70, 2003.

14. Alexa, A., J. Rahnenfuhrer, and T. Lengauer, "Improved scoring of functional groups from gene expression data by decorrelating go graph structure," *Bioinformatics*, Vol. 22, no. 13, pp. 1600–1607, 2006.

15. Lu, Y., R. Rosenfeld, I. Simon, G. J. Nau, and Z. Bar-Joseph, "A probabilistic generative model for go enrichment analysis," *Nucleic Acids Research*, Vol. 36, no. 17, pp. e109–e109, 2008.

16. Bauer, S., J. Gagneur, and P. N. Robinson, "Going bayesian: model-based gene set analysis of genome-scale data," *Nucleic Acids Research*, Vol. 38, no. 11, pp. 3523–3532, 2010.

17. Galperin, M. Y., and X. M. Fernández-Suárez, "The 2012 nucleic acids research database issue and the online molecular biology database collection," *Nucleic Acids Research*, Vol. 40, no. D1, pp. D1–D8, 2012.

18. Pearl, J., and T. Verma, "A theory of inferred causation," in *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pp. 441–452, 1991. Morgan Kaufmann Publishers Inc.

19. Neapolitan, R. E., *Learning Bayesian Networks*, Pearson Prentice Hall Upper Saddle River, 2004.

20. Friedman, N., and D. Koller, "Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks," *Machine Learning*, Vol. 50, no. 1-2, pp. 95–125, 2003.

21. Lawrence, N., M. Girolami, and M. Rattray, *Learning and Inference in Computational Systems Biology*, MIT Press, 2010.

22. Collins, F. S., M. Morgan, and A. Patrinos, "The human genome project: lessons from large-scale biology," *Science*, Vol. 300, no. 5617, pp. 286–290, 2003.

23. Watson, J. D., and F. H. C. Crick, "Molecular structure of nucleic acids - a structure for deoxyribose nucleic acid," *Nature*, Vol. 171, no. 4356, pp. 737–738, 1953.

24. Crick, F., "Central dogma of molecular biology," *Nature*, Vol. 227, no. 5258, pp. 561–563, 1970.

25. Splinter, E., and W. de Laat, "The complex transcription regulatory landscape of our genome: control in three dimensions," *The EMBO journal*, Vol. 30, no. 21, pp. 4345–4355, 2011.

26. Ganis, J. J., N. Hsia, E. Trompouki, J. L. de Jong, A. DiBiase, J. S. Lambert, Z. Jia, P. J. Sabo, M. Weaver, R. Sandstrom, *et al.*, "Zebrafish globin switching occurs in two developmental stages and is controlled by the lcr," *Developmental Biology*, 2012.

27. DeRisi, J. L., V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, Vol. 278, no. 5338, pp. 680–686, 1997.

28. Lockhart, D. J., H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Norton, *et al.*, "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nature Biotechnology*, Vol. 14, no. 13, pp. 1675–1680, 1996.

29. Segal, E., 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*, Vol. 34, no. 2, pp. 166–176, 2003.

30. Friedman, N., "Inferring cellular networks using probabilistic graphical models," *Science*, Vol. 303, no. 5659, pp. 799–805, 2004.

31. Hasty, J., D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: in numero molecular biology," *Nature Reviews Genetics*, Vol. 2, no. 4, pp. 268–279, 2001.

32. Brazhnik, P., A. de la Fuente, and P. Mendes, "Gene networks: how to put the function in genomics," *TRENDS in Biotechnology*, Vol. 20, no. 11, pp. 467–472, 2002.

33. Stuart, J. M., E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, Vol. 302, no. 5643, pp. 249–255, 2003.

34. Heckerman, D., D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *The Journal of Machine Learning Research*, Vol. 1, pp. 49–75, 2001.

35. Pearl, J., *Causality: Models, Reasoning and Inference*, Cambridge University Press, 2000.

36. Spirtes, P., C. N. Glymour, and R. Scheines, *Causation Prediction & Search*, MIT press, 2000.

37. Verma, T., and J. Pearl, "Equivalence and synthesis of causal models," in *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 255–270, Elsevier Science Inc., 1991.

38. Murphy, K., S. Mian, *et al.*, "Modelling gene expression data using dynamic bayesian networks," tech. rep., Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

39. Liang, S., S. Fuhrman, R. Somogyi, *et al.*, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," in *Proceedings of the Pacific Symposium on Biocomputing (PSB'98)*, p. 2, Pacific Symposium on Biocomputing, 1998.

40. Murphy, K., *et al.*, "The bayes net toolbox for matlab," *Computing Science and Statistics*, Vol. 33, no. 2, pp. 1024–1034, 2001.

41. Schwarz, G., "Estimating the dimension of a model," *The annals of statistics*, Vol. 6, no. 2, pp. 461–464, 1978.

42. Gelman, A., J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, 2nd. ed., 1995.

43. Cooper, G. F., and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, Vol. 9, no. 4, pp. 309–347, 1992.

44. Heckerman, D., D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, Vol. 20, no. 3, pp. 197–243, 1995.

45. Geiger, D., and D. Heckerman, "Learning gaussian networks," in *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pp. 235–243, Morgan Kaufmann Publishers Inc., 1994.

46. Friedman, N., and M. Goldszmidt, "Learning bayesian networks with local structure," in *Learning in Graphical Models*, pp. 421–459, Springer, 1998.

47. Chickering, D. M., D. Heckerman, and C. Meek, "A bayesian approach to learning bayesian networks with local structure," in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pp. 80–89, Morgan Kaufmann Publishers Inc., 1997.

48. Robinson, R. W., "Counting labeled acyclic digraphs," *New directions in the theory of graphs*, Vol. 239, p. 279, 1973.

49. Madigan, D., 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*, Vol. 25, no. 11, pp. 2493–2519, 1996.

50. Friedman, N., I. Nachman, and D. Peér, "Learning bayesian network structure from massive datasets: the «sparse candidate «algorithm," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 206–215, Morgan Kaufmann Publishers Inc., 1999.

51. Moore, A., and W.-K. Wong, "Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning," in *Proceedings of International Conference on Machine Learning*, Vol. 20, p. 552, AAAI, 2003.

52. Pena, J. M., J. Björkegren, and J. Tegnér, "Growing bayesian network models of gene networks from seed genes," *Bioinformatics*, Vol. 21, no. suppl 2, pp. ii224–ii229, 2005.

53. Friedman, N., "Learning belief networks in the presence of missing values and hidden variables," in *Machine Learning-International Workshop*, pp. 125–133, Morgan Kaufmann Publishers Inc., 1997.

54. Friedman, N., "The bayesian structural em algorithm," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 129–138, Morgan Kaufmann Publishers Inc., 1998.

55. Jones, J., H. Otu, F. Grall, D. Spentzos, H. Can, M. Aivado, A. S. Belldegrun, A. J. Pantuck, and T. A. Libermann, "Proteomic identification of interleukin-2 therapy response in metastatic renal cell cancer," *The Journal of Urology*, Vol. 179, no. 2, pp. 730–736, 2008.

56. Mills, E. J., B. Rachlis, C. O'Regan, L. Thabane, and D. Perri, "Metastatic renal cell cancer treatments: an indirect comparison meta-analysis," *BMC Cancer*, Vol. 9, p. 34, 2009.

57. Brugarolas, J., *et al.*, "Renal-cell carcinoma–molecular pathways and therapies," *The New England Journal of Medicine*, Vol. 356, no. 2, p. 185, 2007.

58. Jones, J., H. Otu, D. Spentzos, S. Kolia, M. Inan, W. D. Beecken, C. Fellbaum, X. Gu, M. Joseph, A. J. Pantuck, *et al.*, "Gene signatures of progression and metastasis in renal cell cancer," *Clinical Cancer Research*, Vol. 11, no. 16, pp. 5730–5739, 2005.

59. Furge, K. A., J. Chen, J. Koeman, P. Swiatek, K. Dykema, K. Lucin, R. Kahnoski, X. J. Yang, and B. T. Teh, "Detection of dna copy number changes and oncogenic signaling abnormalities from gene expression data reveals myc activation in high-grade papillary renal cell carcinoma," *Cancer Research*, Vol. 67, no. 7, pp. 3171–3176, 2007.

60. Yang, X. J., M.-H. Tan, H. L. Kim, J. A. Ditlev, M. W. Betten, C. E. Png, E. J. Kort, K. Futami, K. A. Furge, M. Takahashi, *et al.*, "A molecular classification of papillary renal cell carcinoma," *Cancer Research*, Vol. 65, no. 13, pp. 5628–5637, 2005.

61. Kanehisa, M., M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, *et al.*, "Kegg for linking genomes to life and the environment," *Nucleic Acids Research*, Vol. 36, no. suppl 1, pp. D480–D484, 2008.

62. Schaefer, C. F., K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow, "Pid: the pathway interaction database," *Nucleic Acids Research*, Vol. 37, no. suppl 1, pp. D674–D679, 2009.

63. Vastrik, I., P. D'Eustachio, E. Schmidt, G. Joshi-Tope, G. Gopinath, D. Croft, B. de Bono, M. Gillespie, B. Jassal, S. Lewis, *et al.*, "Reactome: a knowledge base of biologic pathways and processes," *Genome Biology*, Vol. 8, no. 3, p. R39, 2007.

64. Romero, P., J. Wagg, M. L. Green, D. Kaiser, M. Krummenacker, and P. D. Karp, "Computational prediction of human metabolic pathways from the complete human genome," *Genome Biology*, Vol. 6, no. 1, p. R2, 2004.

65. Spirtes, P., "Directed cyclic graphical representations of feedback models," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 491–498, Morgan Kaufmann Publishers Inc., 1995.

66. Tarjan, R., "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, Vol. 1, no. 2, pp. 146–160, 1972.

67. Hoaglin, D. C., F. Mosteller, and J. W. Tukey, *Understanding Robust and Exploratory Data Analysis*, Wiley New York, 2000.

68. Davison, A. C., and D. V. Hinkley, *Bootstrap Methods and Their Application*, Cambridge University Press, 1997.

69. Efron, B., and R. Tibshirani, *An Introduction to The Bootstrap*, Chapman & Hall/CRC, 1993.

70. Brown, J., "Bootstrap hypothesis tests for evolutionary trees and other dendrograms," *Proceedings of the National Academy of Sciences*, Vol. 91, no. 25, pp. 12293–12297, 1994.

71. Benjamini, Y., and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 57, pp. 289–300, 1995.

72. Beinlich, I. A., H. J. Suermondt, R. M. Chavez, and G. F. Cooper, *The ALARM Monitoring Aystem: A Case Study with Two Probabilistic Inference Techniques for Belief Networks*, Springer, 1989.

73. Lauritzen, S. L., and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 50, pp. 157–224, 1988.

74. Bollen, K. A., *Structural Equation Models*, Wiley Online Library, 1998.

75. Scheines, R., "Estimating latent causal influence: Tetrad ii model selection and bayesian parameter estimation," in *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, pp. 445–456, Morgan Kaufmann Publishers Inc., 1996.

76. Subramanian, A., H. Kuehn, J. Gould, P. Tamayo, and J. P. Mesirov, "Gsea-p: a desktop application for gene set enrichment analysis," *Bioinformatics*, Vol. 23, no. 23, pp. 3251–3253, 2007.

77. Goeman, J. J., S. A. Van De Geer, F. De Kort, and H. C. Van Houwelingen, "A global test for groups of genes: testing association with a clinical outcome," *Bioinformatics*, Vol. 20, no. 1, pp. 93–99, 2004.

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

79. Perroud, B., J. Lee, N. Valkova, A. Dhirapong, P.-Y. Lin, O. Fiehn, D. Kültz, and R. H. Weiss, "Pathway analysis of kidney cancer using proteomics and metabolic profiling," *Molecular Cancer*, Vol. 5, no. 1, p. 64, 2006.

80. Gatti, D., W. Barry, A. Nobel, I. Rusyn, and F. Wright, "Heading down the wrong pathway: on the influence of correlation within gene sets," *BMC Genomics*, Vol. 11, no. 1, p. 574, 2010.

81. Liu, Q., I. Dinu, A. Adewale, J. Potter, and Y. Yasui, "Comparative evaluation of gene-set analysis methods," *BMC Bioinformatics*, Vol. 8, no. 1, p. 431, 2007.

82. Mansmann, U., and R. Meister, "Testing differential gene expression in functional groups. goeman's global test versus an ancova approach," *Methods Inf Med*, Vol. 44, no. 3, pp. 449–453, 2005.

83. Creighton, C., S. Hanash, and D. Beer, "Gene expression patterns define pathways correlated with loss of differentiation in lung adenocarcinomas," *FEBS letters*, Vol. 540, no. 1, pp. 167–170, 2003.

84. Troyanskaya, O. G., K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae)„" *Proceedings of the National Academy of Sciences*, Vol. 100, no. 14, pp. 8348–8353, 2003.

85. Su, A. I., T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, D. Block, J. Zhang, R. Soden, M. Hayakawa, G. Kreiman, *et al.*, "A gene atlas of the mouse and human protein-encoding transcriptomes," *Proceedings of the National Academy of Sciences*, Vol. 101, no. 16, pp. 6062–6067, 2004.

86. Kanehisa, M., S. Goto, Y. Sato, M. Furumichi, and M. Tanabe, "Kegg for integration and interpretation of large-scale molecular data sets," *Nucleic Acids Research*, Vol. 40, no. D1, pp. D109–D114, 2012.

87. Stark, C., 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*, Vol. 39, no. suppl 1, pp. D698–D704, 2011.

88. Scutari, M., "Learning bayesian networks with the bnlearn r package," *Journal of Statistical Software*, Vol. 35, no. 3, pp. 1–22, 2010.

89. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference*, San Francisco: Morgan Kaufmann, 1988.

90. Nagarajan, R., S. Datta, M. Scutari, M. L. Beggs, G. T. Nolen, and C. A. Peterson, "Functional relationships between genes associated with differentiation potential of aged myogenic progenitors," *Frontiers in Physiology*, Vol. 1, pp. 1–8, 2010.

91. da Piedade, I., M.-H. E. Tang, and O. Elemento, "DISPARE: DIScriminative PAttern REfinement for position weight matrices.," *BMC Bioinformatics*, Vol. 10, p. 388, 2009.

92. Li, Y., L. Liu, X. Bai, H. Cai, W. Ji, D. Guo, and Y. Zhu, "Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks," *BMC Bioinformatics*, Vol. 11, p. 520, 2010.

93. Kort, E. J., L. Farber, M. Tretiakova, D. Petillo, K. A. Furge, X. J. Yang, A. Cornelius, and B. T. Teh, "The e2f3-oncomir-1 axis is activated in wilms' tumor," *Cancer Research*, Vol. 68, no. 11, pp. 4034–4038, 2008.

94. Koeman, J. M., R. C. Russell, M.-H. Tan, D. Petillo, M. Westphal, K. Koelzer, J. L. Metcalf, Z. Zhang, D. Matsuda, K. J. Dykema, *et al.*, "Somatic pairing of chromosome 19 in renal oncocytoma is associated with deregulated elgn2-mediated oxygen-sensing response," *PLoS Genetics*, Vol. 4, no. 9, p. e1000176, 2008.

95. Friedman, N., and I. Nachman, "Gaussian process networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 211–219, Morgan Kaufmann Publishers Inc., 2000.