DISCOVERING A GENE INTERACTION ATLAS USING BAYESIAN NETWORKS AND EXTERNAL BIOLOGICAL KNOWLEDGE

by

Haluk Doğan

B.S., Computer Science, İstanbul Bilgi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2013

ACKNOWLEDGEMENTS

I wish to thank my advisers and professors for all the excellent assistance with which they provided me throughout my degree work, including my dissertation. It was a privilege to work with such consummate professionals. I am confident that the educational experience which I had at Boğaziçi University will hold me in good stead in my continuing studies towards the PhD.

I particularly want to thank my advisers Hasan H. Otu and Arzucan Özgür. Without their support, I can't accomplish this dissertation.

ABSTRACT

DISCOVERING A GENE INTERACTION ATLAS USING BAYESIAN NETWORKS AND EXTERNAL BIOLOGICAL KNOWLEDGE

Recent advances have enlightened that biological pathways are far more complicated than once thought, due to the inclusion of interconnected complex cellular actions, which made hard understanding the multifaceted mechanisms behind the biological phenomena. As a panacea, the bioinformatics community has brought up the modularity concept to ease the understanding of biological ground truth. A microarray is a high-throughput technology, which provides a global view of the genome in a single experiment with a systematic manner by enabling the analysis of the expression levels of a large number of genes simultaneously. Bayesian networks are probabilistic graphical models, which are well proven technique to infer gene regulatory networks from microarray data because of their ability to incorporate prior knowledge. In this study, we present an algorithm, called BNP, to infer biological pathways. Fortifying the results obtained by our model and exploring the novel interactions between genes, we construct a gene interaction atlas via Bayesian networks by incorporating external biological knowledge. Furthermore, a comparison of our methodology with the FLAT method, which does not use any external knowledge, shows that BNP outperforms it in all simulations.

ÖZET

BAYES AĞLARI VE HARİCİ BİYOLOJİK BİLGİLER KULLANARAK GEN ETKİLEŞİM ATLASI ÇIKARIMI

Son ilerlemeler ışığa çıkardı ki, birbirleriyle bağlantılı karmaşık hücresel olayları içeren biyolojik patikalar düşünüldüğünden çok daha karmaşık yapılıdırlar. Bu zor durum biyolojik fenomenin ardındaki çok dallı mekanizmayı anlamayı zorlaştırmaktadır. Buna çözüm olarak biyoenformatik dünyası modülerlik kavramını biyolojik gerçekliklerin anlaşılmasını kolaylaştırmak için öne sürmektedir. Yüksek işlem hacimli veri teknolojilerinden biri olan mikrodiziler güçlü bir araç olup sistematik tarzda tek bir deneydeki genomun genel bir tasvirinin sağlanmasının yanı sıra, gen türleri ve gen kavramlarının paralel analizi için tasarlanmıştır. Öncelikli bilgi birleşiminin yanı sıra yetersiz örnek sayısı ve deneysel hatalar ile başa çıkma yeteneklerinden dolayı, olasılıklı grafik modeli olan Bayes ağları, mikrodizi verilerinden gen düzenleyici ağ oluşturmada kendisini iyi kanıtlamış bir tekniktir. Bu çalışmada BNP adı ile sunduğumuz algoritma, seçilen biyolojik patikaların ilişkilerine derinlemesine bir anlam kazandırır. Modelimizden alınan sonucu güçlendirmek ve genler arasındaki yeni etkileşimleri keşfetmek için, harici biyolojik bilgiyi dahil ederek Bayes ağları üzerinden bir gen etkileşimi atlası inşa ettik. Ayrıca FLAT adındaki, harici bilgi kullanmaksızın yapılan hesaplamalarla, kendi metodolojimizi karşılaştırdık. Tüm simülasyonların sonuçlarına göre BNP'nin FLAT'tan daha iyi performans gösterdiğini gördük.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$1_{(\cdot)}$	Indicator function
i	Index for counting nodes
k_i	The degree of any node
q_i	The number of different states of node's parents
r_i	The set of values a node can take on
s_{ijk}	The total number of times in the sample
A	Random variable
AG	Adjacency matrix of candidate graph G
B	The prior information matrix
C	Scaling constant
C_i	Clustering coefficient of node i
D	Data
$Dir(\theta_{jn}\|j_{mn1},\ldots,j_{mnr})$	Dirichlet distribution
E(G)	Total energy of G
L_{ij}	The shortest path length between node i and j
N	Number of nodes in G
N_{jmn}	Number of samples
N_{ij}	Sum of corresponding Dirichlet distribution distribution hyper-parameters
$P(D\|\theta)$	The joint likelihood of training data
P(G)	Informative structure prior
$P(\theta)$	A prior distribution
$P(\theta_G\|G)$	Global parameter independence
$P(\theta_i\|G)$	Local parameter independence
X_i	Node
α_{ij}	Adjacency matrix element
α_{ijk}	Dirichlet distribution hyper-parameters
β	Hyper-parameter

θ	Represents model inferred from the Bayesian Network graph
Γ	Gamma function

LIST OF ACRONYMS/ABBREVIATIONS

A	Adenine
AUC	Area Under Curve
BDe	Bayesian Dirichlet Equivalent
BN	Bayesian Network
BNP	Bayesian Network Prior
C	Cytosine
CPTs	Conditional Probability Tables
DAG	Directed Acyclic Graph
DNA	Deoxyribonucleic acid
EM	Expectation Maximization
G	Guanine
GenMAPP	Gene Map Annotator and Pathway Profiler
GI	Gene Interaction
GO	Gene Ontology
GRNs	Gene Regulatory Networks
KEGG	Kyoto Encyclopedia of Genes and Genomes
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
NCI	National Cancer Institute
NP	Nondeterministic Polynomial
PPI	Protein-Protein Interaction
RNA	Ribonucleic acid
ROC	Receiver Operating Characteristic
T	Thymine
U	Uracil
cDNA	complementary DNA

mRNA	Messenger RNA

1. INTRODUCTION

1.1. Motivation and Scope of the Thesis

A biological pathway is a series of actions between molecules in a cell that results in a certain product or a change in a cell. There are many types of biological pathways. Some of the most common ones are involved in metabolism, the regulation of genes and the transmission of signals. Recent advances have shown that biological pathways are far more complicated than once thought [1]. Most pathways do not start and end at certain points. In fact, many pathways have no real boundaries, and they often work together to accomplish certain tasks. Biological pathways are discovered through laboratory studies of cultured cells, bacteria, fruit flies, mice and other organisms. Many of the pathways identified in these model systems are the same or have similar counterparts in humans. It is estimated that there are still many biological pathways that remain to be found. Identifying and understanding the complex connections among the molecules in the biological pathways, as well as understanding how these pathways work together is an important problem in biology [2].

The rapid growth of the biological technologies, such as the mapping of human genome, has resulted in massive amounts of data. As in many fields, the challenge has shifted from collecting a sufficient amount of data to understanding and gaining insight from the huge amount of data that are already available. Recent advances in high-throughput technologies have given rise to large-scale biological data in the form of expression profiles of tens of thousands of genes and proteins.

Microarray is a robust and high-throughput technique, which was designed for parallel analysis of genotypes and gene expressions and provides a global view of the genome in a single experiment with a systematic and comprehensive manner. Recently, the microarray technology has become a widely-used platform to find solutions for a wide spectrum of biological and clinical problems [3].

Pathway and functional analysis is an indispensable part of microarray data analysis. GenMAPP, KEGG pathway information and Gene Ontology (GO) terms are included in many probe annotations by the microarray chip producer. Linking gene expression data to such pathway and functional information brings out valuable expression and regulation patterns. Initially single gene analysis was widely used to analyze microarray data. This approach mainly focused on finding a list of interesting (e.g. differentially expressed) genes. However, due to the limitations of single gene analysis, which did not look into the functional or relational aspects of gene lists, scientists became interested in examining the relations between known pathways and outcomes such as the way of processing drugs or fertilization of the egg [4].

Gene regulatory networks (GRNs) are structured sets of complex information which represent regulatory relationships between genes. Gene regulatory networks play a critical role in determining cellular functions and are of interest for identifying various disease factors. Formal understanding of gene regulation is an emerging field in systems biology. The inference of gene regulation networks from high-throughput biological data is an important and challenging task. Interactions between genes can be identified from gene expression data using reverse engineering methods. To date, many reverse engineering methods have been proposed. Some of the methods are Bayesian networks (BN) [5], Boolean networks [6], and linear and non-linear differential equations [7].

Bayesian network is a promising method to describe relationships between genes in gene regulatory networks [5]. Learning BNs from observed data is a popular research field which is based on search and score based techniques. A scoring mechanism is used for identifying the most probable a posteriori network given the data and a priori knowledge. Generally, the proposed algorithms to learn BNs are heuristic and attempt to maximize the score function. Previously, proposed BN learning algorithms have been only dependent on observed data such as microarray data in the case of inferring gene regulatory networks. However, studies have shown that incorporating prior knowledge to score the candidate networks has significantly increased the success of inferring mechanisms [8].

Real networks are heterogeneous in their degree distributions and have organizations that diverge from the random network models. Community structures, or modules, which represent tightly knit nodes with similar properties/roles, are commonly observed in real networks [9, 10]. This community structure often represents a hierarchical pattern, similar to the organization of the human body where cells make up tissues, tissues make up organs, and organs make up systems. Therefore, in building real networks, such as gene interaction networks, one plausible approach is first to identify the modules that exist in the final graph, and then to identify the structural organization of these modules to establish the final graph [11].

Biological networks are complex in their nature. However, they can be separated into subunits, namely modules, which are relatively autonomous with respect to other subunits. The idea of modularity started to gradually gain attention in systems biology, molecular biology, developmental biology, and evolutionary biology [12]. Modularity brings us to understand the latent structure in complex networks, identify heterogeneities, and predict missing links between nodes. Modules of biological networks are associated with certain biological processes. Therefore, determining the interaction of biological modules is of great interest to understand how organisms and their subunits function together.

Modular network learning has been applied to biological networks mostly in identifying protein-protein interaction (PPI) networks. Signaling proteins of the yeast Saccharomyces cerevisiae participating in shaping the organism into a filamentous form were represented in the form of an interaction network yielding separate clusters with edges between clusters as important points of communication [13]. Spirin and Mirny applied clique detection and superparamagnetic clustering to the yeast proteome identifying modules with significant occurrence when compared to random graphs with the same degree distribution [14]. The resulting modules consisted of known protein complexes and were composed of proteins with the same or similar biological functions. Another study on estimating the PPI network in yeast incorporated the microarray expression profiles and used the resulting network to predict the functions of genes that have not yet been well characterized [15]. Similar analyses have been done in hu-

man and other organisms [16] where the topological parameters such as the clustering coefficient and link density of the modules have been shown to be a good indicator of biological homogeneity in the module. Applications of modular learning to metabolic and gene interaction networks have resulted in similar findings with an additional necessity citing that the size of the data and the large number of nodes size hamper learning the complete network [17–19].

This thesis proposes a method for an exhaustive inference of gene regulatory networks from microarray data as well as for the construction of an atlas to depict relationships in a group of genes by means of Bayesian networks. In achieving this task, we utilize external biological knowledge to guide the construction of module networks that make up the interaction atlas. In our proposed model, a module is a set of genes, which have the same expression profile and share similar biological functions. Our primary reason for using module networks in our model is that, module networks are mostly applicable for larger networks, are error tolerant, and enable inferring functionally coherent modules [20]. Chapter 3 explains the proposed workflow in detail.

1.2. Biological Background

Proteins are essential macromolecules and play a very important role in the human body. Proteins are composed of amino-acids, and amino-acids are encoded using 3 out of 4 types of nucleotides (Adenine, Guanine, Cytosine, Uracil). The chain of nucleotides forms the "Deoxyribonucleic acid" (DNA), and the information needed to produce proteins is stored in DNA.

DNA is comprised of a chain of nucleotides. There are four different nucleotides, Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). DNA is in the well-known "double helix" form, and is stored in a single chain for prokaryotes, while in eukaryotes DNA can be packed in individual units called chromosomes. In the double helix form Adenine binds only with Thymine, while Guanine binds only with Cytosine. This binding plays critical role during the replication and gene expression processes.

Unlike the DNA molecule, another important molecule called "Ribonucleic acid" (RNA) is single stranded and is made up of the same nucleotides as the DNA but includes Uracil (U) instead of Thymine. Even though, RNA has multiple functions, it is primarily an intermediate molecule to produce protein from DNA.

1.2.1. Protein Synthesis

Genes, which are the hereditary units for the living organisms, are continuous stretches of DNA molecules. They are responsible for producing particular protein molecules. The "Central Dogma" theory pinpoints the roles of DNA and amino-acid molecules in protein synthesis. Although there are some exceptions, the framework of this complicated theory is that "DNA makes RNA makes protein". Under the light of this theory, synthesizing proteins in eukaryotes can be distinguished into three principal stages: transcription, splicing and translation.

- Transcription: This process is carried out for the purpose of transferring the information from a section of DNA to a pre-mRNA (immature single strand of mRNA). The transferred information is read only from one strand of DNA.
- Splicing: The pre-mRNA is comprised of introns and exons. Introns are removed in this stage and the remaining exons that are attached back to back represent the coding sequences of the genes, which are used for protein production. At the end of this phase, the mature mRNA is produced and passed on to the ribosomes.
- Translation: This last step is directly related to the gene expression. Based on the sequence of the mRNA, amino-acids are joined together, which leads to produce proteins.

Gene expression analysis deals with assessing the types and quantities of mRNA molecules in a biological sample at a given time. Although not directly related to the protein profile of the cell, gene expression, also called transcriptomics, provides valuable information about the functional state of a cell.

1.3. Microarrays

The advent of high-throughput technologies has reshaped the way researchers analyze biological problems by generating hypotheses based upon genome wide integrative reductionist queries [21]. One of the popular high-throughput technologies, the microarray technology, enables us to measure the expression of thousands of genes of organisms by conducting a single experiment. This technology produces massive amounts of data and brings with itself new computational and statistical challenges. A typical microarray experiment can play distinctive roles in identifying differentially expressed genes, predicting treatment response, discovering population-specific diseases, and determining gene interactions by pathway analysis.

DNA hybridization is the key component of microarray experiments to measure the abundance of mRNA, which states the gene expression level. The amount of mRNA can be used to understand the activities of certain genes under different circumstances. A typical microarray has thousands of cells over its surface and these cells are called probes. Each probe is designed as complementary to known genes for specified organisms so that cDNA strands, obtained from mRNAs can bind to these specified probes. Figure 1.1 depicts the principal steps used in the microarray technology.

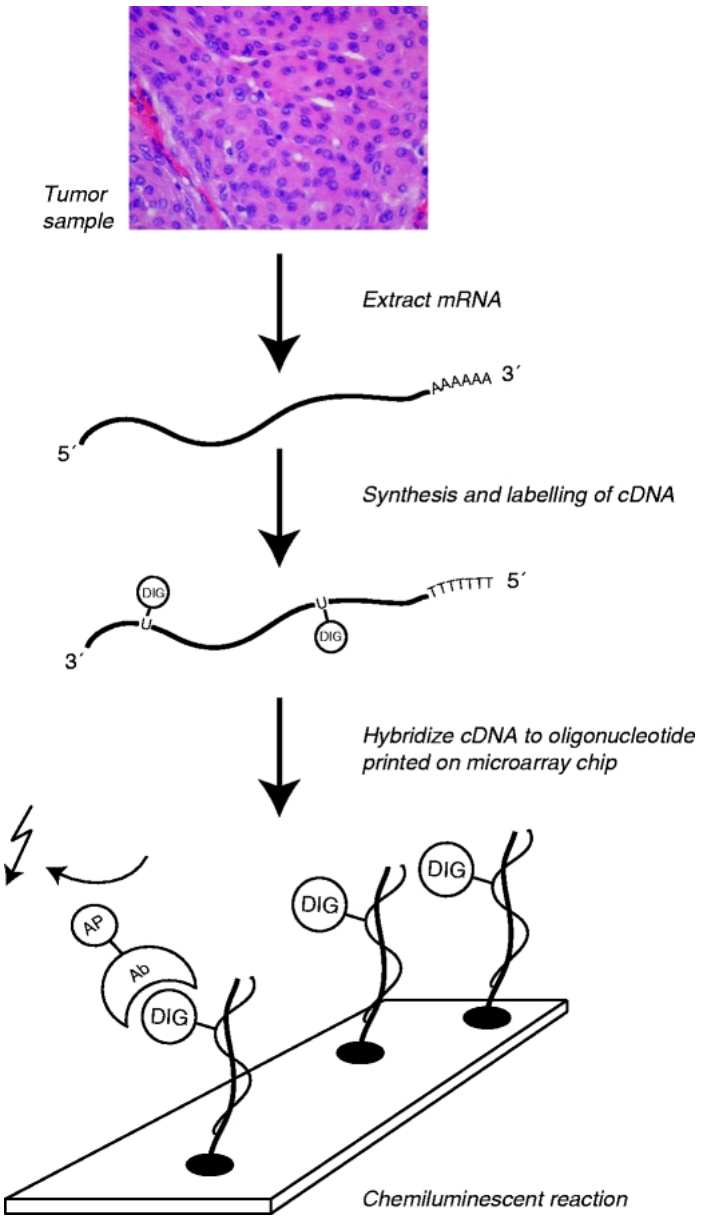

Figure 1.1. Principle Steps in Microarray Experiments [22].

There are two types of extensively used microarrays, namely cDNA and oligonucleotide chips. The main differences of microarray types rely on the design of the probes.

- cDNA microarrays: Each probe represents only one gene. There are two biological samples flown in two different channels that hybridize to the same surface.

- Oligonucleotide chips: Affymetrix is one of the leading companies of this chip design. A gene is represented by more than one probe. One biological sample is used for one chip.

1.3.1. Expression Measures

After the DNA hybridization step, microarrays quantify gene expression levels using fluorescence intensity. Fluorescently lit spots on the probes indicate the identity of the targets and intensity of the signal, which is associated with the amount of the target mRNA. The brightness level of each probe is positively correlated with the abundance of the corresponding gene. The intensities are measured from the image obtained by scanning the microarray surface. These outputs usually contain noise due to experimental errors and random factors, which are dealt with specialized image processing algorithms. An experimental design utilizing microarrays usually contain many chips (e.g. chips run for N cancer and M normal samples) and the measurements obtained from each chip must be made comparable before any downstream analysis. This procedure, referred to as normalization, can be based on different approaches such as using housekeeping genes or setting the overall average of the chips to a certain given value. The former depends on the assumption that there exists some genes whose abundance should be the same across different biological or clinical states and the latter assumes that the starting biological material contains roughly the same amount of mRNA (despite originating from different biological conditions) and thus the overall intensities must be roughly the same.

To this end, we have a data matrix representing the numeric relative gene expression values associating each gene and sample. These normalized gene expression levels are usually subject to the following analysis modules:

- Filtering: In this step genes that do not have reliable detection levels and/or genes with expression levels that are very similar in different states (e.g. cancer and normal) are eliminated.
- Differential gene expression: Genes that are significantly differentially expressed

between different stated are found. Generally statistical tests such as student's t-test or "Mann-Whitney U" test are employed. This step is usually corrected for multiple hypothesis testing.

- Clustering: Samples or genes are clustered to identify similar phenotypes and genes that have concurrent expression levels across samples, which may imply certain functional processes.
- Biomarker Discovery: A prediction algorithm is devised to identify a relatively small number of genes that can predict samples that belong to a certain phenotype. For example, if a microarray experiment utilizes primary and metastatic tumors, the goal may not only be to find genes differentially expressed between the two states but also to identify a signature set of genes and predict a new sample as primary or metastatic sample.
- Functional or Pathway analysis: Genes that belong to a certain gene set (GO functional category or pathway) are identified and the sets are indicated as being significantly enriched or not.
- Promoter region analysis: Differentially expressed genes are investigated for their promoter regions to identify certain motifs that exist in an abundance that is beyond random chance. This way transcription factors or other molecular mechanisms that orchestrate the observed genes' expression profiles are found.

2. GENE NETWORK MODELING

2.1. Gene Networks

The success of genome sequencing has brought out characterization of hundreds of thousands of genes. The next important task is identifying functions of genes and their interactions. Therefore, learning gene interaction networks using computational and statistical methods is an appealing research area. The terms "gene interaction networks" and "gene regulatory networks" are interchangeably used throughout this thesis.

Gene regulatory networks have been represented by various models. Boolean networks, Petri Net modeling, state machines, π-calculus, and Bayesian networks have been applied to infer gene regulatory networks from microarray data in previous studies [5,23–26]. In this thesis, we have used "Bayesian network" modeling in our methodology. The reasons for choosing Bayesian networks are multiple. Firstly, Bayesian networks is a concrete class of models which provide us to apply our ideas and to assess the results of queries for our inquiries. Secondly, Bayesian networks function at the intersection probability and graph theory. Since we are studying on noisy microarray data and uncertain biological evidence, probability theory is a solid way to interpret our findings. Last but not the least; we have been trying to detect cause-effect relationships from data.

Bayesian networks have become a widely used method for modeling uncertain knowledge. They have become a promising tool for analyzing gene expression patterns [5]. Bayesian networks can be used to model relationships between genes and genetic regulatory networks. The advantages of using Bayesian networks are as follows:

- Compact and intuitive representation of gene relationships.
- Capturing causal relationships between genes.
- Integration of prior knowledge into the network.

- Suitable to work well with noisy data.
- Capable of handle uncertainty.
- Efficiently learn new models of gene relationships.

2.2. Bayesian Network Models

A Bayesian network is a graphical representation of probability distributions over a set of parameters. Its structure is in the form of a directed acyclic graph (**DAG**); each node represents a random variable, and each edge represents dependencies among random variables. If there is a directed edge from node X_i to X_j, formally shown as $X_i \rightarrow X_j$, then X_i is called parent of X_j, and X_j is called a child of X_i. The intuitive meaning of an edge from X_i to X_j is that X_i directly influences X_j. Local probability distributions of each node only depends on parameters of their parents. Formally, each variable X is independent of its non-descendants given its parents. A Bayesian network provides a factored representation of the joint probability distributions.

The Bayesian network structure in Figure 2.1 dictates that the random variable C depends on variables A and B, whereas, A and B are independent from each other. Another rigorous implication of this Bayesian network is the independence of A and D on the condition that the value of C is fixed.

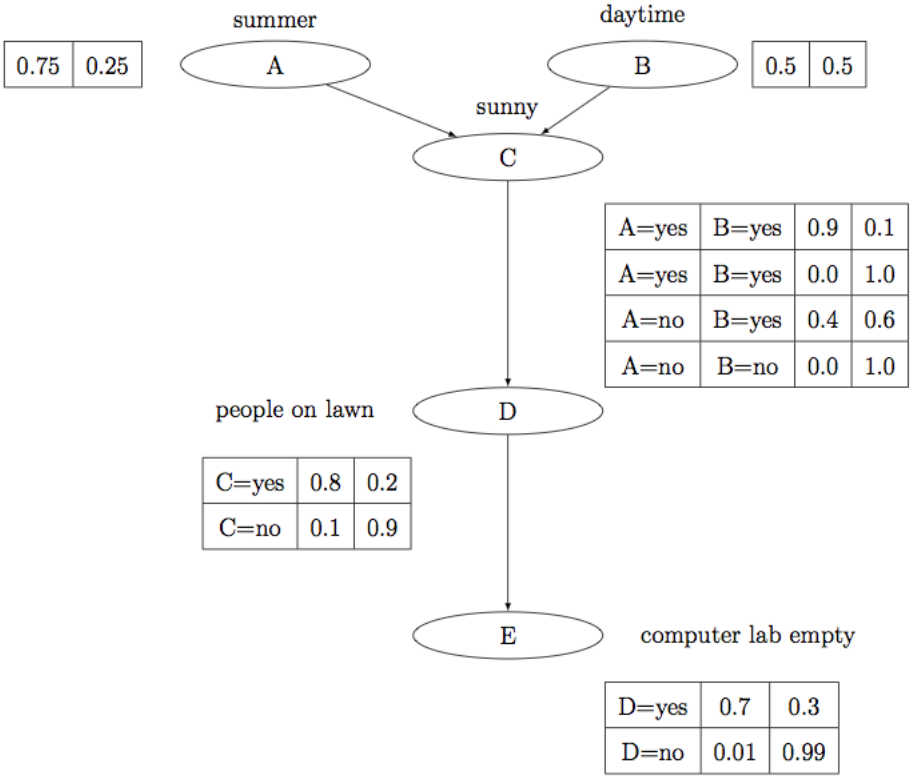

Figure 2.1. An Example Bayesian Network Modeling.

The structure of a Bayesian network implies statistical conditional independence statements, as well as statements of dependence among random variables. We can show the set of all independence relations implied by Bayesian networks from a set of axioms described in [27]. However, determining these relations cannot be an easy task, because it proceeds over and over again until the desired relation is proved or disproved [28]. The method "d-separation" is another approach to determine independence relations from the structures of Bayesian networks. The d-separation rules provide an easier method than the use of the set of axioms, which are described in Pearl et al. Therefore, d-separations rules are widely used in practice [29].

2.3. Learning Bayesian Networks

Learning the structure of BNs from data is one of the most challenging problems, even if data are complete. The problem is known to be NP-hard, and best exact known methods take exponential time on the number of variables and are applicable to small settings (around 30 variables) [30]. In the following subsections, we briefly explain learning the parameters and structure of Bayesian networks. There are two types of algorithms used to learn the structure of Bayesian networks: "score based" and "constrained based" methods.

2.3.1. Learning Parameters

Parameter learning is the indispensable phase to learn the structure of a network given the data. Bayesian networks implement statistics of the study by defining a probability distribution for each node. Based on the Bayes rule, we use the known parameters to update our beliefs into posterior beliefs:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

where D represents the data and θ represents the model inferred from the BN graph. Since $P(D)$ is the same for all different parameters, it can be left out from the above equation, rendering the following approximation:

$$P(\theta|D) \propto P(\theta)P(D|\theta)$$

Typically, parameter learning methods estimate the values of the parameters of a Bayesian network by the following two different methodologies:

- Maximizing the joint likelihood of the training data.
- Computing the posterior over parameters θ given a prior distribution $P(\theta)$.

The first method is regarded as maximum likelihood estimation (MLE). The joint log likelihood of the training data can be written in a discrete form as follows:

$$\begin{aligned}\log P(D|\theta) &= \sum_{i=1}^{N}\sum_{j=1}^{n}\log P(x_j^i|x_{\Pi(j)}^i,\theta)\\ &= \sum_{i=1}^{N}\sum_{j=1}^{n}\log\left(\prod_{mn}\theta_{jmn}^{1_{x_j^i=m,x_{\Pi(j)}^i=n}}\right)\\ &= \sum_{jmn}N_{jmn}\log\theta_{jmn}\end{aligned}$$

where N_{jmn} is the number of samples which satisfy the configuration $[x_j = m, x_{\Pi(j)} = b]$. M_{jmn} can be shown mathematically as follows:

$$M_{jmn} = \sum_{i=1}^{N} 1_{x_j^i=m,x_{\Pi(j)}^i=n}$$

where, $1_{(.)}$ denotes the indicator function. Using Lagrange multipliers to maximize local normalization constraints, one can maximize $\log(P|\theta)$ by the following closed form:

$$\hat{\theta}_{jmn} = \frac{N_{jmn}}{N_{jb}}$$

where $N_{jn} = \sum_a N_{jmn}$.

Maximum a posteriori estimation is the second method to learn parameters from data. We can use the posterior distribution to test goodness of fit of both prior and likelihood (probability distribution of parameters), and the main advantage of Bayesian networks is the ability of incorporating prior knowledge. Prior knowledge can be beneficial to avoid overfitting when one has a small size training data. Given a prior distribution $P(\theta)$, we can learn the parameters by maximizing the posterior. It is mathematically shown as follows:

$$\log P(\theta|D) = \log\frac{P(\theta)P(D|\theta)}{P(D)}$$

The above equation is equivalent to maximizing $\log(P(\theta)P(D|\theta))$ since $P(D)$ is same for all different parameters θ.

Given a directed acyclic graph G such that $P(G) > 0$, we have two parameter independence assumptions, which are satisfied by the Bayesian network:

- Global parameter independence: $P(\Theta_G|G) = \prod_{i=1}^{n} P(\Theta_i|G)$
- Local parameter independence: $P(\Theta_i|G) = \prod_{j=1}^{q_i} P(\Theta_{ij}|G)$ for all $i = 1, \ldots, n$.

The above two parameters independences imply that the prior $P(\theta)$ yields the following:

$$P(\theta) = \prod_{jn} P(\theta_{jn})$$

By assumption, $P(\theta_{jn})$ is coming from Dirichlet distribution. That is;

$$Dir(\theta_{jn}|j_{mn1}, j_{mn2}, \ldots, j_{mnr}) = \frac{1}{Z} \prod_{k=1}^{r} {\theta_{mnk}}^{j_{mnk}-1}$$

where $Z = \frac{\prod_{k=1}^{r} \Gamma(j_{mn})}{\Gamma\left(\sum_{k=1}^{r} j_{mnk}\right)}$ is a normalization factor, and Γ is the gamma function.

For practical purposes, by preferring to use Maximum-Likelihood parameters instead of entire distribution, the MAP for θ_{mnk} will be in the following form:

$$\hat{\theta}_{jmn} = \frac{\alpha_{jmn} + N_{jmn}}{\alpha_{jn} + N_n}$$

Bayesian learning is another alternative method to estimate parameters as in ML and MAP. Unlike ML and MAP, in Bayesian learning, one can estimate all parameter possibilities by storing the posterior parameter distribution for subsequent use. This

is mathematically shown as follows:

$$P(\theta|D) = \prod_{j=1}^{r} \prod_{n} \Gamma(\alpha_{jn} + N_{jn}) \prod_{m} \frac{\theta_{jmn}^{\alpha_{jmn}+N_{mn}-1}}{\Gamma\left(\alpha_{jmn} + N_{jmn}\right)}$$

We can obtain the closed form formulas, as shown above, regardless of using ML, MAP or Bayesian learning for the complete dataset. However, when the data is incomplete, parameter learning becomes difficult due to the reason that log likelihood cannot be discretized to estimate parameters for each variable independently. If the missing data is always unobserved, we can label it as a hidden variable. The EM (Expectation Maximization) algorithm can be used to avoid the missing data problem. However, the details of the EM algorithm are beyond the scope of this thesis.

2.3.2. Learning the Structure via Score-Based Methods

Score based methods are popular for learning Bayesian network structures from data. In each iteration of the score-based algorithms, a score is assigned to a candidate network to state how well the candidate network describes the data. For discrete BNs, most of the learning tasks are performed by calculating $P(D|G)$ with the Bayesian Dirichlet equivalent (BDe) score function, instead of the true parameter $P(G|D)$.

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

where $P(D|G)$ is the marginal likelihood of data D given graph G, $P(D)$ is the probability of the data, $P(G|D)$ is the posterior probability of G.

$$\begin{aligned} P(G|D) \propto P(D|G) &= BayesianScoring_{BD_e}(D, G) \\ &= \prod_{i=1}^{N} \prod_{j=1}^{q_j} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \prod_{k=1}^{r_j} \frac{\Gamma(\alpha_{ijk} + s_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

where N_i 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 α_{ijk}, which need to be assigned by the user. M_{ij} is the

number of times that the parents of node i take on configuration j in the dataset. And 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.

In the score based methods, a heuristic algorithm is employed to walk through the search space. At each iteration, the score of the candidate graph is assessed and the overall procedure is halted when certain optimization criteria are met. Among the most popular learning tasks is the Greedy Hill Climbing algorithm [31], which is employed in this thesis.

3. ATLAS GENERATION

Pathway analysis provides an insight into the underlying biology of differentially expressed genes becoming an indispensable stage for microarray data analysis. However, there may exist undiscovered biological pathways, known pathways may have missing nodes and/or edges and interaction among pathways may reveal new biological hypothesis. In order to provide improvement in these areas, in this thesis, we provide a methodology that generates a gene interaction atlas from microarray data. The construction of a gene regulatory network from massive amounts of microarray data could be cumbersome due to the limitations of computational power and the structure learning algorithms performances. For this reason, heuristic algorithms and module networks have emerged as means to reduce computational limitations. Moreover, modularity would be beneficial to bring out the functional relationships between strongly connected subunits in the network. In Figure 3.1, we depict our overall workflow.

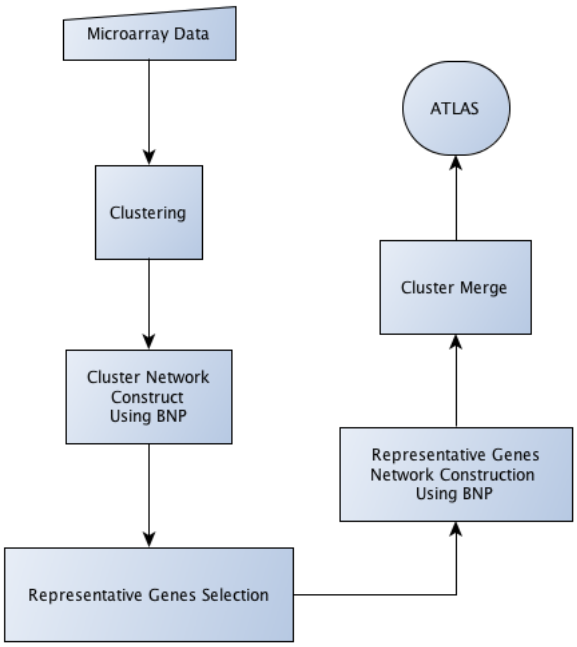

Figure 3.1. A Workflow of the Gene Interaction Atlas Generation.

In the proposed method, gene expression profiles are clustered to identify the modules and the genes that they contain. A Bayesian network Learning approach utilizing external knowledge (called the BNP algorithm) is applied to each module. These networks define the connectivity between the genes in each module. In order to combine these different clusters (the terms modules and clusters are interchangeably used throughout this thesis), we identified representative genes for each cluster with two different approaches. The first one is based on the nodes' out-degrees, and the second one is based on the Pagerank algorithm. Subsequently, we apply the BNP algorithm again, only to the representative genes to understand the relations among the clusters. Finally, we apply the BNP algorithm once again to merge clusters if two clusters have been linked in the previous step.

3.1. Clustering

Constructing a gene interaction atlas by inferring gene regulatory network from high throughput biological data such as microarray data is a computationally intensive task due to the following reasons:

- Thousands of genes interact with each other.
- Complex relationships exists between genes.
- The data is noisy.
- The sample size is in general inadequate.

Learning the structure of BNs from data is a computationally intensive task and the algorithms proposed so far are only applicable for small number of variables [30]. Currently, most of the common structure learning algorithms are based on the greedy search algorithms, the Hill climbing algorithms and the Markov Chain Monte Carlo (MCMC) search techniques [32]. Module networks approach is a promising technique that can cope with the aforementioned computational and statistical problems. It explicitly partitions the variables into modules and each module represents a set of variables that have the same statistical behavior so that it significantly reduces the complexity of the network along with the number of parameters [19]. Splitting genes into modules

according to their expression profiles is an essential step for learning structure from gene expression data. Genes in each module are expected to have similar expression patterns and functional characteristics. Relations in the modules are well defined as co-expressed genes are expected to reside in the same module [19]. Clustering is a way to partition genes according to their expression profiles. The k-means clustering algorithm is a simple, but popular clustering algorithm. The k-means algorithm starts with a collection of n items, in our case gene expressions, and a chosen number of clusters, k, that we want to apportion the items into. Initially k items are selected randomly as the centroid of clusters. Then, n items are assigned to the closest cluster according to the specified distance metrics such as Euclidean distance. Subsequently, clustering proceeds repeatedly by the following three steps:

- The means of all items in each cluster are computed.
- New centroids are specified as the means of the clusters.
- Items are reassigned to the clusters according their distances to the new centroids.

Finally, the algorithm terminates when the assignment of items to clusters is not changed. In the proposed workflow, we adopted the k-means clustering algorithm to identify the genes in each module.

3.2. Incorporation of External Knowledge

The inference of gene interaction networks from high-throughput biological data is an important and challenging task in systems biology. Throughout the literature, the term "gene interaction" has been used in a broad sense implying direct or indirect interactions between genes and/or gene products. Several machine learning and statistical methods have been proposed for the problem [20,31,33–36] and Bayesian network models have gained popularity for the task of inferring gene networks [5, 37–39]. Because of the complexity of gene interaction networks and the sparse, noisy nature of experimental data, machine learning and statistical methods may lead to poor reconstruction accuracy for the underlying network. One way to overcome this problem would be to incorporate prior biological knowledge when making network inferences

using experimental data. Due to technological advances in sequencing and microarray technologies in proteomics and related fields, biological and clinical data are being produced at an ever increasing rate. The 2013 special database issue of the Nucleic Acids Research journal lists 1,512 molecular biology databases, which provide a vast amount of annotated data and meta data that could be used in a systematic way [40].

Many BN structure learning algorithms are based on heuristic search techniques with the likelihood approximation because of the infeasible computational complexity. These approaches may lead to a false model as neither the search technique nor the objective functions guarantee the optimal solution. Informative priors generated from existing biological information can improve structure learning to get better models to describe the underlying gene interactions. In several studies the use of prior biological knowledge in conjunction with gene expression data has been shown to improve the fidelity of network reconstruction [41–45]. 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 these approaches manual curation and/or manual incorporation of the external knowledge were employed. In this thesis, we use a framework to incorporate 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 the use of this information in BN modeling. The proposed method is fully automatic and does not use likelihood approximations to find the optimal network that explains the observed experimental data. The proposed framework uses the BN infrastructure itself to incorporate external biological knowledge when learning the networks. This infrastructure yields gene interaction information for pairs of genes, which can be used as informative priors to calculate the probability of a candidate graph, G. This information is then incorporated in the network learning process that tries to identify the most probable graph given the data. The schematic depiction of the overall proposed method is presented in Figure 3.2.

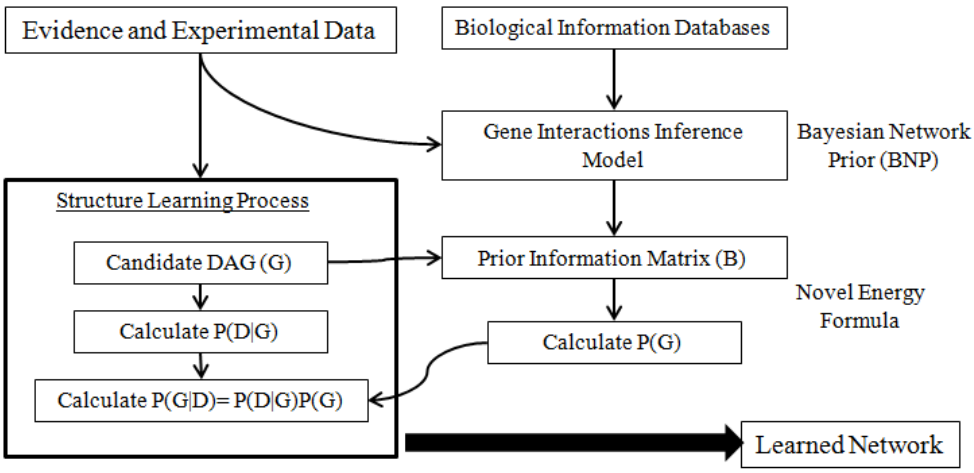

Figure 3.2. Overall workflow of the proposed method. BNP is constructed using gene interaction information from external biological databases and when instantiated with an evidence vector for a pair of genes, the gene interaction probability is inferred. For a list of genes, the pairwise interaction information is stored in the prior matrix B, which is used to calculate the probability of a candidate graph G in the structure learning process.

Pairwise interaction information is gathered from biological databases and a BN model for prior knowledge, Bayesian Network Prior (BNP) is developed. In BNP, one node is depicted as "Gene Interaction" (GI) and the topology represents the dependence structure among different evidence types, as well as dependence structure between them and the GI node. For a set of genes, the model is instantiated with the given evidence and/or experimental data for each pair of genes. The GI node is used to infer whether the gene pair is related or not, represented by a prediction value between 0 and 1. A prior knowledge matrix, B, is populated with these prediction values for all gene pairs. This prior knowledge is utilized to calculate the probability of a candidate DAG, G, in the structure learning process. This parameter is used to optimize $P(G|D)$ instead of the likelihood, $P(D|G)$, used by the existing structure learning algorithms.

3.2.1. Informative Structure Priors

In gene interaction network modeling studies using BNs, X_i represents a gene and the edges represent relationships among genes. The task of network inference (i.e. structure learning) is to make inferences regarding the graph G that best explains the data. This can be achieved by finding the DAG G that maximizes $P(G|D) = \frac{P(D|G)P(G)}{P(D)}$ where $P(D|G)$ is the likelihood, $P(D)$ is the probability of the data, $P(G)$ is the structure prior (or network prior) probability of the graph G, and $P(G|D)$ is the posterior probability of G. In commonly used heuristic structure learning algorithms, $P(D|G)$ is optimized instead of the true model $P(G|D)$. The likelihood criterion does not guarantee to find the optimum solution even if a heuristic approach is not employed. Nevertheless, optimizing the likelihood can be justified by assuming $P(D)$ and $P(G)$ to be equal for all G. The former assumption can be regarded as reasonable as D is observed. However, the latter assumption is generally not correct and is made mainly due to difficulties in calculating $P(G)$ and/or lack of prior knowledge on G. Use of uniform (flat) priors for Gs ignores the contribution of $P(G)$ and this may cause failure in differentiating between DAGs that are in the same Markov equivalence set. Therefore, the true DAG among the ones that support the same conditional probability distribution cannot be identified. BNP aims to calculate $P(G)$ using external knowledge and provide improvements in the structure learning phase for gene interaction networks.

For discrete BNs, most of the learning tasks are performed by calculating $P(D|G)$ with the Bayesian Dirichlet equivalent (BDe) scoring function and by assuming uniform (flat) prior structure for all possible candidate DAGs [46]. In the proposed approach, we employ a greedy search algorithm that aims to maximize $P(G|D)$. For a given candidate DAG, G, we calculate $P(G)$ by first obtaining the prior information matrix, B. Unlike existing methods, the proposed approach does not use categorized prior knowledge but assigns probabilities to each candidate edge. The matrix B is obtained by instantiating BNP with the evidence vector for each pair of genes in the input gene set. These evidence vectors can originate from any performed experimental data at hand, or external knowledge, or both.

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

$$E(G) = \sum_{i,j} \frac{U(i,j)}{N^2}$$

where N is the number of nodes in G. This way, we do not assign categorical values to $U(i,j)$ and exploit fully the information about prior existence of an edge. Informative structure prior is formulated as:

$$P(G) = Ce^{\beta E(G)}$$

where C is a scaling constant. The choice of C does not affect the relative comparison during scoring of graphs in structure learning. The hyperparameter β can be marginalized using the following equation:

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

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

3.3. Bayesian Network Prior

The goal in building BNP is to construct a framework such that the distilled external biological knowledge is used in an intelligent way to make an assessment about the interaction of a pair of genes. Previously, Troyanskaya et al. proposed a Bayesian framework for combining various data sources for gene function prediction [47]. In

this method a Naive Bayes 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 the bottom level nodes with evidence and the probability of the functional relationship between the two genes was updated. The model was designed for functional prediction, not for gene interaction network learning. The prior knowledge inference model used here automatically learns the parameters of the nodes in BNP that predicts if two genes interact using external biological knowledge. The model organism chosen for BNP was human and the external data came from pathway, microarray, gene and protein interaction databases. The assembled information source is made up of "evidence types", each making a "Yes" or "No" call about the interaction of two genes and BNP is the BN that represents the relation between these evidence types and the GI.

The data sources used in calculating BNP came from microarray coexpression, KEGG [48], NCI/NATURE [49], and Reactome [50] databases. After all sources were merged, 60,950 pairwise gene interactions based on 19 evidence types were obtained. The 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 [29].

3.3.1. Constructing the Bayesian Network Prior (BNP)

BNP was built using the gene interaction evidence matrix that contained over 60,000 pairs of genes. The model was trained and tested using a 5-fold cross validation approach, where the dataset was randomized and 80% of the data was used to train the model and 20% of the data was used to test the model. The success rate of the model with respect to the GI data label is calculated as the classification error, which is the percentage of mismatching real and predicted values of the GI node. This procedure was repeated 5 times and average error values were calculated. At each iteration, after BNP was built with 80% of the evidence matrix using the Greedy Hill Climbing method, the remaining 20% of the data matrix was tested by inferring the value of the

GI node. This test was done through instantiation of BNP using the evidence vector of a given pair of genes. Loopy Belief Propagation inference algorithm was used for inference. If the inference value was greater than 0.5, the GI node was taken to be "true". The classification error rate for the 5-fold cross validation was 0.105 ∓ 0.003 implying an accuracy of $\sim$90% when estimating if two genes interact given the external biological knowledge. The final BNP was constructed using the entire evidence matrix. The strength of the probabilistic relationships expressed by the edges of the BNP was measured using Friedman's bootstrap method with 1,000 repeats [51]. Model averaging was used to build a consensus DAG of BNP, which is shown in Figure 3.3. BNP provides a unique depiction about how different experimental assays are related to each other and to the event of gene interaction, which opens ways to new hypotheses about assay type interrelation.

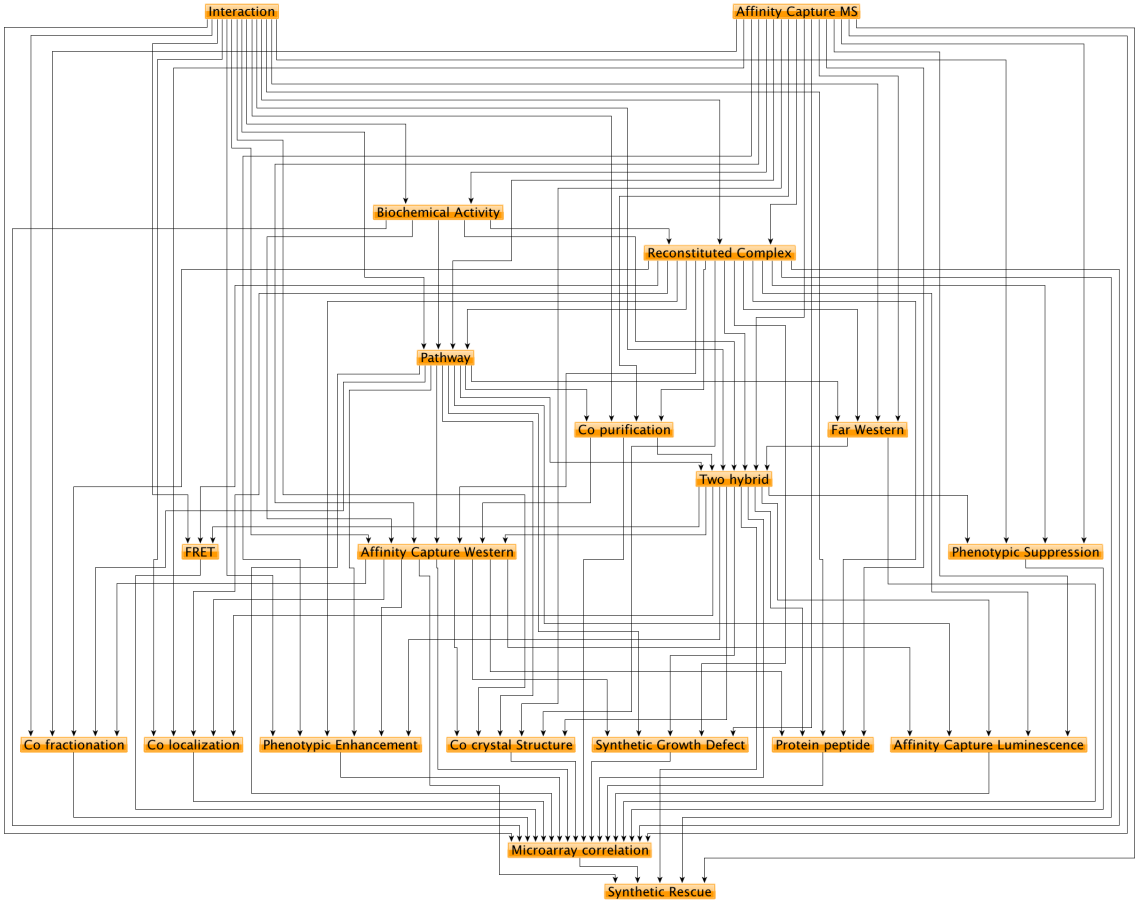

Figure 3.3. Topology of the Bayesian Network Prior (BNP). BNP depicts the conditional dependence structure between various evidence types and the Gene Interaction node based on external biological knowledge. BNP is used to predict the interaction probability for two genes using the provided experimental data combined with external information.

3.3.2. Application of BNP to Microarray Data

We currently use BNP on microarray data containing two types of samples (i.e. cancer vs. normal). The input data for structure learning was obtained as previously described in [52]. Briefly, columns represent genes 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. For example, if we have 10 control and 10 test samples,

the input matrix consisted of 100 observations (10 control x 10 test) and reflected the distribution of fold change values between the two classes of samples. This matrix was discretized into 3-levels using k-means clustering [53]. The inferred DAGs using prior knowledge (proposed method) and uniform prior knowledge (flat prior, standard methods) were compared to the original pathway structures using AUC values.

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, B. During the instantiation, the evidence vector used was composed of the existing evidence information for the gene pair in the databases and the microarray correlation value calculated by the input gene expression 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 the predicted and true DAGs for the 14 pathways using simulated gene expression data are assessed.

4. EXPERIMENTS AND RESULTS

Graphical representations of biological pathways show us vital biological and enzymatic reactions in a cell. There are two types of biological pathway analyses. In the first type of analysis, only one pathway is taken into account and its robustness, steady states, modular structure, and network motifs are examined. In the second type of analysis, multiple pathways are of interest to identify similarities between them. These similarities may instruct to rectify and explore new pathways or develop new drugs and determine missing enzymes [54].

Kyoto Encyclopedia of Genes and Genomes (KEGG) is a curated knowledge based biological pathways database, which stores molecular interaction networks in various cellular processes among different organisms. To date, KEGG database contains 232 biological pathways for human. The relations between pathway nodes in KEGG are activation, inhibition, expression, repression, indirect effect, state change, binding/association, dissociation, missing interaction, phosphorylation, dephosphorylation, glycosylation, ubiquitination, and methylation.

Same biological phenomena can play role in different biological pathways rendering overlapping genes in different pathways. When generating the simulated microarray data to test the proposed algorithm, we chose pathways that did not have any genes in common. Considering a graph for the KEGG pathways where nodes represent pathways, and edge weights represent the number of common genes between two pathways, our pathway set can be found using the independent vertex set problem, which is an NP-complete problem and returns a subset of pairwise non-adjacent vertices [55]. In our domain, the independent vertex set algorithm gives us a list of KEGG pathways, no two of which share any genes. We found a maximal independent vertex set with length 29. For the sake of simplicity, we chose 14 pathways instead of 29. SynTReN v1.12 [56] was used to generate the signal levels for the genes in each of the 14 pathways with 20 control and 20 test samples and 10% background noise [56]. This way we have a simulated microarray data that follows the case where the chosen pathways have been

set to as active. To this end, we wish to cluster the microarray data and build BNs for each module.

4.1. Clustering

To be able to assess the performance of our work, and demonstrate the worst case, we partitioned our synthetic gene expression data, generated from the selected 14 pathways, into 14 clusters via the k-means clustering algorithm. It should be noted that, k-means clustering algorithm may not give the same final clustering due to the selection of initial centroids randomly. In order to minimize the deficiency of the clustering solution, we ran the k-means procedure 5 times and the consensus of five clustering results was used for further analyses. Each time initial centroids for k-means are selected randomly and consensus of clustering has been achieved by forming a consensus matrix whose entries denotes the number of times gene i was clustered with gene j. Consequently, consensus matrix is a similarity matrix, therefore we can use it as input to k-means clustering algorithm. Minimum and maximum cluster sizes are 2 and 35 respectively, the mean is 15.86, and the standard deviation is 9.39. In Table 4.1 and Table 4.2, the number of genes corresponding to each of the original KEGG pathways and the number of genes in each cluster are shown.

Table 4.1. Selected Pathway IDs and Number of Genes in Each Pathway.

Pathway ID	Number of Genes
hsa00100	19
hsa00120	16
hsa00232	7
hsa00471	4
hsa00531	19
hsa00592	20
hsa00630	18
hsa00770	17
hsa00780	2
hsa00790	13
hsa00920	13
hsa04710	23
hsa04744	29
hsa05216	29
	Total: 222

Table 4.2. Cluster IDs and Number of Genes in Each Cluster.

Cluster ID	Number of Genes
Cluster 1	11
Cluster 2	13
Cluster 3	22
Cluster 4	13
Cluster 5	35
Cluster 6	23
Cluster 7	17
Cluster 8	25
Cluster 9	13
Cluster 10	4
Cluster 11	2
Cluster 12	7
Cluster 13	8
Cluster 14	29
	Total: 222

In Tables 4.3 through 4.16 we list the number of genes each pathway is represented by in each of the 14 clusters.

Table 4.3. The Relation of Cluster 1 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00770	4	36.36
hsa00592	2	18.18
hsa00630	2	18.18
hsa05216	1	9.09
hsa04710	1	9.09
hsa00120	1	9.09

Table 4.4. The Relation of Cluster 2 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa05216	4	30.77
hsa00630	3	23.08
hsa00790	2	15.38
hsa00100	2	15.38
hsa04710	1	7.69
hsa00531	1	7.69

Table 4.5. The Relation of Cluster 3 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa04744	6	27.27
hsa00790	4	18.18
hsa00630	2	9.09
hsa05216	2	9.09
hsa00770	2	9.09
hsa00232	2	9.09
hsa00780	1	4.55
hsa04710	1	4.55
hsa00120	1	4.55
hsa00531	1	4.55

Table 4.6. The Relation of Cluster 4 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00531	6	46.15
hsa04744	4	30.77
hsa00120	1	7.69
hsa05216	1	7.69
hsa00780	1	7.69

Table 4.7. The Relation of Cluster 5 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa04710	8	22.86
hsa00120	5	14.29
hsa00592	4	11.43
hsa05216	3	8.57
hsa00100	3	8.57
hsa00920	3	8.57
hsa00531	3	8.57
hsa00770	2	5.71
hsa04744	2	5.71
hsa00790	1	2.86
hsa00630	1	2.86

Table 4.8. The Relation of Cluster 6 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa05216	6	26.09
hsa04744	6	26.09
hsa00592	4	17.39
hsa00630	2	8.70
hsa00790	2	8.70
hsa00770	2	8.70
hsa04710	1	4.35

Table 4.9. The Relation of Cluster 7 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00770	5	29.41
hsa00592	2	11.76
hsa00630	2	11.76
hsa00100	2	11.76
hsa00920	2	11.76
hsa05216	1	5.88
hsa00120	1	5.88
hsa04744	1	5.88
hsa00531	1	5.88

Table 4.10. The Relation of Cluster 8 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00100	5	20.00
hsa00232	4	16.00
hsa04710	4	16.00
hsa00120	4	16.00
hsa05216	2	8.00
hsa00770	2	8.00
hsa04744	2	8.00
hsa00630	1	4.00
hsa00790	1	4.00

Table 4.11. The Relation of Cluster 9 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa04744	4	30.77
hsa00471	3	23.08
hsa00531	3	23.08
hsa00790	2	15.38
hsa00100	1	7.69

Table 4.12. The Relation of Cluster 10 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00592	1	25.00
hsa00630	1	25.00
hsa00920	1	25.00
hsa04744	1	25.00

Table 4.13. The Relation of Cluster 11 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa04710	1	50.00
hsa00592	1	50.00

Table 4.14. The Relation of Cluster 12 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00592	2	28.57
hsa04710	2	28.57
hsa00471	1	14.29
hsa00120	1	14.29
hsa00531	1	14.29

Table 4.15. The Relation of Cluster 13 and KEGG Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa00592	2	25.00
hsa04744	2	25.00
hsa00630	1	12.50
hsa00100	1	12.50
hsa00920	1	12.50
hsa00531	1	12.50

Table 4.16. The Relation of Cluster 14 and Kegg Pathways.

KEGG Pathway	Occurrence	Ratio %
hsa05216	8	27.59
hsa00920	6	20.69
hsa00100	5	17.24
hsa04710	4	13.79
hsa00120	2	6.90
hsa00531	2	6.80
hsa00790	1	3.45
hsa00232	1	3.45

These results suggest that in most of the cases, a given cluster dominantly represents genes from one or a few pathways.

4.2. Atlas Construction

The primary goal of pathway atlas construction is to see the extended exhibition of the pathway interactions. Our test data for atlas construction is generated synthetically from 14 pathways that contain a total of 222 genes. Learning the structure of 222 genes is computationally expensive, since the space of all possible structures grows exponentially in the number of variables n. That is, there are $\frac{n(n-1)}{2}$ possible undirected edges and $2^{\frac{n(n-1)}{2}}$ subset of these edges. Besides, more than one orientation of edges may exist. Therefore, we have clustered the synthetic data into 14 clusters to reduce the search space.

We have applied the structure learning algorithm into 14 clusters by using two methods: BNP and FLAT. The former one uses external biological knowledge in its model to score the candidate graphs and the latter does not resort external knowledge. Results of these runs show us the internal topology of each clusters. Yet, our goal is

to construct the interactions of the given pathways. Therefore, relations in between clusters are also important to discover significant inter-relations.

Two different strategies were used to merge the clusters using pairwise links between them. In the first strategy, we selected representative genes by taking into account their out-degrees in the graphs of the corresponding clusters. A high out-degree represents more involved genes in the interaction network. Each cluster is represented by the gene with the highest out-degree in the cluster's learned network. However, there may be more than one gene with the maximum out-degree in a given cluster. In those cases, we used all the genes with the maximum out-degree as the cluster's representative. The representative genes' expression values were sifted from synthetic microarray data and subjected to our structure learning algorithm. All tied genes in the same cluster (i.e., genes with the same out-degrees) were reduced into one representative gene for the corresponding clusters by performing row-wise and column-wise OR operation in the adjacency matrix of representative genes' interaction graph. If there is an edge between representative genes of two clusters, we sift genes that are in these two clusters from synthetic microarray data, and perform the proposed structure learning using the sifted gene expression data. Finally, we combine the adjacency matrices of clusters and merged the clusters to construct the interaction atlas.

In the second strategy, we have applied Google's Pagerank algorithm with %90 damping parameter [57] to select representative genes and apply the same procedure as above to combine clusters and construct the atlas. The adjacency matrix of the clusters and the representative genes are in Appendix A. In Figure 4.1, we show the interaction network between clusters based on the network used with representative genes found based on their out-degrees.

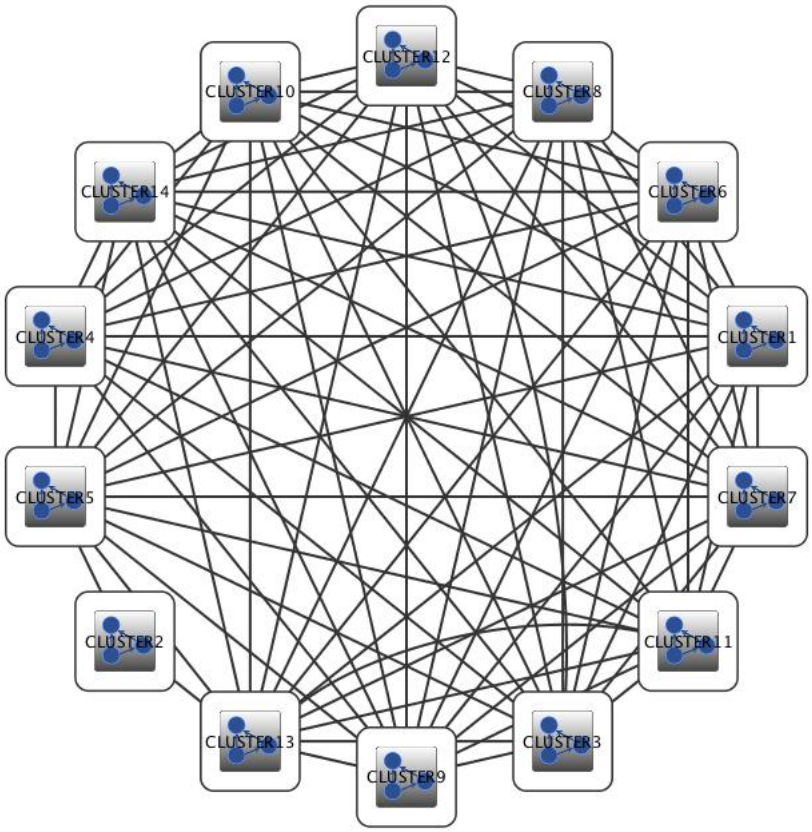

Figure 4.1. Interactions Between Clusters.

We assess the performance of our methodology by reverse engineering the original KEGG pathways from constructed gene interaction atlas. Our performance measuring criteria is in terms of area under the receiver operating characteristic curve (ROC AUC). We compare the deduced network from our atlas using the genes in a given KEGG pathway to the original KEGG pathway. The highest and lowest AUC values using the BNP algorithm with representative gene selection by the out-degree approach were 0.87 and 0.53, respectively, with an average value of 0.73. However, the success of the BNP counterpart, namely FLAT, had a maximum of 0.72, and minimum of 0.06 AUC values, with an average value of 0.54. By changing the representative gene selection approach from out-degree to Google's Pagerank, BNP resulted in a maximum of 0.75, and minimum of 0.48 AUC values, with an average value of 0.6. Using the Pagerank approach, the FLAT algorithm resulted in 0.61, and 0.40 highest and lowest

AUC values, respectively, with an average value of 0.51. These results, summarized in Figure 4.2, suggest that the proposed method outperforms the FLAT method in terms of the accuracy of atlas generation based on known pathways that make up the atlas. The added value due to the incorporation of the external biological knowledge is reflected in the accuracy performance of the proposed method. Moreover, the out-degree based selection method for the representative genes yielded higher AUC values compared to the Google's Pagerank method. In Table 4.17 and Table 4.18, we show the AUC values attained by the BNP and FLAT algorithms for each selected KEGG pathway with different representative gene selection methods. This could be due to the community structure seen in biological pathways which is represented by their scale-free nature. These networks generally hold a few hubs (nodes with large degrees), which in our framework ends up to be a good representative node reflecting most of the dependency structure.

Table 4.17. AUC Values for the BNP (Out-Degree) and FLAT (Out-Degree) Algorithms.

Pathway ID (Number of Nodes, Number of Edges)	BNP (Out-Degree)	FLAT (Out-Degree)
hsa00100 (19, 43)	0.76	0.68
hsa00120 (16, 23)	0.81	0.69
hsa00232 (7, 7)	0.64	0.45
hsa00471 (4, 6)	0.80	0.50
hsa00531 (19, 24)	0.86	0.72
hsa00592 (18, 17)	0.53	0.06
hsa00630 (15, 25)	0.77	0.61
hsa00770 (17, 44)	0.87	0.53
hsa00780 (2, 2)	0.75	0.50
hsa00790 (13, 16)	0.66	0.55
hsa00920 (13, 26)	0.73	0.64
hsa04710 (23, 104)	0.63	0.60
hsa04744 (28, 49)	0.63	0.47
hsa05216 (28, 49)	0.75	0.56

Table 4.18. AUC Values for the BNP (Pagerank) and Flat (Pagerank) Algorithms.

Pathway ID (Number of Nodes, Number of Edges)	BNP (Pagerank)	FLAT (Pagerank)
hsa00100 (19, 43)	0.61	0.61
hsa00120 (16, 23)	0.63	0.55
hsa00232 (7, 7)	0.48	0.48
hsa00471 (4, 6)	0.70	0.40
hsa00531 (19, 24)	0.67	0.55
hsa00592 (18, 17)	0.63	0.40
hsa00630 (15, 25)	0.54	0.57
hsa00770 (17, 44)	0.57	0.51
hsa00780 (2, 2)	0.75	0.50
hsa00790 (13, 16)	0.54	0.47
hsa00920 (13, 26)	0.67	0.51
hsa04710 (23, 104)	0.55	0.53
hsa04744 (28, 49)	0.51	0.48
hsa05216 (28, 49)	0.57	0.52

4.2.1. Topological Parameters of the Resulting Networks

A graph or network can be represented as an adjacency matrix, and any element of this matrix is given as follows:

$$\alpha_{ij} = \begin{cases} 1, & \text{if } i \to j \\ 0, & \text{if } i \nrightarrow j \end{cases}$$

By using this representation, various graph parameters can be calculated, which can provide an insight to understand complex graphs such as a gene interaction atlas.

The degree of any node i is represented by $k_i = \sum_j \alpha_{ij}$. The higher degree of a node implies stronger connectivity of the node in the graph. Moreover, the average

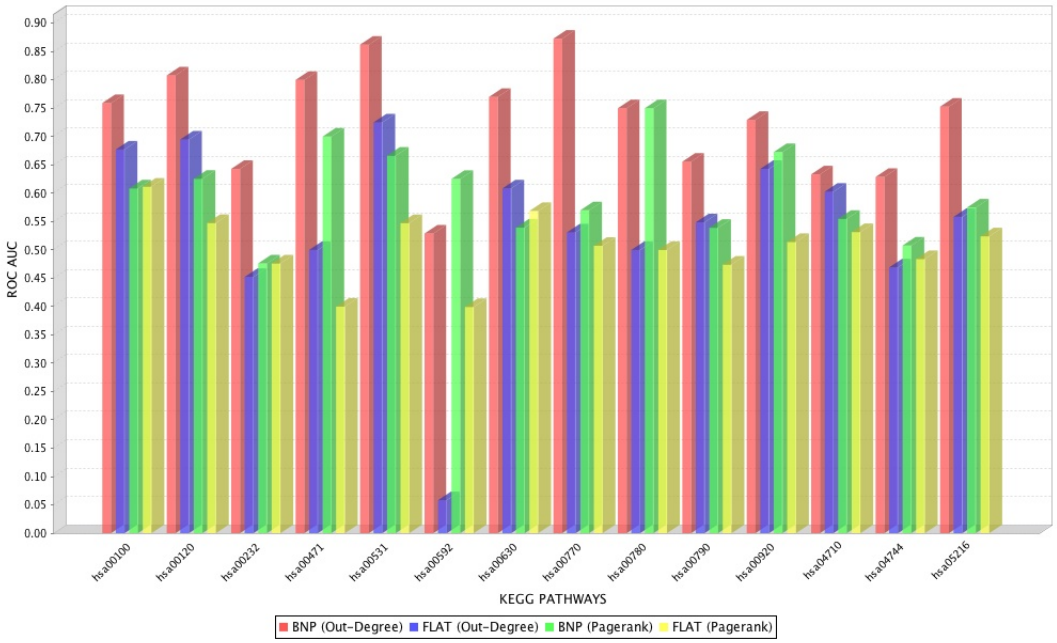

Figure 4.2. Performance Comparison of Different Strategies.

degree of a graph is given by $\bar{k} = \frac{\sum_i k_i}{N}$ where N is the total number of nodes in the graph. The higher average degree implies good inter-connectivity among the nodes in the graph.

Characteristic path length of a graph tells us the shortest path length between two nodes averaged over all pairs of nodes. Its formula is represented by $L = \frac{\sum_i \sum_j L_{ij}}{N(N-1)}$ where L_{ij} is the shortest path length between node i and node j. A high characteristic path length asserts that the graph is almost linear, and a low characteristic path length shows that the graph is in compact form.

Clustering coefficient is a measure for local density in the graph. Clustering coefficient C_i of a node i is given by $C_i = \frac{e_i}{\frac{k_i(k_i-1)}{2}}$ where e_i is the number of edges between the nearest neighbors of the i^{th} node, and k_i is the number neighbors of node i. The average clustering coefficient of a graph is given by $\bar{C} = \frac{\sum_i C_i}{N}$. The value of clustering coefficient denotes the probability that two adjacencies of a node i are connected to each other.

Network diameter is the largest distance between two nodes. A network with low diameter is called "small world" network, and biological networks tend to have low diameters [58]. For instance, metabolic networks have low diameters, which may enable to reduce transition times between metabolic states.

Betweenness centrality is a measure of a node's centrality in the graph. It denotes the fraction of all of the shortest paths between all nodes in a network that pass through a given node [59]. The betweenness centrality assumes that there aren't any isolated components because its calculation relies on the shortest paths. Moreover, the measurement closeness centrality is also based on the shortest path length. When the average path length between a node i and the rest of the nodes is low, centrality of the node i would be high. On the other hand, stress centrality represents the number of shortest paths between all node pairs that pass through a particular node. Intuitively, stress centrality denotes the amount of work performed by each node in the graph. A higher value of closeness or betweenness centrality can be interpreted as initial candidates for regulatory genes. However, either closeness or betweenness centrality is measured based on the shortest path which leads to ignore spreading information through non-shortest paths. Moreover, high values of betweenness centrality shows more efficiently organized metabolic networks.

The in-degree and out-degree distribution of the atlas roughly follows a power-law, which is similarly seen in other biological networks. In this scale-free structure, we see few nodes with a high number of connections, which represents the hubs. The average clustering coefficient of an atlas is 0.209, which is relatively smaller than the ones observed in other biological networks, which tend to be around 0.5 [60]. Therefore, we believe the generated atlas presents a less isolated community structure than generally seen. This could be due to the liberal procedure followed in merging the clusters. If a more stringent criterion were used, we might have observed a more hierarchical structure with higher tightly knit, separated communities. The average shortest path length of the atlas is about 3, which implies a reachable network where any given node is, on average, can reach to another node in about three hops. Nevertheless, these parameters diverge from the corresponding values seen in random networks and can

be considered similar to the ones observed in real biological networks. The network diameter of an atlas is 7, which is relatively small, and may direct researchers to make a further investigation on genes 7 apart from any of each.

Table 4.19. Statistics of Learned Gene Interaction Atlas.

Clustering Coefficient	0.21
Network Diameter	7
Characteristic Path Length	2.97
Average Number of Neighbors	27.65
Number of Nodes	222

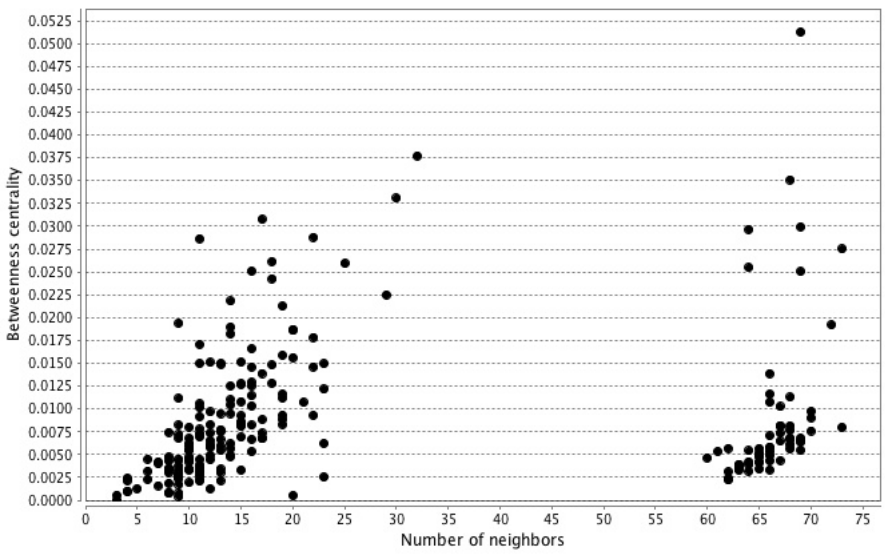

Figure 4.3. Betweenness Centrality of Gene Interaction Atlas.

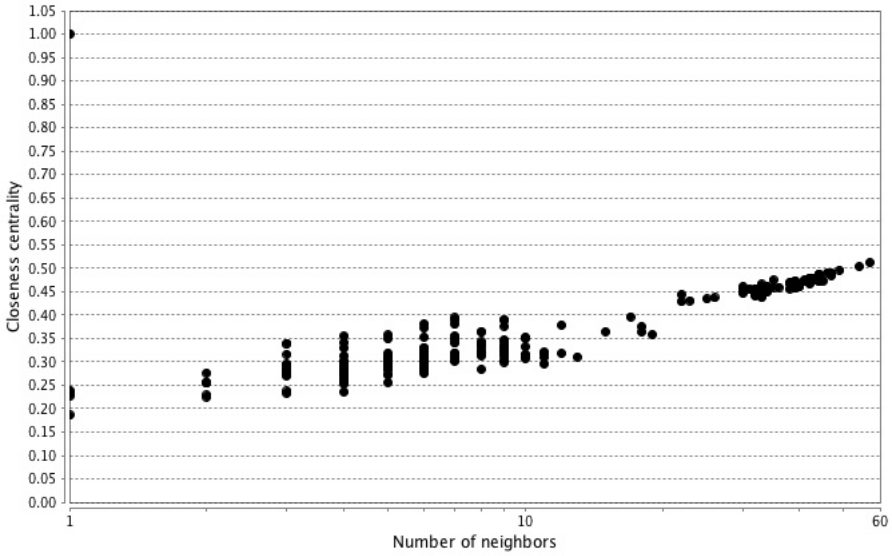

Figure 4.4. Closeness Centrality of Gene Interaction Atlas.

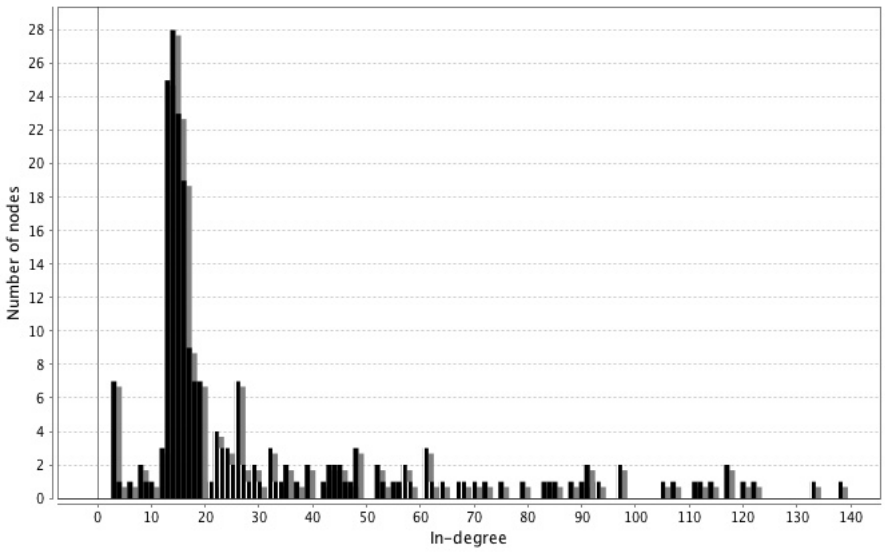

Figure 4.5. In-Degree Distribution of Gene Interaction Atlas.

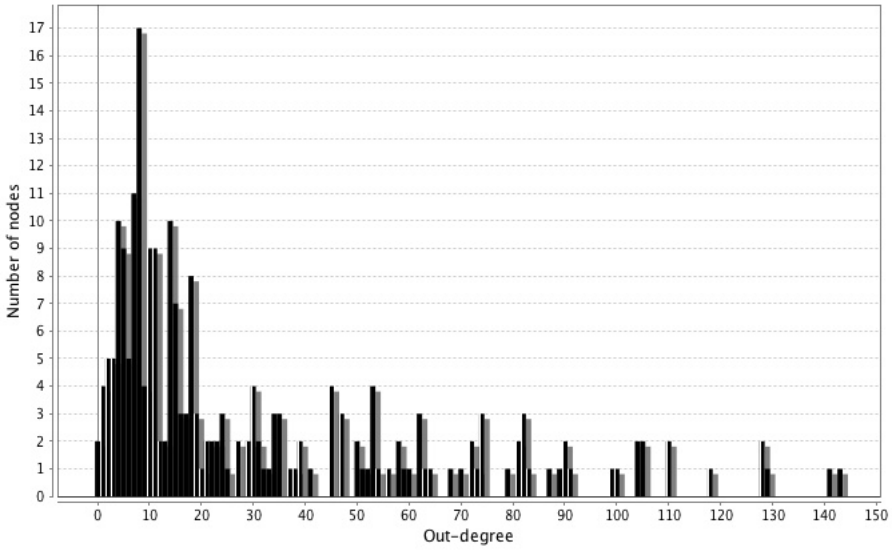

Figure 4.6. Out-Degree Distribution of Gene Interaction Atlas.

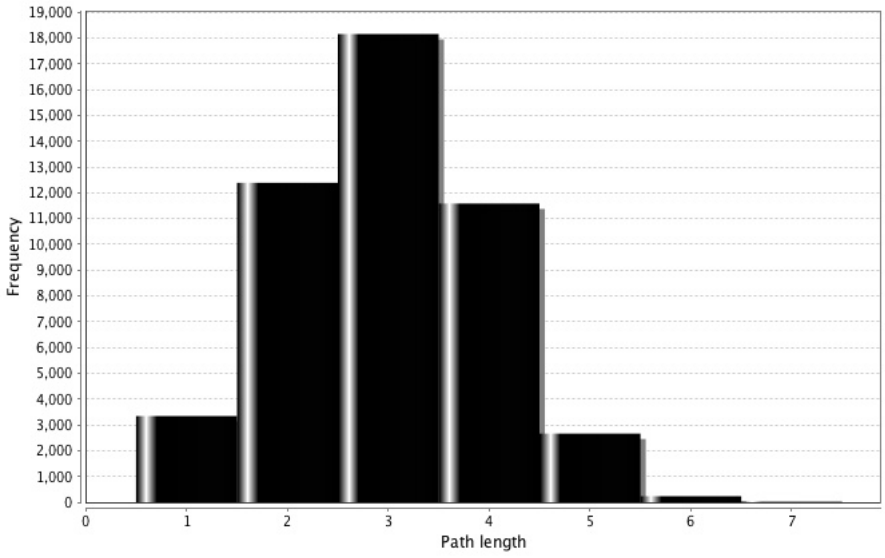

Figure 4.7. Histogram of the Shortest Path in the Atlas.

5. CONCLUSION

This thesis has investigated a computational method for inferring a gene regulatory atlas from microarray data by incorporating external biological knowledge. In this investigation, the aim was to explore undiscovered gene-gene interactions in order to provide an insight for discovering new biological pathways, and updating our knowledge on already identified biological phenomena. Incorporating external knowledge into inferring gene regulatory networks reduces the data required. Since Bayesian networks can integrate external knowledge into its calculations, this study was undertaken to design a computational method by means of Bayesian networks. The results of this research support the idea that external biological knowledge significantly increases the success rate of network inference. This research will serve as a base for future studies, and it can provide an answer to unsolved biological questions in the basis of gene regulations. Several limitations to this pilot study need to be acknowledged. Only synthetic data is used in simulations and due to the limitations of time and computational resources, we have limited ourselves only 14 biological pathways. Future research should therefore concentrate on the investigation of gene regulatory networks from real microarray data, and increase the gene interaction atlas by using more biological pathways.

APPENDIX A: CLUSTERS AND REPRESENTATIVE GENES

Adjacency matrices of 14 clusters are shown in Table A.1 through Table A.14. Node labels with bold faces denotes representative genes in the respected cluster.

Table A.1. Adjacency Matrix of Cluster 1.

	80270	81579	**84647**	84693	81888	80347	79717	79646	80025	79365	83439
80270	0	0	0	0	0	0	1	0	0	0	0
81579	0	0	0	0	1	0	0	0	0	0	0
84647	1	1	0	1	0	0	0	0	0	0	1
84693	0	0	0	0	0	0	0	0	0	0	0
81888	0	0	0	0	0	0	0	0	0	0	0
80347	0	0	0	0	0	0	0	0	0	0	0
79717	0	0	0	0	0	0	0	1	1	0	0
79646	0	0	0	0	0	1	0	0	0	0	0
80025	0	0	0	0	0	1	0	0	0	1	0
79365	0	0	0	0	0	0	0	0	0	0	0
83439	0	0	0	0	0	0	0	0	0	0	0

Table A.2. Adjacency Matrix of Cluster 2.

	3988	4047	4669	4190	**4191**	4594	4337	4338	4862	3845	4893	4914	4609
3988	0	0	0	0	0	0	0	0	0	0	0	0	0
4047	0	0	0	0	0	0	0	0	1	0	0	0	0
4669	0	0	0	0	0	0	0	0	0	0	0	0	0
4190	0	0	0	0	0	0	0	0	0	0	0	0	0
4191	1	0	0	1	0	1	0	0	0	0	1	0	0
4594	0	0	0	0	0	0	0	0	0	0	0	0	1
4337	0	0	0	0	0	0	0	1	0	0	0	1	0
4338	0	0	0	0	0	0	0	0	0	0	0	0	0
4862	0	0	0	0	0	0	0	0	0	0	0	0	0
3845	0	1	1	0	0	0	0	0	0	0	0	0	0
4893	0	0	0	1	0	0	0	0	0	1	0	0	0
4914	0	0	0	0	1	0	0	0	0	1	1	0	0
4609	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.3. Adjacency Matrix of Cluster 3.

	570	10	9	411	48	50	586	587	686	248	249	250	251	406	**408**	**409**	801	805	808	810	673	595
570	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
411	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
50	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
586	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
587	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
686	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
248	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
249	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
251	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
406	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
408	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0
409	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	0	0
801	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0
805	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
808	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0
810	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
673	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
595	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Table A.4. Adjacency Matrix of Cluster 4.

	3295	3423	3073	3074	2990	3425	3373	3141	**2978**	2986	3000	2979	3265
3295	0	0	0	0	0	0	0	0	0	0	0	0	0
3423	0	0	0	0	0	1	0	1	0	0	0	0	0
3073	0	0	0	0	0	0	0	0	0	0	0	0	0
3074	0	0	1	0	0	0	0	0	0	0	0	0	0
2990	0	0	0	0	0	0	0	0	1	0	0	1	0
3425	0	0	0	0	0	0	1	0	0	0	0	0	0
3373	0	0	0	0	0	0	0	0	0	0	0	0	0
3141	0	0	0	0	0	0	0	0	0	0	0	0	0
2978	1	1	0	0	0	0	0	0	0	0	1	1	0
2986	0	0	0	1	0	0	0	0	0	0	0	0	0
3000	0	0	0	0	0	0	0	0	0	0	0	0	0
2979	0	0	0	0	0	0	0	0	0	1	1	0	1
3265	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.5. Adjacency Matrix of Cluster 5.

	8435	10682	10826	10858	10998	9420	8309	9023	10855	8372	8692	9415	**8398**	8399	8681	9380	8875	8876	8836	10380	9060	9061	8863	8864	9575	8553	9572	8945	8454	9978	8787	9626	10342	8030	8031
8435	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10682	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10826	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10858	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
10998	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9420	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8309	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9023	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
10855	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
8372	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8692	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9415	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8398	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
8399	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
8681	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
9380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8875	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
8876	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
8836	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9060	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9061	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8863	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8864	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
9575	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0
8553	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9572	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8945	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8454	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9978	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8787	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
9626	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
10342	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8030	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
8031	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Table A.6. Adjacency Matrix of Cluster 6.

	5319	5320	5321	5322	5095	5096	5167	5169	5805	5860	5187	6010	5145	5148	5158	5957	6011	**5468**	5604	5594	5595	5605	5979
5319	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5320	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
5321	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1
5322	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5095	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5096	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
5167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5169	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5805	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5860	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5187	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6010	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
5145	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5148	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5158	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5957	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
6011	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5468	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
5604	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5594	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
5595	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
5605	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5979	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.7. Adjacency Matrix of Cluster 7.

	50814	51478	51302	60495	50487	64600	51179	54363	51733	53354	60490	55229	60496	50515	55501	**51806**	51176
50814	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51302	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
60495	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
50487	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
64600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
51179	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54363	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51733	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53354	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
60490	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
55229	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
60496	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50515	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
55501	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51806	0	0	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0
51176	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Table A.8. Adjacency Matrix of Cluster 8.

	1056	1718	1594	1595	1717	1109	1593	1581	1582	1544	1548	1549	1553	1431	1806	1807	1719	1407	1408	1453	1454	1258	1259	1499	999
1056	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1718	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1594	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1717	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1109	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1593	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1581	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1544	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1548	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1549	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1553	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1431	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1806	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1807	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1719	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1407	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1408	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1453	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1454	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1258	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1259	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1499	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
999	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0

Table A.9. Adjacency Matrix of Cluster 9.

	2222	2746	2747	**2744**	2588	2720	2799	2356	2643	2779	2780	2782	2792
2222	0	0	0	0	1	0	0	0	0	0	0	0	0
2746	0	0	0	0	0	0	0	0	0	0	0	0	0
2747	0	0	0	0	0	0	0	0	0	0	0	0	0
2744	1	1	1	0	1	0	0	0	0	0	0	0	0
2588	0	0	0	0	0	1	0	0	0	0	0	0	0
2720	0	0	0	0	0	0	1	0	0	1	0	0	0
2799	0	0	0	0	0	0	0	1	0	0	0	0	0
2356	0	0	0	0	0	0	0	0	0	0	0	0	0
2643	1	0	0	1	1	0	0	0	0	0	0	0	0
2779	0	0	0	0	0	0	0	0	0	0	1	1	0
2780	0	0	0	0	0	0	0	0	0	0	0	1	0
2782	0	0	1	0	0	0	0	0	0	0	0	0	1
2792	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.10. Adjacency Matrix of Cluster 10.

	391013	**283871**	445329	346562
391013	0	0	1	0
283871	1	0	0	1
445329	0	0	0	0
346562	0	0	0	0

Table A.11. Adjacency Matrix of Cluster 11.

	100137049	100506332
100137049	0	1
100506332	0	0

Table A.12. Adjacency Matrix of Cluster 12.

	23600	**27165**	23553	26279	30814	23291	26224
23600	0	0	1	0	0	0	0
27165	0	0	0	1	1	0	1
23553	0	0	0	0	0	1	0
26279	0	0	0	0	1	0	0
30814	0	0	0	0	0	0	0
23291	0	0	0	0	0	0	0
26224	1	0	0	1	1	0	0

Table A.13. Adjacency Matrix of Cluster 13.

	120227	138050	123745	**151056**	132158	166012	131890	163688
120227	0	0	1	0	0	1	0	1
138050	0	0	0	0	1	0	0	0
123745	0	0	0	0	0	0	1	1
151056	1	0	1	0	0	0	0	1
132158	0	0	0	0	0	0	0	0
166012	0	0	0	0	0	0	0	0
131890	0	1	0	0	0	0	0	0
163688	0	0	0	0	0	0	0	0

Table A.14. Adjacency Matrix of Cluster 14.

	6646	6309	7108	6307	6713	6718	6342	7498	6448	6677	6697	6821	6783	6799	6817	6818	6820	6095	6096	6097	6500	6932	6934	6256	6257	6258	7170	7175	**7849**
6646	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6309	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7108	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6713	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6718	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6342	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7498	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6448	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6677	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6697	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6821	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
6783	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
6799	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6817	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
6818	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6820	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6095	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6096	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6097	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6932	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6934	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6256	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
6257	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
6258	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7170	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7175	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
7849	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0

Table A.15. Adjacency Matrix of Representative Genes.

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8	Cluster9	Cluster10	Cluster11	Cluster12	Cluster13	Cluster14
Cluster1	0	0	1	0	1	1	1	1	1	1	1	1	1	1
Cluster2	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Cluster3	0	0	0	0	0	0	0	0	1	0	0	1	0	0
Cluster4	1	0	1	0	1	1	1	1	1	1	1	1	0	1
Cluster5	0	1	1	0	0	0	1	0	1	1	1	1	1	1
Cluster6	0	0	1	0	1	0	1	1	1	1	1	1	1	1
Cluster7	0	0	1	0	0	0	0	0	1	1	1	1	1	1
Cluster8	0	0	1	0	1	0	1	0	1	1	1	1	1	1
Cluster9	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Cluster10	0	0	1	0	0	0	0	0	1	0	1	1	0	0
Cluster11	0	0	1	0	0	0	0	0	1	0	0	1	1	0
Cluster12	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Cluster13	0	0	1	0	0	0	0	0	0	1	0	0	0	1
Cluster14	0	0	1	0	0	0	0	0	1	1	1	0	0	0

REFERENCES

1. Wu, L. C., C. W. Chang, T. M. Chao, R. H. Yeh and J. T. Horng, "A System to Discover Correlations within a Biological Pathway between the Expression Levels of Genes", *Bioinformatics and Bioengineering (BIBE), 2011 IEEE 11th International Conference on*, pp. 15–20, IEEE, 2011.

2. Szederkényi, G., J. Banga and A. Alonso, "Inference of Complex Biological Networks: Distinguishability Issues and Optimization-Based Solutions", *BMC systems biology*, Vol. 5, No. 1, p. 177, 2011.

3. Sharma, V., *A Text Book of Bioinformatics*, Rastogi Publications, New Delhi, 2008.

4. 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 of the United States of America*, Vol. 102, No. 43, pp. 15545–15550, 2005.

5. 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.

6. Akutsu, T., S. Miyano, S. Kuhara *et al.*, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model", *Pacific Symposium on Biocomputing*, Vol. 4, pp. 17–28, World Scientific Maui, Hawaii, 1999.

7. de Jong, H. and M. Page, "Search for steady states of piecewise-linear differential equation models of genetic regulatory networks", *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, Vol. 5, No. 2, pp. 208–222, 2008.

8. Pohorille, A., "Incorporating biological knowledge into evaluation of causal regulatory hypotheses", *Pacific Symposium on Biocomputing 2003: Kauai, Hawaii, 3-7 January 2003*, p. 128, World Scientific Publishing Company, 2002.

9. Schaeffer, S. E., "Graph clustering", *Computer Science Review*, Vol. 1, No. 1, pp. 27–64, 2007.

10. Newman, M. E., "Detecting community structure in networks", *The European Physical Journal B-Condensed Matter and Complex Systems*, Vol. 38, No. 2, pp. 321–330, 2004.

11. Fortunato, S., "Community detection in graphs", *Physics Reports*, Vol. 486, No. 3, pp. 75–174, 2010.

12. Wagner, G. P., M. Pavlicev and J. M. Cheverud, "The road to modularity", *Nature Reviews Genetics*, Vol. 8, No. 12, pp. 921–931, 2007.

13. Rives, A. W. and T. Galitski, "Modular organization of cellular networks", *Proceedings of the National Academy of Sciences*, Vol. 100, No. 3, pp. 1128–1133, 2003.

14. Spirin, V. and L. A. Mirny, "Protein complexes and functional modules in molecular networks", *Proceedings of the National Academy of Sciences*, Vol. 100, No. 21, pp. 12123–12128, 2003.

15. Chen, J. and B. Yuan, "Detecting functional modules in the yeast protein–protein interaction network", *Bioinformatics*, Vol. 22, No. 18, pp. 2283–2290, 2006.

16. Farutin, V., K. Robison, E. Lightcap, V. Dancik, A. Ruttenberg, S. Letovsky and J. Pradines, "Edge-count probabilities for the identification of local protein communities and their organization", *Proteins: Structure, Function, and Bioinformatics*, Vol. 62, No. 3, pp. 800–818, 2006.

17. Guimera, R. and L. A. N. Amaral, "Functional cartography of complex metabolic networks", *Nature*, Vol. 433, No. 7028, pp. 895–900, 2005.

18. Wilkinson, D. M. and B. A. Huberman, "A method for finding communities of related genes", *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 101, No. Suppl 1, pp. 5241–5248, 2004.

19. Segal, E., D. Pe'er, A. Regev and D. Koller, "Learning module networks", *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 525–534, Morgan Kaufmann Publishers Inc., 2002.

20. D'haeseleer, P., S. Liang and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering", *Bioinformatics*, Vol. 16, No. 8, pp. 707–726, 2000.

21. Gibson, G., "Microarray analysis", *PLoS Biology*, Vol. 1, No. 1, p. e15, 2003.

22. Aarhus, M., M. Lund-Johansen and P. M. Knappskog, "Gene expression profiling of meningiomas: current status after a decade of microarray-based transcriptomic studies", *Acta neurochirurgica*, Vol. 153, No. 3, pp. 447–456, 2011.

23. Lähdesmäki, H., I. Shmulevich and O. Yli-Harja, "On learning gene regulatory networks under the Boolean network model", *Machine Learning*, Vol. 52, No. 1-2, pp. 147–167, 2003.

24. Fujita, S., M. Matsui, H. Matsuno and S. Miyano, "Modeling and simulation of fission yeast cell cycle on hybrid functional Petri net", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 87, No. 11, pp. 2919–2928, 2004.

25. Kam, N., D. Harel, H. Kugler, R. Marelly, A. Pnueli, J. A. Hubbard and M. J. Stern, "Formal modelling of C. elegans development. A scenario-based approach", *Modelling in molecular biology*, pp. 151–173, Springer, 2004.

26. Curti, M., P. Degano, C. Priami and C. T. Baldari, "Modelling biochemical pathways through enhanced π-calculus", *Theoretical Computer Science*, Vol. 325, No. 1, pp. 111–140, 2004.

27. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference*, Morgan Kaufmann Pub, 1988.

28. Cheng, J., D. A. Bell and W. Liu, "An algorithm for Bayesian belief network construction from data", *proceedings of AI & STAT'97*, pp. 83–90, 1997.

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

30. Chickering, D. M., D. Heckerman and C. Meek, "Large-sample learning of Bayesian networks is NP-hard", *The Journal of Machine Learning Research*, Vol. 5, pp. 1287–1330, 2004.

31. Tsamardinos, I., L. E. Brown and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm", *Machine learning*, Vol. 65, No. 1, pp. 31–78, 2006.

32. Yu, J., V. A. Smith, P. P. Wang, A. J. Hartemink and E. D. Jarvis, "Advances to Bayesian network inference for generating causal networks from observational biological data", *Bioinformatics*, Vol. 20, No. 18, pp. 3594–3603, 2004.

33. 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, `http://www.sciencedirect.com/science/article/pii/S0303264708002608`.

34. Lezon, T. R., J. R. Banavar, M. Cieplak, A. Maritan and N. V. Fedoroff, "Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns", *Proceedings of the Na-*

tional Academy of Sciences, Vol. 103, No. 50, pp. 19033–19038, 2006, `http://www.pnas.org/content/103/50/19033.abstract`.

35. Liang, S., S. Fuhrman, R. Somogyi *et al.*, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures", *Pacific symposium on biocomputing*, Vol. 3, p. 2, 1998.

36. Yeung, M. S., J. Tegnér and J. J. Collins, "Reverse engineering gene networks using singular value decomposition and robust regression", *Proceedings of the National Academy of Sciences*, Vol. 99, No. 9, pp. 6163–6168, 2002.

37. 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.

38. Hartemink, A. J. *et al.*, "Reverse engineering gene regulatory networks", *Nature biotechnology*, Vol. 23, No. 5, pp. 554–555, 2005.

39. Kim, S. Y., S. Imoto and S. Miyano, "Inferring gene networks from time series microarray data using dynamic Bayesian networks", *Briefings in bioinformatics*, Vol. 4, No. 3, pp. 228–235, 2003.

40. Fernández-Suárez, X. M. and M. Y. Galperin, "The 2013 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection", *Nucleic acids research*, Vol. 41, No. D1, pp. D1–D7, 2013.

41. 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", *Proceedings of the Pacific Symposium on Biocomputing (PSB'02)*, pp. 437–449, 2002.

42. 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 net-

work model with promoter element detection", *Bioinformatics*, Vol. 19, No. suppl 2, pp. ii227–ii236, 2003.

43. 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.

44. Werhli, A. V., D. Husmeier *et al.*, "Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge", *Stat Appl Genet Mol Biol*, Vol. 6, No. 1, p. 15, 2007.

45. 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.

46. 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.

47. 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.

48. 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.

49. 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.

50. 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.

51. Friedman, N., M. Goldszmidt and A. Wyner, "Data analysis with Bayesian networks: A bootstrap approach", *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 196–205, Morgan Kaufmann Publishers Inc., 1999.

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

53. 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, No. 1, p. 520, 2010.

54. Ay, F., M. Kellis and T. Kahveci, "SubMAP: aligning metabolic pathways with subnetwork mappings", *Journal of Computational Biology*, Vol. 18, No. 3, pp. 219–235, 2011.

55. Tarjan, R. E. and A. E. Trojanowski, "Finding a maximum independent set", *SIAM Journal on Computing*, Vol. 6, No. 3, pp. 537–546, 1977.

56. 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.

57. Rogers, I., "The Google Pagerank algorithm and how it works", *IPR Computing*, 2002.

58. Newman, M. E., "The structure and function of complex networks", *SIAM review*,

Vol. 45, No. 2, pp. 167–256, 2003.

59. Seebacher, J. and A.-C. Gavin, "SnapShot: Protein-protein interaction networks", *Cell*, Vol. 144, No. 6, p. 1000, 2011.

60. Lima-Mendez, G. and J. van Helden, "The powerful law of the power law and other myths in network biology", *Molecular BioSystems*, Vol. 5, No. 12, pp. 1482–1493, 2009.