Gene Regulatory Network Reconstruction Using Dynamic Bayesian Networks

Haoni Li
University of Southern Mississippi

Follow this and additional works at: https://aquila.usm.edu/dissertations

Part of the Applied Mathematics Commons, and the Computer Sciences Commons

Recommended Citation
https://aquila.usm.edu/dissertations/523

This Dissertation is brought to you for free and open access by The Aquila Digital Community. It has been accepted for inclusion in Dissertations by an authorized administrator of The Aquila Digital Community. For more information, please contact Joshua.Cromwell@usm.edu.
GENE REGULATORY NETWORK RECONSTRUCTION USING
DYNAMIC BAYESIAN NETWORKS

by

Haoni Li

Abstract of a Dissertation
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

May 2013
ABSTRACT

GENE REGULATORY NETWORK RECONSTRUCTION USING DYNAMIC BAYESIAN NETWORKS

by Haoni Li

May 2013

High-content technologies such as DNA microarrays can provide a system-scale overview of how genes interact with each other in a network context. Various mathematical methods and computational approaches have been proposed to reconstruct GRNs, including Boolean networks, information theory, differential equations and Bayesian networks. GRN reconstruction faces huge intrinsic challenges on both experimental and theoretical fronts, because the inputs and outputs of the molecular processes are unclear and the underlying principles are unknown or too complex.

In this work, we focused on improving the accuracy and speed of GRN reconstruction with Dynamic Bayesian based method. A commonly used structure-learning algorithm is based on REVEAL (Reverse Engineering Algorithm). However, this method has some limitations when it is used for reconstructing GRNs. For instance, the two-stage temporal Bayes network (2TBN) cannot be well recovered by application of REVEAL; it has low accuracy and speed for high dimensionality networks that has above a hundred nodes; and it even cannot accomplish the task of reconstructing a network with 400 nodes. We implemented an algorithm for DBN structure learning with Friedman’s score function to replace REVEAL, and tested it on reconstruction of both synthetic networks and real yeast networks and compared it with REVEAL in the absence or presence of preprocessed network generated by Zou and Conzen’s algorithm. The new
score metric improved the precision and recall of GRN reconstruction. Networks of gene interactions were reconstructed using a Dynamic Bayesian Network (DBN) approach and were analyzed to identify the mechanism of chemical-induced reversible neurotoxicity through reconstruction of gene regulatory networks in earthworms with tools curating relevant genes from non-model organism’s pathway to model organism pathway.
The University of Southern Mississippi

GENE REGULATORY NETWORK RECONSTRUCTION USING
DYNAMIC BAYESIAN NETWORKS

by

Haoni Li

A Dissertation
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:

Chaoyang Zhang
Director

Ping Gong

Nang Wang

Dia Ali

Ras B Pandey

Louis Perkins

Susan A. Siltanen
Dean of the Graduate School

May 2013
DEDICATION

This dissertation is dedicated to my parents for their love and support.
ACKNOWLEDGMENTS

I would like to thank Dr. Chaoyang Zhang, Dr. Ping Gong, and all the other committee members, Dr. Nan Wang, Dr. Dia Ali, Dr. Louis Perkins, and Dr. Ras B. Pandey for their suggestions and advices to improve my work during the whole process. I gratefully acknowledge my advisors, Dr. Zhang and Dr. Gong, for their generous helps in the past four years. Along the path of research that has led to this dissertation, they have constantly been there to provide me guidance, support and suggestions.

I am forever indebted to my parents, who have been extremely supportive and helpful despite living thousands miles away. My father always encourages me to do what I want to do, and my mother believes I am the best in the world, which gave me a lot of confidence.
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... ii

DEDICATION ......................................................................................................................... iv

ACKNOWLEDGMENTS ........................................................................................................ v

LIST OF TABLES ................................................................................................................... viii

LIST OF ILLUSTRATIONS ................................................................................................... ix

CHAPTER

I. INTRODUCTION .................................................................................................................. 1

Biological Background
DNA Microarray
Computational Analysis of Biological Networks
Contributions
Dissertation Organization

II. REVIEW OF MICROARRAY .............................................................................................. 10

DNA Microarray Experiment
Microarray Analysis

III. COMPUTATIONAL APPROACHES TO INFERENCE GRNS ........................................... 21

Differential Equations
Boolean Network
Probabilistic Graph

IV. PROBABILISTIC GRAPHICAL MODELS REPRESENTATION ....................................... 28

Undirected Graphical Models: Markov Networks
Bayesian Network Representation
Template-Based Representations

V. STRUCTURE LEARNING IN BAYESIAN NETWORKS ..................................................... 36

Constraint-Based Method
Score Based Method
Structure Search
VI. RECONSTRUCT GRNS FROM YEAST MICROARRAY DATA .................. 59

Saccharomyces Cell Cycle Data
Saccharomyces Genome Database Description
Reconstruct GRNs Using Time-Delayed DBN Approach

VII. IMPROVING STRUCTURE LEARNING FOR DBN.......................... 72

Algorithm Description
Existing Approaches for Comparison
Synthetic Data Description
Results for Synthetic Data
Results for Real Yeast Benchmark Dataset
Conclusions

VIII. RECONSTRUCTION OF GRNS FOR NEUROTOXICITY ...................... 86

Multi-Class Earthworm Microarray Dataset
Experimental Design and Dataset Generation
Data Pre-processing
Feature Filtering by Univariate Statistical Analysis
Identification of Significant Pathways/GRNs
Pathway Analysis

IX. CONCLUSIONS .................................................................................. 120

Summary and Conclusions
Future Directions

REFERENCES ................................................................................................. 125
LIST OF TABLES

Table

1. Gene expression data from four methods in yeast cell cycle ..................... 60
2. Description of 24 genes selected for gene regulatory network reconstruction .... 66
3. Comparison of Inferred GRNs at different phases of yeast cell cycle with the true network ................................................................. 71
4. Comparison of performance between different structure learning algorithms using synthetic dataset .................................................................................................................. 83
5. Comparison of performance between different structure learning algorithms using yeast benchmark dataset .................................................................................................................. 84
6. Significantly influenced probe by RDX treatment that can be mapped to nervous system-related KEGG pathways ........................................................................................................... 100
LIST OF ILLUSTRATIONS

Figure

1. Process of cDNA microarray experiment design ............................................. 12
2. Example of a M-A plot .................................................................................. 14
3. Intensity distribution of arrays before and after median normalization .............. 15
4. Spot intensity plots with different lowess window width .................................. 16
5. Comparison of KNN, SVD ............................................................................ 17
6. Volcano plot showing metabolomic data .......................................................... 19
7. Agglomerative clustering and divisive clustering ............................................... 20
8. An example of a Boolean network .................................................................. 26
9. Markov network ............................................................................................ 29
10. A Markov chain with 5 states with selected state transitions ......................... 34
11. A highly simplified DBN for monitoring a vehicle ......................................... 35
12. An order graph of four variables .................................................................. 51
13. Generating successors of a node in the order graph ....................................... 51
14. Process of time lag DBN ............................................................................... 53
15. The transcriptional time lag between the potential regulator and target gene .... 54
16. Sparse representation of the contingency table .............................................. 55
17. The algorithm to build a sparse Adtree ......................................................... 57
18. The number of interactions annotated for each experimental method ............. 61
19. Physical relationship from SGD ..................................................................... 67
20. Genetic relationship from SGD ..................................................................... 68
21. Location of protein components of the spindle pole body ............................ 69
22. True *Saccharomyces cerevisiae* GRNs built from SGD.................................70
23. Inferred GRNs at different cycle phase of *Saccharomyces cerevisiae*........71
24. Dynamic Bayesian Network structure learning algorithm using The Friedman’s Bayesian Information Criterion (BIC) score metrics........................................73
25. Generation and simulation of *in silico* gene network models using GNW. ..........77
26. A transition network of 10 genes learned by Friedman score metrics ...............78
27. The 50-gene network reconstructed by different algorithms..........................82
28. Comparison of performance between different structure learning algorithms using synthetic dataset.........................................................................................82
29. The real yeast network reconstructed by different algorithms..........................84
30. Array distributions to three treatments and 31 time points.................................91
31. Overview of study design and sampling scheme of the time-series earthworm toxicity study using the 44K-probe earthworm microarray........................................91
32. The percentage of genes significantly influenced by RDX or Carbaryl treatment in the total number of genes for each mapped KEGG pathway. .........................94
33. Genes altered by RDX treatment mapped to the long-term potentiation pathway...107
34. Genes altered by RDX treatment mapped to the Synaptic vesicle cycle pathway. 108
35. Genes altered by RDX treatment mapped to the Neurotrophin Signaling Pathway. 109
36. Genes altered by RDX treatment mapped to Glutamatergic Synapse Pathway......110
37. Genes altered by RDX treatment mapped to GABAergic synapse. .....................111
38. Genes altered by RDX treatment mapped to Dopaminergic synapse....................112
39. Genes altered by RDX treatment mapped to Long-term depression.....................113
40. Reconstructed GRNs mapped to Dopaminergic synapse: Control Exposure.......114
41. Reconstructed GRNs mapped to Dopaminergic synapse: Control Recovery .......115
42. Reconstructed GRNs mapped to Dopaminergic synapse: RDX Exposure ............116
43. Reconstructed GRNs mapped to Dopaminergic synapse: RDX Recovery ..............117

44. Reconstructed GRNs mapped to Dopaminergic synapse: Carbaryl Exposure .........118

45. Reconstructed GRNs mapped to Dopaminergic synapse: Carbaryl Recovery .........119
CHAPTER I
INTRODUCTION

Biological Background

*Genes*

Deoxyribonucleic acid (DNA) carries the genetic information of an organism, which is helix-shaped molecule whose constituents are two parallel strands of nucleotides. There are four types of nucleotides in DNA, which are adenine (A), thymine (T), cytosine (C) and guanine (G). Genes are contiguous pieces of DNA strand associated with certain functions in the living organism. DNA is stable, packaged and inert; and actively transcribe into a short-term form of nucleic acids, which is Ribonucleic acid or RNA. Varied types of RNA are essential for synthesis of proteins and regulation of gene expression, like messenger RNA (mRNA), transfer RNA (tRNA), ribosomal RNA (rRNA), MicroRNA(miRNA), small interfering RNAs(siRNA), etc. Messenger RNA (mRNA) carries information about a protein sequence, where every three nucleotides (a codon) correspond to one amino acid. There are 77 codons found to encode for up to 20 different amino acids. In eukaryotic cells, once mRNA has been transcribed from DNA, the mRNA is then exported from the nucleus to the cytoplasm, where it is bound to ribosome and translated into corresponding protein with the help of rRNA, which is a small RNA chain of about 80 nucleotides that transfers a specific amino acid to a growing polypeptide chain at ribosomal site during translation.

*Regulation of Gene Expression*

Of the 35,000 genes in the human genome, only a fraction is expressed in a cell at any given time. Regulation of gene expression includes the process that cells and viruses
use to regulate the way that the information is turned into gene products, which is essential for increasing the versatility and adaptability, and development of an organism by allowing the cell to express protein when needed. Any step of gene’s expression may be modulated, from DNA-RNA transcription to post-translational modification of a protein.

In eukaryotes, the accessibility of DNA regions depends on its chromatin structure, which can be altered as a result of histone modifications directed by DNA methylation, ncRNA, or DNA-binding protein. These modifications may up or down regulate the expression of a gene.

Regulation of transcription controls when transcription occurs and how much RNA is created by RNA polymerase. There are at least five mechanisms, (1) Specificity factors alter the specificity of RNA polymerase for a given promoter or set of promoters, making it more or less likely to bind to them; (2) Repressor bind to operator, which is sequence on the DNA strand that are close or overlapping the promoter region, impeding RNA polymerase’s progress along the strand, and impeding the expression of gene; (3) General transcription factors position RNA polymerase at start of protein-coding sequence and then release the polymerase to transcribe the mRNA; (4) Activators enhance RNA polymerase and promoter binding, and encouraging RNA translation; (5) Enhancers are sites on the DNA helix that are bound to activator in order to loop the DNA bringing a specific promoter to the initiation complex.

After the DNA is transcribed and mRNA is formed, there is also some sort of regulation on how much the mRNA is translated into proteins. Cells do this by modulating
the capping, splicing, addition of a Poly (A) Tail, and sequence-specific nuclear export rates.

**DNA Microarray**

Functional genomics involves the analysis of large datasets of information derived from various biological experiments. Large-scale experiment involves monitoring the expression levels of thousands genes simultaneously under a particular condition. Cheaper, higher-resolution, and high through-put technologies were developed over the past decades in genomics, transcriptomics and proteomics, which enable integrated understanding of molecular and cellular system. DNA microarray is a multiplex technology used in molecular biology, which is used to measure the expression levels of a large number of genes simultaneously or to genotype multiple regions of a genome. DNA microarray is a collection of microscopic DNA spots attached to a solid surface. Each DNA spot contains picomoles \((10^{-12} \text{ moles})\) of a specific DNA sequence, which can be a short section of a gene or other DNA element that are used to hybridize a cDNA or cRNA(also called anti-sense RNA) sample. Probe-target hybridization is usually detected and quantified by detection of fluorophore-, silver-, or chemiluminescence- labeled targets to determine relative abundance of nucleic acid sequences in the target.

Microarray technology evolved from Southern blotting, where fragmented DNA is attached to a substrate and then probed with a known gene or fragment. The use of a collection of distinct DNAs in arrays for expression profiling was first described in 1987, and the arrayed DNAs were used to identify genes whose expression is modulated by interferon [48]. These early gene arrays were made by spotting cDNAs onto filter paper with a pin-spotting device. The use of miniaturized microarrays for gene expression
profiling was first reported in 1995 [49], and a complete eukaryotic genome on a microarray was published in 1997 [50].

There exist many types of microarrays and they are differed by whether being spatially arranged on a surface or on coded beads. The early-stage array is a collection of orderly microscopic spots. Each spot is combined with a specific probe attached to a solid surface, such as silicon, glass or plastic biochip. The location of a certain probe has been arranged and thousands of these probes are placed on a single DNA microarray. On the alternative, bead array is a collection of microscopic polystyrene beads. A specific probe and a ratio of two or more dyes are combined with each bead. Thus, they do not interfere with the fluorescent dyes used on the target sequence. DNA microarrays technology can be used in many areas such as gene expression profiling, comparative genomic hybridization, chromatin immunoprecipitation on Chip (ChIP), SNP detection, alternative splicing detection [51,52] and etc. Further introduction of DNA microarray analysis is included in Chapter II.

There exist many challenges in microarray experiments analysis. Microarray data sets are normally of huge amount, and its analytical precision is influenced by several variables. Statistical challenges include effects of image background noise, whether appropriate normalization and transformation techniques are conducted, identification of significantly differentially expressed genes (DEGs) [52, 53, 54, and 55] as well as inference of gene regulatory networks [56]. How to reduce the dimensionality of microarray dataset in order to obtain more comprehension and focused analysis requires further preprocessing of microarray data [57].
Computational Analysis of Biological Networks

Reverse engineering is the process of elucidating the structure of a system by reasoning backwards from observations of its behavior. In the context of molecular biology, the reverse-engineering of biochemical networks from experimental data has become a central focus in systems biology. Gene regulatory networks reconstruction is one of the research areas in reverse engineering, which aims to reveal the causality of this process through activation or repression of targets by regulatory proteins. Mathematical and statistical tools have been developed for reverse engineering, where Gene regulatory networks is defined as a mixed graph over a set of nodes (corresponding to genes or gene activities) with directed or undirected edges (representing causal interactions or associations between gene activities) [1]. Various mathematical methods and computational approaches have been proposed to reconstruct GRNs, including Boolean networks [2], information theory [3, 4], differential equations [5] and Bayesian networks [6, 7, and 8]. GRN reconstruction faces huge intrinsic challenges on both experimental and theoretical fronts, because the inputs and outputs of the molecular processes are unclear and the underlying principles are unknown or too complex.

Dynamic Bayesian networks (DBNs) are belief networks that represent the stochastic process of a set of random variables over time. The hidden Markov model (HMM) and the Kalman filter can be considered as the simplest DBNs. However, Kalman filters can only handle unimodal posterior distributions and linear models, whereas parameterization of HMM grows exponentially with the number of state variables [10]. Several algorithms have been developed to learn structure for belief networks from both complete [6, 10, 11, 12] (without missing values) and incomplete
(with missing values) datasets. Structure Expectation-Maximization (SEM) has been developed for learning a probabilistic network structure from data with hidden variables and missing values [13]. A structure learning algorithm has also been developed for high-order and non-stationary dynamic probabilistic models [15].

Contributions

In this dissertation, we compared and improved different computational models for reconstructing gene regulatory network, summarized as below.

Evaluation of Dynamic Bayesian Networks

Reconstruction of gene regulatory networks (GRNs) is a challenging inverse problem. Most existing approaches have understandably low accuracy because of the intrinsic complexity of a biology system and a limited amount of available data. Dynamic Bayesian Network (DBN) is one of the widely used approaches to identify the signals and interactions within gene regulatory pathways of cells. However, the impacts of network topology, properties of the time series gene expression data, and the number of time points on the inference accuracy of DBN are still unknown or have not been fully investigated. In first part of our work, the performance of DBN is evaluated in terms of precision and recall using both in-silicon yeast data and three growth phases of Yeast Saccharomyces cerevisiae cell cycle data with different time points. 35 genes were randomly selected from the gene set, including four transcription factors: CLN3, CLB6, CLB2 and SWI4. These genes were known to be expressed in G1, S, G2, M and M/G1 using a standard Pearson correlation function. There are 18, 14 and 24 time points in phase 1, phase 2 and phase 3, respectively. The GRNs were inferred using DBN corresponding to different subsets of genes in three phases. The inferred GRNs were
compared with the actual GRN obtained from SGD (The Saccharomyces Genome Database). At meanwhile, to better understand the meanings of edges from SGD, which was used for evaluating network, genes were also selected from the literature related to particular functions in budding yeast cell cycle. 24 genes were selected from the Spellman microarray data set [19], including 14 genes that encode proteins for organelles, and 9 genes that are cell cycle regulators. Physical and Genetic relationships which is identified by SGD and also found literatures are identified.

**Improvement of Dynamic Bayesian Network (DBN)**

In previous work, we compared two very popular and powerful methods to model gene regulatory networks. The performance of Probabilistic Boolean network and dynamic Bayesian network is compared using a biological time series dataset from Drosophila Interaction Database to construct a Drosophila gene network [9]. The results show that dynamic Bayesian network is more accurate than Probabilistic Boolean network. However, Denis not computationally efficient and has low accuracy. To improve its performance, we implemented a more sophisticated score metrics for Dynamic Bayesian Network which was first proposed by Nir Friedman for stationary DBNs structure learning of both initial and transition networks, and tested it on reconstruction of both synthetic networks and real yeast networks and compared it with REVEAL.

**Dissertation Organization**

This dissertation is organized as follows: Chapter II introduces the backgrounds and technical details of microarray experiments, data processing methods, including transformation, normalization, etc. It also discusses microarray data analysis methods,
including statistical models to identify differentially expressed genes such as t-test, ANOVA, volcano plot for either two-class comparison or multi-class comparison, and machine learning methods such as clustering, classification methods.

Chapter III presents details of various mathematical and statistical models for computational analysis of biological networks, including Boolean networks, information theory, differential equations and Bayesian networks.

Chapter IV focuses on one particular computational model for infer gene regulatory networks, which is also fundamental of our method- Probabilistic Graphical Model. It discusses probabilistic graphical model representations, theories for parameter learning, and structure learning and also introduces the methodologies to improve current Dynamic Bayesian Network approach.

Chapter VI evaluates the performance of DBN using both in-silico yeast data and three growth phases of Yeast *Saccharomyces cerevisiae* cell cycle data with different time points in terms of precision and recall. The reconstructed GRNs were compared with the actual GRNs obtained from SGD.

Chapter VII presents a more sophisticated score metrics for Dynamic Bayesian Network which was first proposed by Nir Friedman for stationary DBNs structure learning of both initial and transition networks. K2 algorithm was modified and it implemented Friedman’s Bayesian Information Criterion (BIC) score function. It was tested on reconstruction of both synthetic networks and real yeast networks and its performance was compared with REVEAL in the absence or presence of preprocessed network generated by Zou and Conzen’s algorithm [17].
Chapter VIII identifies the mechanism of chemical-induced reversible neurotoxicity through reconstruction of gene regulatory networks in earthworms received different treatments (control, Carbaryl and RDX) during exposure and recovery phases, using Chapter VII algorithm combined tools mapping non-model organism’s pathway to model organism pathway.

Chapter IV concludes the dissertation by summarizing our work and providing a brief discussion of future work.
CHAPTER II

REVIEW OF MICROARRAY

DNA Microarray Experiment

**DNA Microarray Experiment**

Microarray evolved from Southern blotting, where fragmented DNA is attached to a substrate and then probed with a known gene or fragment. The use of a collection of distinct DNAs for expression profiling was first described in 1987. These early gene arrays were made by spotting cDNA onto filter paper with a spin-spotting device. The use of miniaturized microarray for gene expression profiling was first reported in 1995, and a complete eukaryotic genome on a microarray was published in 1997. High-throughput technologies are used to measuring the mRNA gene expression values simultaneously and these generated gene expression data are very crucial to identify diseased and discover gene expression patterns.

As Figure 1 shows, on the microarray, a known sequence cDNA or oligonucleotide DNA sequence is fixed on a glass, nylon, or quartz wafer, as probes. The array is then hybridized with sample RNA extracted from biology samples. The principle behind microarrays is hybridization between two DNA strands, so that the particular type of RNA in the sample is hybridized with the cDNA probe on microarray. After washing off of none-specific bonding sequences, only strongly paired strands will remain hybridized. Fluorescently labeled target sequences that bind to a probe sequence generate a signal, and the intensity of the signal is measured to quantify the relative amount of the probe sequence in the sample. In oligonucleotide microarrays, the probes are short sequences designed to match parts of the sequence of known or predicted Expressed
Sequence Tags (ESTs), which is a short sub-sequence of a transcribed cDNA sequence [59]. Oligonucleotide arrays are produced by printing short oligonucleotide sequences designed to represent a single gene or family of gene splice-variants by synthesizing this sequence directly onto the array surface instead of depositing intact sequences. Sequences may be longer (60-mer probes such as the Agilent design) or shorter (25-mer probes produced by Affymetrix) depending on the desired purpose; longer probes are more specific to individual target genes, shorter probes may be spotted in higher density across the array [60].

Two-color microarray are typically hybridized with cDNA prepared from two samples to be compared (e.g. diseased tissue versus healthy tissue) and that are labeled with two different fluorophores, such as Cy3, which has a fluorescence emission wavelength of 570nm (green), and Cy5 with a fluorescence emission wavelength of 670nm (red). Relative intensities of each spots can be used in ratio-based analysis to identify up regulated and down regulated genes. In single-channel or one-color microarrays, the array provides intensity data for each probe indicating a relative level of hybridization with the labeled target. However, this intensity data is not true indicator of abundance level of a gene, but rather a relative abundance when compared with other samples processed in the same experiment. In one-color array chip, an aberrant sample cannot affect the raw data derived from other samples. While in two-color array chip, a single low-quality sample may drastically impinge on the precision of overall data set even if the other sample is of high quality.
Image Processing Analysis

Image processing is needed in the next step to grid, spot recognize of the scanned image (segmentation algorithm), remove or mark of poor quality and low-intensity features. It involves the following steps: 1. Identification of the spots and distinguish them from spurious signals. 2. Determination of the spot area will be surveyed and determination of the local region is used to estimate background hybridization. 3. Reporting summary statistics and assigning spot intensity after subtracting for background intensity [9].

Data processing

Raw image data after image processing need to be further processed before using, which involves background subtraction, spot intensities and intensity ratios determination,
visualization of data, global or local normalization of intensity ratios, and segmentation into different copy number regions using step detection algorithms.

Microarray data is often normalized within arrays to control the systematic biases in dye coupling and hybridization efficiencies, as well as other technical biases in the DNA probes. By minimizing these systematic variations, true biological differences can be found. To determine whether normalization is needed, Cy5(G) intensities against Cy3(R) intensities are plotted to see whether the slope of the line is around 1. An improved method, MA-plot is a 45 degree rotation of the R vs. G plot, which can be interpreted as distribution of red/green intensity ratio (M) plotted by the average intensity (A). M and A are defined by the following equations.

\[
M = \log_2 \left( \frac{R}{G} \right) = \log_2 R - \log_2 G
\]

\[
A = \frac{1}{2} \log_2 (RG) = \frac{1}{2} (\log_2 R + \log_2 G)
\]

The MA plot gives a quick overview of the distribution of the data, as shown in Figure 2. In many microarray gene expression experiments, most of the genes would not show any change in their expression, where the majority of the points on the y axis (M) would be located at 0. If this is not the case, then a normalization method such as LOESS should be applied to the data before statistical analysis.
There are currently four major normalization options: median normalization, housekeeping gene normalization, lowess normalization as well as print-tip group normalization. The first two are available for both single-channel and dual-channel data, but the last two are only for dual-channel data.

**Median normalization**

For single-channel data, the user needs to choose a reference array against which all other arrays will be normalized. The “median” reference array is selected as following algorithm as showed in Figure 3:

1. Let N be the number of experiments, and let \( i \) be an index of experiments running from 1 to N.
2. For each array \( i \), the median log-intensity of the array (denoted \( M_i \)) will be computed.
(3) A median $M$ will be selected from the $\{M_1, ..., M_N\}$ values. If $N$ is even, then the median $M$ will be the lower of the two middle values.

(4) The array whose median log-intensity $M_i$ equals the overall median $M$ will be chosen as the median array.

Then, the median normalization is performed by subtracting out the median log-ration for each array, so that each normalized array has a median log-ration of 0. Such median normalization is called per-gene normalization. Besides, per chip normalization is performed by computing a gene-by-gene difference between each array and the reference array, and subtracting the median difference from the log-intensities on that array, so that the gene-by-gene difference between the normalized array and the reference array is 0.

Figure 3. Intensity distribution of arrays before (left) and after (right) median normalization.

Locally weighted linear regression (LOWESS)

For dual-channel data, locally weighted linear regression (LOWESS) normalization is normally used. In the lowess normalization, a non-linear lowess smoother function is fit to the graph of un-normalized log-ratio on the $y$-axis versus
average log intensity (i.e., \([\log(R)+\log(G)]/2\)) on the x-axis. That is, LOWESS normalization assumes that the dye bias appears to be dependent on spot intensity. The adjusted ratio is computed by the following Equation 2.1

\[
\log \frac{R}{G} \rightarrow \log \frac{R}{G} - c(A) \quad (2.1)
\]

Where \(c(A)\) is the lowess fit to the \(\log R/G\) versus \(\log \sqrt[2]{R \times G}\) plot.

LOWESS regression is a technique for fitting a smoothing curve to a dataset. The degree of smoothing is determined by the window width parameter. In general, a larger window width results in a smoother curve, while a smaller window results in local variation [40, 41, 42, 43]. Figure 4 shows the plots under different LOWESS window width.

\[\text{Figure 4. Spot intensity plots with different lowess window width.}\]

\textit{Missing Values}

If missing values are treated as intensity value of zero when calculated, it will certainly affect the accuracy and validity of analysis results. Therefore, methods for imputing missing data are needed to minimize the effect of incomplete data sets.
Previously, three most popular methods to impute missing values are proposed, namely, Singular Value Decomposition (SVD) based method (SVDimpute) [45, 46, 47], weighted K-nearest neighbors (KNNimpute) [44], as well as row average. The KNN-based method selects genes whose expression profiles are similar to the gene of interest to impute missing values. Suppose there is a missing value in experiment 1 for gene A, KNN impute will find K other genes whose expression values are most similar to A in experiments 2 to N. Euclidean distance, which is the metric for gene similarity is used during the imputing process. The row average technique is trivial as calculating the average of the row containing missing values and filling them with it. SVD impute method can only be performed on complete matrices, so row average is imputed for all missing values and then utilize an expectation maximization method to arrive at the final estimate. Troyanskaya et al. compared the above three missing value imputation techniques and KNN-based estimations turned out to have best performance among the three on the same data set as showed in Figure 5.

![Figure 5](image-url) 

*Figure 5. Comparison of KNN, SVD, and row average based estimations’ performance on the same data set [44].*
Microarray Analysis

Identification of Differentially Expressed Genes

In two channel-microarrays, genes with statistically significant changes can be identified, which is the interested gene that is influenced by the condition. There are many methods to identify the genes exhibiting the most significant variation, such as a fixed fold-change cut-off method, t-test, ANOVA, Mann-Whitney test, Z-score, and volcano plot.

Volcano plot as showed in Figure 6 combines a statistical test(e.g., p-value, ANOVA), is a type of scatter-plot that is used to quickly identify changes in large dataset, which is increasingly common in experiments such as genomics, proteomics, where often involves in thousands of replicate data points between two conditions. A volcano plot is constructed by plotting the negative log of the p-value (log based) on the y-axis. This results in data with low p-values (highly significant) appearing towards the top of the plot. The x-axis is the log of the fold change between the two conditions. In a result, those points that are found towards the top of the plot or far to either the left- or the right-hand side are the interested data points with significant changes.
Figure 6. Volcano plot showing metabolic data. The red arrows indicate points-of-interest that display both large-magnitude fold-changes (x-axis) as well as high statistical significance (-log10 of p-value, y-axis). The dashed red line shows where $p = 0.05$ with points above the line having $p < 0.05$ and points below the line having $p > 0.05$. This plot is colored such that those points having a fold-change less than 2 ($\log_2 = 1$) are shown in gray.

Hierarchical Clustering

Hierarchical clustering as showed in Figure 7 may be agglomerative (starting with the each object and grouping similar objects into bigger clusters) or divisive (starting from grouping all objects into one cluster and subsequently breaking the big cluster into smaller clusters with similar properties). For a hierarchical agglomerative clustering procedure, the process begins at each object. In each step, pairwise distances between them are calculated to group similar objects into group. The distance between clusters can be measured using single linkage clustering (minimum distance), complete linkage clustering (maximum distance), average linkage clustering and centroid linkage clustering. For a hierarchical divisive clustering procedure, entire set of objects is
considered as a single cluster and is broken down into two or more clusters that have similar expression profiles.

**Figure 7.** Agglomerative clustering and divisive clustering. (source from http://www.mrc-lmb.cam.ac.uk/genomes/madanm/microarray)
CHAPTER III

COMPUTATIONAL APPROACHES TO INFER GRNS

A variety of statistical and mathematical methods have been studied to find either associative or causative interactions or dependencies among gene products. In this type of analysis, a network of molecular interactions is viewed as a directed graph: a pair \((V,E)\) where \(V\) is a set of vertices (or nodes) and \(E\) a set of directed edges, i.e. pairs \((i,j)\) of nodes, where \(i\) is the source node and \(j\) is the target node. This chapter summarizes several existing methods for gene regulatory network reconstruction.

Differential Equations

As an important mathematical model in computational biology, differential equations can model complex dynamic behavior such as oscillations, cyclical patterns, multi-stationary and switch-like behavior. For computational biologists, the first step is to find differential equations which can represent the system under study precisely. It is essential to know about the processes in the system and to have large amounts of data available to infer the unknown parameters in gene regulatory networks. Necessary foundations for a good description with differential equations would be the knowledge about which gene regulates another and in which way, as well as knowledge about the degradation and maximal production rates of the associated proteins. Differential equations describe gene expression changes as a function of the expression of other genes and environmental factors. Thus, they are adequate to model the dynamic behaviors of gene regulatory networks in a more quantitative manner. In general, if the gene regulatory networks are without constraints, there are multiple solutions, i.e. the ODE system is not uniquely identifiable from gene expression data achieved from experiments.
In recent years, some more complex variants of differential equation models have been proposed to describe complicated gene regulatory networks, such as stochastic differential equations that are incorporated with the stochastic of gene expression, which may be very useful especially in transcriptional regulatory networks [20, 21].

**Linear Differential Equations**

Modeling biological data with linear differential equations was first considered theoretically by Chen [22]. In this model, both the mRNA and protein concentrations were described by a system of linear differential equations. Such a system can be described as following:

$$\frac{dx_i}{dt} = \sum_{j=1}^{n} w_{i,j} \cdot x_j + b_i \cdot s, \text{ where } i = 1, 2, \ldots, n \quad (3.1)$$

where $w_{i,j}$ is the $i,j$ entry of the interaction matrix $W$, and $b_i$ quantify the impact of the perturbations on gene expression.

In order to solve linear differential equations by well-established methods for linear algebra, linear differential Equation (3.1) becomes the linear difference Equation (3.2)

$$\frac{x_i[t + \Delta t] - x_i[t]}{\Delta t} = \sum_{j=1}^{n} w_{i,j} \cdot x_j + b_i \cdot s, \text{ where } i = 1, 2, \ldots, n \quad (3.2)$$

By this way, we can obtain a linear algebraic equation system that can be solved by methods of linear algebra, in which singular value decomposition (SVD) [23, 24] and regularized least squares regression methods are the most prominent ones used to solve the linear equation system with the constraint of sparseness of the interaction matrix. For instance, the LASSO (Least Absolute Shrinkage and Selection Operator) provides a
robust estimation of a network with limited connectivity and low model prediction error [25]. Further inference algorithms based on linear difference equation models are NIR (Network Identification by multiple Regression [26]), MNI(Microarray Network Identification[27]) and TSNI(Time-Series Network Identification[28]).

Non-linear Ordinary Differential Equations

Ordinary differential equations (ODEs) have been widely used to analyze gene regulatory networks, and it is probably the most popular formalism to model dynamical systems in computational biology. The ODE formalism models the concentrations of RNA, proteins, and other elements of the system by time-dependent variables with values contained in the set of non-negative real numbers. Regulatory interactions take the form of functional and differential relations between the concentration variables.

More specifically, gene regulation is modeled by reaction-rate equations expressing the rate of production of a gene product (a protein or an mRNA ) as a function of the concentrations of other components of the system. Reaction-rate equations have the following mathematic form:

$$\frac{dx_i}{dt} = f_i(x), x_i > 0, 1 < i < n \quad (3.3)$$

where $x$ is the vector of concentrations of proteins, mRNA, or small metabolites, and $f_i$ usually is a nonlinear function. The rate of synthesis of $i$ is consider to be dependent upon the concentrations $x$, possibly including $x_i$. The equations can be extended to take into account concentrations of $s>0$ input elements, e.g. externally-supplied nutrients, as given in Equation (3.4):

$$\frac{dx_i}{dt} = f_i(x, s), x_i > 0, 1 < i < n \quad (3.4)$$
They may also take into account discrete time delays arising from time required to complete transcription, translation, and diffusion to the place of action of a protein.

The identification of non-linear models is not only limited by mathematical difficulties and computational efforts for numerical ODE solution and parameter identification, but also mainly by the fact that the sample size $M$ is usually too small for the reliable identification of non-linear interactions. Thus, the search space for non-linear model structure identification has to be stringently restricted. For that reason, inference of non-linear systems employ predefined functions that reflect available knowledge.

*Partial Differential Equations*

The gene regulatory networks are implicitly assumed to be spatially homogeneous. There are some cases in which these assumptions are not correct. It might be necessary, for instance, to distinguish between different compartments of a cell, say the nucleus and the cytoplasm, and to consider diffusion of regulatory proteins or metabolites from one compartment to another. Moreover, gradients of protein concentrations across cell tissues are critical features in embryonic development. The introduction of time delays for diffusion effects allow some aspects of spatial inhomogeneities to be dealt with, where preserving the basic form of the reaction-rate equations [30]. However, in the case that multiple compartments of a cell, or multiple cells, need to be explicitly modeled.

Suppose that a multi-cellular regulatory system is considered, where $p$ cells are arranged in a row. A new vector $\mathbf{x}^{(l)}(t)$ is introduced, in which the time-varying concentration of gene products is denoted in cell $l$, $l$ is a discrete variable ranging from 1 to $p$. Within each cell, regulation of gene expression occurs in the manner described by Equation (3.5). Between a pairs of adjacent cells $l$ and $l+1$, diffusion of gene products is
assumed to take place proportional to the concentration differences $x_i^{(l+1)} - x_i^l$, $x_i^{(l-1)}$ and a diffusion constant $\delta_i$. Taken together, this leads to a system of coupled ODEs, so-called reaction-diffusion equations:

$$\frac{dx_i}{dt} = f_i(x_i) + \delta_i \left( x_i^{(l+1)} - 2x_i^l + x_i^{(l-1)} \right), x_i^l > 0, 1 < i < n(3.5)$$

Reaction-diffusion equations and partial differential equations have been used in computational biology to study pattern formation in development, such as [31,32]. The induction of models from measurements of at a sequence of time-points is made attractive by the growing availability of gene expression data. However, precise measurements of absolute expression levels are currently difficult to achieve. In addition, as a consequence of the dimensionality problem, the models need to be simple and are usually strong abstractions of biological processes [33]. For larger and more complicated models, the computational costs of finding an optimal match between the parameter values and the data may be extremely high.

Boolean Network

The first Boolean networks were proposed by Stuart A. Kauffman in 1969, as random models for genetic regulatory networks. A Boolean network $G(V,F)$ is defined by a set of nodes (variables) representing genes $V=\{x_1, x_2, \ldots x_n\}$ (where $x_i \in \{0,1\}$ is a binary variable) and a set of Boolean functions $F=\{ f_1, f_2, \ldots f_n \}$, which represents the transitional relationships between different time points. A Boolean function $f(x_{f_1(t)}, x_{f_2(t)}, \ldots x_{f_k(i)(t)})$ with $k(i)$ specified input nodes is assigned to node $x_i$. The gene status (state) at time point $t+1$ is determined by the values of some other genes at previous time point $t$ using one Boolean function $f_i$ taken from a set of Boolean functions $F$. So we can define the transition as
\[ x_{i}(t + 1) = f(x_{f_{1}(t)}x_{f_{2}(t)} \ldots x_{f_{k}(t)}) \]

where each \( x_i \) represents the expression value of gene \( i \), if \( x_i = 0 \), gene \( i \) is inhibited; if \( x_i = 1 \), it is activated. The variable \( j_{k(i)} \) represents the mapping between gene networks at different time points. Boolean function \( F \) represents the rules of regulatory interactions between genes. An example of a Boolean network is given in Figure 8. The connected graph is represented by (a), and the transition function is defined by (b).

![Figure 8](image)

*Figure 8.* An example of a Boolean network: (a) the wiring diagram; (b) the updating rules; (c) a state transition table; (d) the state space of the network.

**Probabilistic Graph**

Generally, probabilistic graphical models use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution. Two branches of graphical representations of distributions are commonly used, namely, Bayesian networks and Markov networks. Both families encompass the properties of factorization and independences, but they differ in the set of
independences they can encode and the factorization of the distribution that they induce.

The representation of probabilistic graphical models is detailed in Chapter IV.
CHAPTER IV
PROBABILITY GRAPHICAL MODELS REPRESENTATION

Generally, probabilistic graphical models use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution. Two branches of graphical representations of distributions are commonly used, namely, Bayesian networks and Markov networks. Both families encompass the properties of factorization and independences, but they differ in the set of independences they can encode and the factorization of the distribution that they induce. This chapter introduces the representation of probabilistic graphical models.

Undirected Graphical Models: Markov Networks

A Markov random field (often abbreviated as MRF), Markov network or undirected graphical model is a set of random variables having a Markov property described by an undirected graph. A Markov random field is similar to a Bayesian network in its representation of dependencies; the differences being that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); On the other hand, it cannot represent certain dependencies that a Bayesian network can(such as induced dependencies). An example of Markov network is shown in Figure 9.
Bayesian Network Representation

**Local Independencies**

A Bayesian network structure $\mathcal{G}$ is a directed acyclic graph whose nodes represent random variables $X_1, \ldots, X_n$. Let $\text{Pa}_{X_i}^\mathcal{G}$ denote the parents of $X_i$ in $\mathcal{G}$, and $\text{NonDescendants}_{X_i}$ denote the variables in the graph that are not descendants of $X_i$. Then $\mathcal{G}$ encodes the following set of conditional independence assumptions, called the local independencies, and denoted by $I_\ell(\mathcal{G})$:

For each variable $X_i$, where $X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}_{X_i}^\mathcal{G}$

**I-Maps**

Let $P$ be a distribution over $\mathcal{X}$. We define $I(P)$ to be the set of independence assertions of the form $(X \perp Y | Z)$ that hold in $P$. We can now rewrite the statement that “$P$ satisfies the local independencies associated with $\mathcal{G}$” simply as $I_\ell(\mathcal{G}) \subseteq I(P)$. In this case, we say that $\mathcal{G}$ is an I-map (independency map) for $P$. That is to say, as $\mathcal{G}$ is an I-map for $P$, we have that $(X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}_{X_i}^\mathcal{G}) \in I(P)$. Let $K$ be any graph associated with a set of independencies $I(K)$. $K$ is an I-map for a set of independencies $I$ if $I(K) \subseteq I$. 
I-Maps Factorization

A BN structure $\mathcal{G}$ encodes a set of conditional independence assumptions. Consider the distribution $P$ for which Student BN $\mathcal{G}_{\text{student}}$ is an I-map for our distribution $P$.

$$P(I, D, G, L, S) = P(I) P(D|I) P(G|I,D) P(L|I,D,G) P(S|I,D,G,L)$$

This transformation relies on no assumptions and it holds for any joint distribution $P$. This form can be simplified by the conditional independence assumptions induced from the BN. Such that we have $(D \perp I) \in I(P)$. We can conclude that $P(D|I) = P(D)$; and $(L \perp I, D|G) \in I(P)$. Hence, $P(L|I,D,G) = P(L|G)$

$$P(I, D, G, L, S) = P(I) P(D) P(G|I,D) P(L|G) P(S|I)$$

Factorization: Let be a BN graph over the variables $X_1, \ldots, X_n$. We say that a distribution $P$ over the same space factorizes according to $\mathcal{G}$ can be expressed as a product

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\text{Pa}_G^C) \quad (4.1)$$

This is also called the chain rule for Bayesian networks.

Theorem: Let $\mathcal{G}$ be a BN structure over a set of random variables $X$, and let $P$ be a joint distribution over the same space. If $G$ is I-map for $P$, $P$ factorizes according to $G$.

**D-separation, Soundness, Completeness, and Faithful**

The aim of d-separation is to understand the situation when we can guarantee that an independence $(X \perp Y|Z)$ holds in a distribution associated with a BN structure.

Direct connection: It can be interpreted as, if $X$ and $Y$ are directly connected, we can always get examples where they influence each other, regardless of $Z$. 
Undirected connection: If X and Y are not directly connected, the situation will be more complicated. Z is a trial between X and Y, then the observation of dependence of X and Y will be influenced by whether or not Z is observed.

Causal trial $X \rightarrow Z \rightarrow Y$: X and Y is dependent if and only if Z is not observed

Evidential trial $X \leftarrow Z \leftarrow Y$: X and Y is dependent if and only if Z is not observed

Common cause $X \leftarrow Z \rightarrow Y$: X and Y is dependent if and only is Z is not observed

Common effect $X \rightarrow Z \leftarrow Y$: X and Y is dependent if and only is either Z or one of Z’s descendants is observed.

d-separation: Let X, Y, Z be three sets of nodes in $\mathcal{G}$. We say that X and Y are d-separated given Z, denoted $d$-sep$_G(X; Y|Z)$, if there is no active trail between any node $X \in X$ and $Y \in Y$, given Z.

We use $I(\mathcal{G})$ to denote the set of independencies that correspond to d-separation:

$$I(\mathcal{G}) = \{(X \perp Y|Z): d-sep_G(X; Y|Z),\}$$

This is also called global Markov independence.

Soundness: if we find that two nodes X and Y are d-separated, given some Z, we are guaranteed that they are, in fact, conditionally independent given Z.

Completeness: d-separation detects all possible independencies. More precisely, if we have that two variables X and Y are independent given Z, then they are d-separated.

Faithful: A distribution P is faithful to G if, whenever $(X \perp Y|Z) \in I(P)$, then $d-sep_G(X; Y|Z)$. In other words, any independence in P is reflected in the d-separation properties of the graph.

Theorem: Let $\mathcal{G}$ be a BN structure. If X and Y are not d-separated given Z in $\mathcal{G}$, then X and Y are dependent given Z in some distribution P that factorizes over $\mathcal{G}$. 
**I-Equivalence**

Considering, for example, the three networks in figure 3.5 a, b, c, all of them encode precisely the same independence assumptions($X \perp Y|Z$).

I-equivalence of two graphs immediately implies that any distribution $P$ that can be factorized over one of these graphs can be factorized over the other.

Skeleton: The skeleton of a Bayesian network graph $\mathcal{G}$ over $X$ is an undirected graph over $X$ that contains an edge{$X,Y$} for every edge($X,Y$) in $\mathcal{G}$.

Theorem: Let $G_1$ and $G_2$ be two graphs over $X$. If $G_1$ and $G_2$ have the same skeleton and the same set of v-structures then they are I-equivalent.

**Immorality:** A v-structure $X \rightarrow Z \leftarrow Y$ is an immorality if there is no direct edge between $X$ and $Y$. If there is such an edge, it is called a covering edge for the v-structure.

Theorem: Let $G_1$ and $G_2$ be two graphs over $X$. Then $G_1$ and $G_2$ have the same skeleton and the same set of immoralities if and only if they are I-equivalent.

**Minimal I-Maps**

Complete graph is an I-map for any distribution, where $I(P) \subseteq \Phi$, yet it does not reveal any of the independence structure in the distribution.

Minimal I-Maps: A graph $K$ is a minimal I-map for a set of independencies $I$ if it is an I-map for $I$, and if the removal of even a single edge from $K$ renders it is not an I-map.

**Perfect Maps**

Definition: We say that a graph $K$ is a perfect map for a set of independencies $I$ if we have that $I(K) = I$. We say that $K$ is a perfect map for $P$ if $I(K) = I(P)$.
The definition means that if we can obtain a graph $G$ that is a perfect map for a distribution $P$, then we can read the independencies in $P$ directly from $G$. Not every distribution has a perfect map. Also, a perfect map of a distribution, if one exists, is generally not unique. A distribution $P$ can have many perfect maps, but all of them are $I$-equivalent.

Template-Based Representations

In a temporal setting, we wish to represent distributions over systems whose state changes over time. We use $X_i^{(t)}$ to represent the instantiation of the variable $X_i$ at time $t$. Note that $X_i$ itself is no longer a variable that takes a value; rather, it is a template variable. This template is instantiated at different points in time $t$, and each $X_i^{(t)}$ is a variable that takes a value in $\text{Val}(X_i)$.

Hidden Markov Networks

A hidden Markov model (HMM), which is developed by L.E. Baum, et al [34], is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved(hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. Figure 10 show an example of hidden Markov model.

Consider a system which may be described at any time as being in one of a set of $N$ distinct states, $S_1, S_1, \ldots S_N$. At regularly spaced discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. For the special case of a discrete, first order, Markov chain, the probabilistic description follows Markovian assumption, which is elaborated in section 4.3.2.
Figure 10. A Markov chain with 5 states with selected state transitions [100].

Dynamic Bayesian Networks

Markov assumption (or Markovian): We say that a dynamic system over the template variables X satisfies the Markov assumption if, for all \( t > 0 \),

\[
(X_{i}^{(t+1)} \perp X_{i}^{(0:t-1)} | X_{i}^{(t)})
\]

The Markov assumption would correspond to the constraint on the graph that there are no edges into \( X_{i}^{(t+1)} \) from variables in time slices \( t-1 \) or earlier.

\[
P(X^0, X^1, \ldots, X^T) = P(X^0) \prod_{t=0}^{T-1} P(X^{t+1} | X^t)
\]

Stationary dynamic system: We say that a Markovian dynamic system is stationary (also called time invariant or homogeneous) if \( P(X^{t+1} | X^t) \) is the same for all \( t \). In this case, we can represent the process using a transition model \( P(X' | X) \).

Dynamic Bayesian Network is based on the two assumptions above, which makes the probability distribution over infinite trajectories very compact: we need only represent the initial state distribution and the transition model \( P(X' | X) \).
Taking an example, the current observation depends on the car’s location and on the error status of the sensor. Bad weather makes the sensor more likely to fail, and the car’s location depends on the previous position and the velocity. So we can have a dynamic Bayesian network, as show in Figure 11.

Figure 11. A highly simplified DBN for monitoring a vehicle: (a) the 2-TBN (b) the time0 networks ; (c) resulting unrolled DBN over three time slice[100].

This type of conditional Bayesian network is called a 2-time-slice Bayesian network (2TBN). The simplest example of a temporal model of this kind is the hidden Markov model. It contains only a single variable S and a single observation variable O.
CHAPTER V

STRUCTURE LEARNING IN BAYESIAN NETWORKS

Because Dynamic Bayesian Network is derived from Bayesian Network, the learning theory has similarities. Thus for a better understanding, we discuss Bayesian learning first. First of all, two distinct learning approaches, constraint-based structure learning and score-based structure learning are compared. Since the score-based structure learning is outperformed by the constraint-based structure learning, so we focus on the latter and describe two fundamental parameter-estimation theories maximum likelihood estimation and Bayesian approaches, how they are derived, the relationship between them, and how prior influences the score function. Following that, we discuss Bayesian information criterion (BIC) score and Bayesian Dirichlet Equivalent (BDe) score, and finally give out the counterparts in Dynamic Bayesian Network.

Constraint-based structure learning treats a Bayesian network as a representation of independencies. They search for a witness to the separation between every pair of variable, and used to determine whether the two parents in a v-structure are conditionally independent. The independent test to define a measure of deviance from the null hypothesis can be either based on $\chi^2$ statistic or mutual information. This kind of method is sensitive to the decision rules. A liberal decision rule that accepts many data risks accepting ones that do not satisfy the hypothesis, while a conservative rule that rejects many risks of rejecting ones that satisfy the hypothesis. More detail about this approach is given in the book *Probabilistic Graphical Models* [100].

Comparing to the constraint-based structure learning, score-based structure learning does not have the threshold problem. The scoring function is a statistical
model that measures how well the model fits the observed data. *Score-based structure learning* address learning as a model selection problem, which is aim to finding the model with the highest score that fits the data set best. In general, however, the process is NP-hard, and heuristic search techniques are usually used.

Parameter estimation is used to recover the parameters of the networks while the networks structure is fixed. However it is fundamental to generate the scoring function in Bayesian Learning. There are two main approaches to dealing with the

**Constraint-Based Method**

The general framework of constraint-based approaches is to find the best minimal I-map for the domain, which could answer the question, such as “Does P satisfy \(X_1 \perp X_2, X_3|X_4\)?”. In learning the parents of \(X_i\), the algorithm poses independence queries of the form \(X_i \perp \{X_1, \ldots X_{i-1}\} - U|U\). Building minimum I-map needs to examine all the \(2^{i-1}\) possible subsets of \(X_1, \ldots X_{i-1}\).

**Independent Tests**

In hypothesis testing, the null hypothesis is “the data were sample from a distribution \(P^*(X, Y) = P^*(X) \ast P^*(Y)\). In the discrete-valued case, we expect that the counts \((M(x, y)\) in the data are close to \(M \cdot \hat{P}(x) \cdot \hat{P}(y)\)(where M is the number of samples). We can measure the deviance of the data from \(H_0\) defined in terms of these distances.

\[
d_{x^2}(D) = \sum_{x,y} \frac{(M(x,y) - M \cdot \hat{P}(x) \cdot \hat{P}(y))}{M \cdot \hat{P}(x) \cdot \hat{P}(y)}
\]

A data set that perfectly fits the independence assumption has \(d_{x^2}(D) = 0\), and a data set where the empirical and expected counts diverge significantly has a larger value.
Another potential deviance measure for the same hypothesis is the mutual information, which can be written as.

$$d_{I} (D) = I_{\tilde{p}_{0}} (X; Y) = \frac{1}{M} \sum_{x,y} M(x, y) \log \frac{M(x, y)}{M(x)M(y)}$$

We can set an agreed cutoff to accept the hypothesis for the $\chi^2$ statistic or the empirical mutual information.

If we want to test whether $X$ and $Y$ are independent given $Z$, $H_0$ is $P^{*} (X, Y, Z) = P(Z) \cdot P(X|Z) \cdot P(Y|Z)$.

$$d_{\chi^2} (D) = \sum_{x,y} \frac{(M(x, y, z) - M \cdot P(z) \cdot P(x|z) \cdot P(y|z))}{M \cdot P(z) \cdot P(x|z) \cdot P(y|z)}$$

The counterpart of mutual information to test the same hypothesis is that $l(X; Y|Z) = H(X|Z) - H(X|Z, Y)$, where $H$ is the entropy, $H(X) = -p \log p - (1 - p) \log P$.

Finding Perfect Map

Lemma 1: Let $G^{*}$ be a perfect map of a distribution $P$, and let $X$ and $Y$ be two variables such that $X \rightarrow Y$ is in $G^{*}$. Then $P$ does not have $X \perp Y|U$ for any set $U$.

This lemma implies that if $X$ and $Y$ are directed connected in $G^{*}$, all conditional independence queries that involve both of them would fail.

Lemma 2: Let $G^{*}$ be an I-map of a distribution $P$, and let $X$ and $Y$ be two variables that are not directed connected in $G^{*}$. Then either $P$ has $X \perp Y|Pa_{X}^{G^{*}}$ or $P$ has $X \perp Y|Pa_{Y}^{G^{*}}$.

Thus, if $X$ and $Y$ are not directed connected in $G^{*}$, we can find a set $U$ so that $X \perp Y|U$. We call this set $U$ a witness of their independence. Thus, we can now
construct an algorithm for building a skeleton of $G^*$. For each pair of variables, we consider all potential witness sets and test for independence. If we do not find a witness, we conclude that the two variables are directed connected in $G^*$ and add them to the skeleton.

According to Lemma 2, in the witness test, we can restrict the potential witness set $U$ to $\mathcal{X}-\{\mathcal{X}_i, \mathcal{X}_j\}-\text{Nb}^{\mathcal{X}_i}_{\mathcal{X}_i}$ and $\mathcal{X}-\{\mathcal{X}_i, \mathcal{X}_j\}-\text{Nb}^{\mathcal{X}_j}_{\mathcal{X}_j}$, where $\text{Nb}^{\mathcal{X}_i}_{\mathcal{X}_i}$ are the neighbors of $\mathcal{X}_i$ in the current graph and $\text{Nb}^{\mathcal{X}_j}_{\mathcal{X}_j}$ are the neighbors of $\mathcal{X}_j$.

**Score Based Method**

**Maximum Likelihood Estimation in Bayesian Networks**

The likelihood function is generated to find the best-fit parameter values given an observed sequence, where parameter values with higher likelihood are more likely to generate the observed sequence.

The widely used thumbtack-tossing example for one parameter estimation gives out a good sense about the theory. If a tosses sequence $H,T,T,H,H$ is observed, how to estimate parameter $\theta$, which is the probability of head? The probability to observe the particular sequence is

$$P((H,T,T,H,H): \theta) = \theta(1-\theta)(1-\theta)\theta\theta = \theta^3(1-\theta)^2$$

For different values of $\theta$, we get different probabilities for the sequence, and we want to find the parameter $\theta$ which maximize the probability above. Thus in this case, the likelihood function can be defined as

$$L(\theta: (H,T,T,H,H)) = P((H,T,T,H,H): \theta) = \theta(1-\theta)(1-\theta)\theta\theta = \theta^3(1-\theta)^2$$

Next, we can move to solve the general problem of estimating parameters for a Bayesian network. In this case, our network is parameterized by a parameter vector $\theta$. 
which defines the conditional probability table in the network. We begin with a simple example of two binary variables $X|Y$ with a network $X$ is parent of $Y$. In this particular case, the likelihood function is derived as:

$$L(\theta; D) = \prod_{m=1}^{M} P(x, y; \theta)$$

$$= \prod_{m=1}^{M} P(x; \theta)P(y|x; \theta)$$

$$= \prod_{m=1}^{M} P(x; \theta) \prod_{m=1}^{M} P(y|x; \theta)$$

$M$ is the number of observation. In this example, $\theta_{x=1}$ and $\theta_{x=2}$ specify the probability of $X$; and $\theta_{y=1|x=0}$, $\theta_{y=1|x=1}$, $\theta_{y=0|x=1}$, $\theta_{y=0|x=0}$ specify the probability of $Y$ given $X$.

After giving two simple examples, we can give out the derived Maximum Likelihood Estimation (MLE) for a Bayesian network. Suppose we want to learn the parameters for a Bayesian network with structure $G$ and parameters $\theta$, by given the observations $D$. Our goal is to find both a graph and parameters that maximize the likelihood. The likelihood function of a specified structure can be:

$$L(\theta; D) = \prod_{m=1}^{M} P(D; \theta)$$

$$= \prod_{m=1}^{M} \prod_{i=1}^{N} P(x_i|Pa_{x_i}; \theta)$$

$$= \prod_{i=1}^{N} \left[ \prod_{m=1}^{M} P(x_i|Pa_{x_i}; \theta) \right]$$
N is the number of variables and M is the number of observation. It shows that the likelihood function can be decomposed as a product of independent terms. Cooper[11] proves that each local likelihood function can be maximized independently of rest of the network, and then combine the solutions to get an MLE solution.

Now we discuss the relationship between Maximum Likelihood Estimation and Mutual Information, which is a measurement of independence between variables. Return to the two variable case X and Y. Consider the model $G_0$ where X and Y are independent, we can get the likelihood as

$$L(G_0:D) = \log \prod_{m=1}^{M} P(x, y: \theta)$$

$$= \log \prod_{m=1}^{M} P(x: \theta)P(y: \theta)$$

$$= \sum_{m=1}^{M} (\log P(x: \theta) + \log P(y: \theta))$$

On the other hand, we consider model $G_1$ where X is parents of Y, and get likelihood as

$$L(G_1:D) = \log \prod_{m=1}^{M} P(x, y: \theta)$$

$$= \log \prod_{m=1}^{M} P(x: \theta)P(y|x: \theta)$$

$$= \sum_{m=1}^{M} (\log P(x: \theta) + \log P(y|x: \theta))$$

Let $\hat{P}$ be the empirical distribution observed in the data, we have

$$L(G_1:D) - L(G_0:D) = M \sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(y|x)}{\hat{P}(y)} = M I(X; Y)$$
One can see that the likelihood of the model $G_1$ depends on the mutual information between X and Y. Thus, higher likelihood implies stronger dependency of X and Y.

*Bayesian Parameter Estimation in Bayesian Networks*

*Bayesian Parameter Estimation* is an improvement of Maximum Likelihood Estimation by treating each observation ahead of current one as a prior.

Recall the thumbtack example of MLE, where we treat tosses as independent of each other, so that $\theta$ is fixed for each toss. If we do not know $\theta$, like we don’t know whether the toss is biased or not, then the tosses are not marginally independent, and each toss tells us something about the parameter $\theta$. In this case, we give an alternative, where we integrate the posterior over $\theta$ to predict the probability of the heads for the next toss.

$$P(x_{m+1}|x_1, x_2, \ldots, x_m)$$

$$= P(x_{m+1}|\theta, x_1, x_2, \ldots, x_m) P(\theta|x_1, x_2, \ldots, x_m) d\theta$$

As you can see $P(\theta|x_1, x_2, \ldots, x_m)$ is the *likelihood estimation* of the parameter. $P(x_{m+1}|x_1, x_2, \ldots, x_m)$ is also called the *marginal likelihood*. *Marginal likelihood* is quite different from the maximum likelihood score. Both examine the likelihood of the data given the structure. The maximum likelihood score returns the maximum of the function. In contrast, the marginal likelihood is the average value of this function.

A commonly used non-uniform priors is *Beta distribution*, and *Dirichlet distribution* as its counterparts of a multinomial distribution.

A Beta distribution is parameterized by two hyperparameters $\alpha_1, \alpha_0$, which are positive reals. The distribution is defined as follows:

$$\theta \sim \text{Beta}(\alpha_1, \alpha_0)$$
\[
p(\theta) = \frac{\Gamma(\alpha_1 + \alpha_0)}{\Gamma(\alpha_1)\Gamma(\alpha_0)} \theta^{\alpha_1-1}(1 - \theta)^{\alpha_0-1}
\]

where \(\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt\) is the Gamma distribution, which satisfy the properties \(\Gamma(1) = 1\) and \(\Gamma(x + 1) = x\Gamma(x)\); where factorial is an example of Gamma distribution.

Intuitively, the hyperparameters \(\alpha_1\) and \(\alpha_0\) correspond to the number of imaginary heads and tails that we have “seen” before starting the experiment.

It can be proved that if the prior is a Beta distribution, then the posterior distribution, that is, the prior conditioned on the evidence, is also a Beta distribution. Thus we may give the solution to the Bayesian Parameter Estimation with Beta distribution for the single parameter estimation as

\[
P(x_{m+1}|x_1, x_2, \ldots x_m) = \frac{\alpha_1 + 1}{\alpha_0 + M}
\]

Now we give the Bayesian Parameter Estimation or Bayesian score for a more general Bayesian network.

Given a Bayesian network with structure G and parameters \(\theta\), by giving the observations D, we define a structure prior \(P(G)\) that implies a prior probability on different graph structures, and a parameter prior \(P(\theta_G|G)\), that puts a probability on different choice of parameters once the graph is given. By Bayes rule, we have,

\[
P(G|D) = \frac{P(D|G)P(G)}{P(D)}
\]

where the denominator is simply a normalized factor. Thus, we define the Bayesian score as:

\[
score_B(G: D) = \log P(D|G) + \log P(G)
\]
where

\[ P(D|G) = \int_{\theta_G} P(D|\theta_G, G)P(\theta_G|G)d\theta_G \]

where \( P(D|\theta_G, G) \) is the likelihood of the data given the network \( \langle G, \theta_G \rangle \) and \( P(\theta_G|G) \) is our prior distribution over different parameter values for the network \( G \), which can be given as

\[ P(D|G) = \prod_{m=1}^{M} P(\zeta_m|\zeta_1, \zeta_2, ..., \zeta_m, G) \]

where \( \prod_{m=1}^{M} P(\zeta_m|\zeta_1, \zeta_2, ..., \zeta_m, G) \) is the probability of the \( m \)’th instance using the parameters learned from the first \( m-1 \) instances.

Bayesian score likes to bias toward simpler structures, but tends to recognize a more complex structure as observation grows.

It can be proved that if we use a Dirichlet distribution prior for all parameters in our network, then, when \( M \rightarrow \infty \), we have

\[ \log P(D|G) = l(\theta_G:D) - \frac{\log M}{2} \text{Dim}[G] + O(1) \]

where \( \text{Dim}[G] \) is the model dimension, or the number of independent parameters in \( G \).

This approximation is called the Bayesian information criterion (BIC).

\[ \text{SCORE}_{\text{BIC}}(G:D) = l(\theta_G:D) - \frac{\log M}{2} \text{Dim}[G] \]

The score exhibits a correction of over fitting problem of Bayesian scores. The stronger the dependence of a variable on its parents, the higher the score; the more complex the network, the lower the score.
Another widely used representation of score with Dirichlet distribution prior is Bayesian Dirichlet Equivalent (BDe) score. Because it is much more related to the nature of Dirichlet distribution, we need to expand it a little bit more.

As we said, it can be proved that if the prior is a Beta distribution, or its multinomial distribution counterpart Dirichlet distribution, then the posterior distributions also a Beta distribution or Dirichlet distribution.

In the single variable example of the thumbtack, if we use a Beta prior, the marginal likelihood is

\[
P(x = H|x_1, x_2, x_3 \ldots x_m) = \frac{N_{x=m} + \alpha_1}{m + \alpha}
\]

Where m is the sample size. For example, if \( D = (H, T, T, H, H) \),

\[
P(x_1, \ldots x_5) = \frac{\alpha_1 \alpha_0}{\alpha} \frac{\alpha_0 + 1}{\alpha + 1} \frac{\alpha_1 + 1}{\alpha + 2} \frac{\alpha_1 + 2}{\alpha + 3} \frac{\alpha + 4}{\alpha + 4}
\]

\[
= \frac{1}{2} \frac{2}{3} \frac{2}{4} \frac{3}{5} \frac{6}{6}
\]

If \( \alpha_1 = \alpha_0 = 1, \alpha = \alpha_1 + \alpha_0 = 2 \), we can obtain a result 0.017.

A single variable with multinomial distribution and a Dirichlet prior with hyperparameters \( \alpha_1, \ldots, \alpha_k \) is like:

\[
P(x_1, \ldots x_m) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + M)} \prod_{i=1}^{k} \frac{\Gamma(\alpha_i + N_{x=i})}{\Gamma(\alpha_i)}
\]

Heckerman et al. give out the General Dirichlet Prior Score Metric, which is also the marginal likelihood of Bayesian network is as:

\[
P(B_s, D) = P(B_s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk})}{\Gamma(N_{ij} + N_{ijk})} \frac{\Gamma(N_{ijk} + N_{ijk})}{\Gamma(N_{ijk})}
\]
Where $N_{ij}$ is the corresponding Dirichlet distribution hyperparameters. In K2, uniform distribution is used as prior, where $N_{ijk} = 1$. Heckerman et al. also does not use an arbitrary hyperparameters given by users. They are determined by the equivalent sample size $N'$ and the assumed local joint probability. Specifically,

$$N_{ijk} = N' p(x_i = k, \prod_i = j | B_{sc}^h, \zeta)$$

As an example, let $B_{x \rightarrow y}$ and $B_{y \rightarrow x}$ denote the belief network structures where x points to y and y points to x, respectively. Suppose that $N' = 12$ and that the user’s prior network gives the joint distribution $(x, y | B_{x \rightarrow y}^e, \zeta) = 1/4$, $p(x, \bar{y} | B_{x \rightarrow y}^e, \zeta) = 1/4$, $p(\bar{x}, y | B_{x \rightarrow y}^e, \zeta) = 1/6$, and $p(\bar{x}, \bar{y} | B_{x \rightarrow y}^e, \zeta) = 1/3$. Using the BDe metric, if we observe database $D$ containing a single case with both x and y true, we obtain.

$$p(D, B_{x \rightarrow y}^e | \zeta) = \frac{11!}{12!} \frac{6! 5! 3!}{6! 5! 2!}$$

There are also many other score metrics for Bayesian network, however, only Bayesian information criterion (BIC) score and Bayesian Dirichlet Equivalent (BDe) score have their counterparts in Dynamic Bayesian Network for now. Next we present the BIC and BDe score metrics for Dynamic Bayesian Network.

**Maximum Likelihood’s Information-Theoretic Interpretation**

Considering the model $G_0$ where X and Y are independent, we have

$$\text{score}_L(G_0: D) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}$$

We can consider the model $G_1$ where this is an arc $X \rightarrow Y$. The log-likelihood for this model is
\[
\text{score}_L(G_1 : D) = \sum_m \log \theta_{x[m]} + \log \theta_{y[m] | x[m]}
\]

Thus,
\[
\text{score}_L(G_1 : D) - \text{score}_L(G_0 : D) = \sum_m \log \theta_{y[m] | x[m]} - \log \theta_{y[m]}
\]

By counting how many times each conditional probability parameter appears in this term, we can write this sum as:
\[
\text{score}_L(G_1 : D) - \text{score}_L(G_0 : D) = \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - M[y] \log \hat{\theta}_y
\]

\[
\text{score}_L(G_1 : D) - \text{score}_L(G_0 : D) = M \sum_{x,y} P(x,y) \log \frac{\hat{P}(y|x)}{\hat{P}(y)} = M \cdot I_p(X;Y)
\]

We see that the likelihood measure the difference between independent variables, the higher mutual information implies higher likelihood, and verse versa.

**Bayesian Parameter Estimation in Dynamic Networks**

In sections 5.2.1-5.2.3, we introduce the fundamental theory that Bayesian model based, and how they are related to each other. N. Friedman, et al deduce Bayesian information criterion (BIC) and Bayesian Dirichlet Equivalent (BDe) for DBN[10], which is described in the following sections.

**Bayesian information criterion (BIC) for DBN**

Assuming the dataset D composed of \( N_{\text{seq}} \) complete observations. The \( l \)th such sequence has length \( N_l \) and specifies values for the variables \( x_{l[0]} \ldots x_{l[N_l]} \). With such a dataset, we can learn \( B_0 \) from \( N_{\text{seq}} \) transactions of initial slice, and learn \( B \rightarrow \) by \( N = \sum_i N_i \) for transition slices.
We denote,
\[
\theta_{i,j,k}^{(0)} = \Pr(X_i[0] = k_i \mid Pa(X_i[0]) = j_i)
\]
\[
\theta_{i,j,k}^{\rightarrow} = \Pr(X_i[t] = k_i \mid Pa(X_i[t]) = j_i)
\]
\[
N_{i,j,k}^{(0)} = \sum_t I(X_i[0] = k_i, Pa(X_i[0]) = j_i; X^t)
\]
\[
N_{i,j,k}^{\rightarrow} = \sum_t \sum_I(X_i[t] = k_i, Pa(X_i[t]) = j_i; X^t)
\]
where \(I(\cdot; x')\) is an indicator function which equals 1 if the corresponding event occurs in sequence \(x'\), and 0 otherwise.

The likelihood function decomposes as:
\[
\Pr(D \mid G, \theta_G) = \prod_i \prod_j \prod_{k_i} \theta_{i,j,k}^{(0)} \times \prod_i \prod_j \prod_{k_i} \theta_{i,j,k}^{\rightarrow} 
\]
And the log-likelihood is given by
\[
L(B : D) = \sum_i \sum_j \sum_{k_i} N_{i,j,k}^{(0)} \log \theta_{i,j,k}^{(0)} + \sum_i \sum_j \sum_{k_i} N_{i,j,k}^{\rightarrow} \log \theta_{i,j,k}^{\rightarrow} 
\]
Such decomposition implies that \(B_0\) is independent from \(B \rightarrow\), so we can give

BIC score as
\[
BIC(G : D) = BIC_0 + BIC_\rightarrow 
\]
where,
\[
BIC_0 = \sum_i \sum_j \sum_{k_i} N_{i,j,k}^{(0)} \log \theta_{i,j,k}^{(0)} - \frac{\log N_{\text{seg}}}{2} \#G_0 
\]
\[
BIC_\rightarrow = \sum_i \sum_j \sum_{k_i} N_{i,j,k}^{\rightarrow} \log \theta_{i,j,k}^{\rightarrow} - \frac{\log N}{2} \#G_\rightarrow 
\]
The score function demonstrates that the BIC score for initial network is independent from that of the transition networks.

**Bayesian Dirichlet Equivalent (BDe) for DBN**

Let us assume that for each structure \(G\), we have chosen the hyper parameters \(N_{i,j,k}^{(0)}\) and \(N_{i,j,k}^{\rightarrow}\). Then we can rewrite \(\Pr(D \mid G)\) as a product of two terms
\[
\prod_i \prod_{k_i} \frac{\Gamma(\sum_{k_i'} N_{i,j,k_i'}^{(0)})}{\Gamma(\sum_{k_i'} N_{i,j,k_i'}^{(0)} + N_{i,j,k_i}^{\rightarrow})} \times \prod_{k_i} \frac{\Gamma(\sum_{k_i'} N_{i,j,k_i'}^{\rightarrow} + N_{i,j,k_i}^{(0)})}{\Gamma(N_{i,j,k_i}^{(0)})} 
\]
This still requires us to supply the Dirichlet hyperparameters for each candidate DPN structure. Since the number of possible DPN structure is large, these prior estimates might be hard to assess in practice. We can separate the two networks and given two equivalent sample size \( \hat{N}^{(0)} \) and \( \hat{N}^{-} \). Given these components, we assign the Dirichlet weights as follows:

\[
N_{i,j,k'}^{(0)} = \hat{N}^{(0)} \times P_{B_0}(X_i[0] = k_i|Pa(X_i[0] = j_i))
\]
\[
N_{i,j,k'}^{-} = \hat{N}^{-} \times P_{B_0}(X_i[t] = k_i|Pa(X_i[t] = j_i))
\]

Structure Search

The difficulty of learning Bayesian networks lies in its large search space. The search space for a gene network of \( n \) genes is the space of directed acyclic graphs with \( n \) vertices. A recursive formula as well as an asymptotic expression for the number of directed acyclic graphs with \( n \) vertices \( (c_n) \) was derived by Robinson[15], which is

\[
C_n = \frac{n! \cdot 2^{n(n-1)}}{r \cdot z^n}; r \sim 0.57436; z \sim 1.4881
\]

For example, there are roughly \( 2.34 \cdot 10^{72} \) possible networks with 20 genes, and about \( 2.71 \cdot 10^{158} \) possible solutions for a gene network with 30 genes. Therefore, heuristic approaches such as simulated annealing or greedy algorithms are used to estimate Bayesian networks.

Heuristic Search

Greedy search is the most naïve heuristic structure search algorithm. The algorithm iterates each node, and makes local modifications in the structure, including edge addition, edge deletion, and edge reversal. The states adjacent to a current state are those where we change one edge, either by adding one, deleting one, or reversing the
orientation of one. We then apply the change that leads to the best improvement in the score. This process continues until no modification improves the score.

*Global Optimal Search*

Dynamic programming algorithms are proposed to find the global optimal Bayesian networks structure in $O(n2^n)$ time and memory [39, 40, 41, and 44]. The algorithms derive from the observation that optimal network structure is a DAG. The algorithm recursively searches optimal leaves of subnetworks to find the optimal network structure of variables $V$, where

$$\text{MDL}(V) = \min_{x \in V} \left\{ \text{MDL}(V\{x\}) + \text{BestMDL}(X, V\{x\}) \right\}$$

Where,

$$\text{BestMDL}(X, V\{x\}) = \min_{PAx \subseteq V\{x\}} \text{MDL}(X|PAx)$$

Leaves are recursively added to the subnetworks until all variables have been added and the optimal network has been found. For a problem with order graph has $2^n$ nodes representing all subsets of the variables. Each nodes of the graph represents the subproblem of finding an optimal network for the subset of variables corresponding to that node.
Figure 12. An order graph of four variables.

Figure 12 shows an example of generating successors of an order graph. The generated successors are stored in the next layer, while the remaining queues are used to generate successors for the other graph nodes in the current. After all successors have been generated from a layer, we write the nodes to disk according to the ordering.

Figure 13. Generating successors of a node in the order graph. The top half of the order graph nodes (in white) is the subnetwork; the bottom half is the score of that subnetwork. The top half of the parent graph nodes (shaded) is the variable and the candidate set of
parents; the bottom half is BestMDL for that variable and candidate parent set. (a) The starting queues. (b) The popped off nodes. (c) The generated successors which are stored in the next layer. (d) The new queues [43].

Dynamic programming finds the global optimal score for a Bayesian network. But it cannot scale up for large networks. Even if finding the BestMDL in each node takes O(1), it takes O(2^n) to build the order graph to find the optimal score. Malone, B[43] compares the running time for dynamic programming for a network of size no more than 30, and a number of records from 569-20000. It takes 78055s (21 hours) to obtain the networks for 30 genes!

Other Metrics

Time-delayed Dynamic Bayesian Network

There exist two major problems in the current DBN methods that greatly reduce their effectiveness. The first problem is the lack of a systematic way to determine a biologically relevant transcriptional time lag, which results in relatively low accuracy of predicting gene regulatory networks. The second problem is the excessive computational cost of these analyses, which limits the applicability of current DBN analyses to a large-scale microarray data. Therefore, Min Zou introduces a DBN-based analysis that can predict gene regulatory networks from time course expression data with significantly increased accuracy and reduced computational time. Figure 14 shows the process of approach.
Figure 14. Process of time lag DBN (1) Identification of the initial expression changes. (2) Potential regulators. (3) Estimation of the transcriptional time lag. (4) DBN: statistical analysis of the expression relationship between the potential regulator and its target gene in time slices. (5) Predicted gene regulatory network.

In Murphy’s BNT, all the genes in the dataset are considered as potential regulators of a given target gene, which makes it impossible to model large scale gene network because of exponentially increasing computational time. Most transcriptional regulators exhibit either an earlier or simultaneous change in the expression level when compared to their targets. This is able to limit the potential regulators of each target gene and thus significantly reduce the computational time. The other improvement by Zou is to perform an estimation of the transcriptional time lag between potential regulators and their target genes. The time difference between the initial expression change of a potential regulator and its target gene represents a biologically relevant time period. Figure 15 shows the initial expression change of a potential regulator. This is expected to
allow a more accurate estimation of the transcriptional time lag between potential regulators and their targets, because it takes into account variable expression relationships of different regulator–target pairs. These improvements are related to transcriptional time-delayed lags between regulators and target genes, so it can be considered as a time-delayed DBN.

Figure 15. The transcriptional time lag between the potential regulator and target gene.

AD Tree

Many machine-learning algorithms do frequent counting, so it is necessary to pay a one-time cost for building a caching data structure and doing counting in a constant time. ADtree[47] can be used to minimize memory use, and accelerate Bayesian learning with large dataset with large number of nodes. The cost of building a tree from R records and M attributes is bounded above by

$$\sum_{k=0}^{\log_2 R} \frac{R^M}{2^k \binom{M}{k}}$$
For an ADtree with 95,000 nodes, it required almost 11Mbytes of memory[49].

Compared the costs of performing 100,000 iteration of Bayes net structure search, which means the time to run 100,000 iterations is essentially the time to compute 100,000 contingency tables. For a network of 49,000 nodes and 16 records, ADtree takes 320s to do 100,000 interactions of hill-climbing search. Figure 16 shows an example of a sparse representation of ADtree. Figure 17 shows the algorithm to build ADtree.

\[ \text{Figure 16. The right hand figure is the sparse representation of the contingency table on the left[49].} \]

\[ \text{MakeADTree}(a, \text{RecordNums}) \]

\[ \text{Make a new ADnode called ADN.} \]

\[ \text{ADN.COUNT := |RecordNums|.} \]

\[ \text{For } j := i, i + 1, \ldots, M \]

\[ \text{jth Vary node of } ADN := \text{MakeVaryNode}(a_j, \text{RecordNums}). \]
**MakeVaryNode**(aᵢ, RecordNums)

Make a new Vary node called VN.

For k := 1, 2, ..., nᵢ

Let Childnumsk := {}.

For each j ∈ RecordNums

Let vᵢj = Value of attribute aᵢ in record j.

Add j to the set Childnumskᵢ.

Let VN.MCV := argmaxk | Childnumsk |

For k := 1, 2, ..., nᵢ

If | Childnumsk | = 0 or if k = MCV

Set the aᵢ = k subtree of VN to NULL.

Else

Set the aᵢ = k subtree of VN to **MakeADTree**(aᵢ₊₁, Childnumskᵢ)

---

**Figure 17.** The algorithm to build a sparse Adtree [49].

**Gaussian Bayesian Networks**

We define a Gaussian Bayesian network to be a Bayesian network all of whose variables are continuous, and where all of the CPDs are linear Gaussians.

Let Y be a linear Gaussian of its parents X₁,X₂,...Xₖ:

\[ p(Y|x) = \mathcal{N}(\beta_0 + \beta^T x; \sigma^2) \]

Assume that X₁,X₂,...Xₖ are jointly Gaussian with distribution \( \mathcal{N}(\mu; \Sigma) \). Then the distribution of Y is a normal distribution

\[ \mu_Y = \mu_0 + \beta^T \mu \]

\[ \sigma_Y^2 = \sigma^2 + \beta^T \Sigma \]

MLE principles can be applied in the setting of linear Gaussian Bayesian networks. Consider a variable X with parents U={U₁,...,Uₖ} with a linear Gaussian CPD.

\[ P(X|u) = \mathcal{N}(\beta_0 + \beta_1 u_1 + \cdots + \beta_k u_k; \sigma^2) \]

Our task is to learn the parameters \( \theta_{x|U} = <\beta_0, \ldots, \beta_k, \sigma> \). To find the MLE values of these parameters, we need to differentiate the likelihood and solve the equation...
that defines a stationary point. As usual, it will be easier to work with the log-likelihood function. Using the definition of the Gaussian distribution, we have that

\[ \ell_x(\theta_x|U:D) = \log L_x(\theta_x|U:D) \]

\[ = \sum_m \left[ -\frac{1}{2} \log(2\pi\sigma^2) \right. \]

\[ - \frac{1}{2} \frac{1}{\sigma^2} (\beta_0 + \beta_1 u_1[m] + \cdots + \beta_k u_k[m] - x[m])^2 \]

We start by considering the gradient of the log-likelihood with respect to \( \beta_0 \):

\[ \frac{\partial}{\partial \beta_0} \ell_x(\theta_x|U:D) = \sum_m -\frac{1}{\sigma^2} (\beta_0 + \beta_1 u_1[m] + \cdots + \beta_k u_k[m] - x[m]) \]

\[ = -\frac{1}{\sigma^2} \left( M\beta_0 + \beta_1 \sum_m u_1[m] + \cdots + \beta_k \sum_m u_k[m] - \sum_m x[m] \right) \]

Equating the gradient to 0, and multiplying both sides with \( \frac{\sigma^2}{M} \). We get the equation

\[ \frac{1}{M} \sum_m x[m] = \beta_0 + \beta_1 \sum_m u_1[m] + \cdots + \beta_k \sum_m u_k[m] \]

Each of the terms is the average value of one of the variables to the data.

\[ E_D[x] = \beta_0 + \beta_1 E_D[u_1] + \cdots + \beta_k E_D[u_k] \]

This also reflects the nature of the Gaussian Bayesian representation where the mean of a linear Gaussian variable \( X \) is in terms of the means of its parents \( U_1, \ldots, U_k \). It also tells us that the MLE parameters should be such that the mean of \( X \) in the data is consistent with the predicted mean of \( X \) according to the parameters.
Next, consider the gradient with respect to one of the parameters $\beta_0$. Using similar arithmetic manipulations, we see that the equation $0 = \frac{\partial}{\partial \beta_0} \ell_x(\theta_x|U; D)$ can be formulated as

$$E_D[X \cdot U_i] = \beta_0 E_D[U_i] + \beta_1 E_D[U_1 \cdot U_i] + \cdots + \beta_k E_D[U_k \cdot U_i]$$

At this stage, we have $k+1$ linear equations with $k+1$ unknowns, and we can use standard linear algebra techniques for solving for the value of $\beta_0, \beta_1, \ldots, \beta_k$. 
CHAPTER VI
INFERRING GRNS FROM REAL TIME SERIES MICROARRAY DATA

Inferring gene regulatory networks (GRNs) is a challenging inverse problem. Most existing approaches have understandably low accuracy because of the intrinsic complexity of a biology system and a limited amount of available data. Dynamic Bayesian Network (DBN) is one of the widely used approaches to identify the signals and interactions within gene regulatory pathways of cells. It is well suited for characterizing time series gene expression data. However, the impacts of network topology, properties of the time series gene expression data, and the number of time points on the inference accuracy of DBN are still unknown or have not been fully investigated. In this paper, the performance of DBN is evaluated using both in-silico yeast data and three growth phases of Yeast Saccharomyces cerevisiae cell cycle data with different time points in terms of precision and recall. The reconstructed GRNs were compared with the actual GRNs obtained from SGD. This work may provide insight and guideline for the development and improvement of GRN inference methods.

Saccharomyces Cell Cycle Data

The gene microarray data we used is from Spellman et al. [19]. The Spellman experiment was chosen because it provides a comprehensive series of gene expression datasets for yeast cell cycle. Four time series expression datasets were generated using four different cell synchronization methods: Cdc15, Cdc28, alpha-factor and elutriation with 24, 17, 18 and 14 time points respectively (Table 1). The alpha-factor dataset contained more time points than Cdc28 and Elutriation datasets with fewer missing values than Cdc15. Therefore, we choose to use time series expression data from
alpha-factor method to infer the yeast cell cycle gene regulatory network.

Table 1

*Gene expression data from four methods in yeast cell cycle*

<table>
<thead>
<tr>
<th>Method</th>
<th>Sample Frequency</th>
<th>Cell Cycle Length</th>
<th>Time points</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cdc15</td>
<td>Every 20 min for 1 hr, every 10 min for 3 hr, every 20 min for the final hr</td>
<td>112m</td>
<td>24</td>
<td>10m</td>
<td>290m</td>
</tr>
<tr>
<td>Cdc28</td>
<td>Every 10 min</td>
<td>85m</td>
<td>17</td>
<td>0m</td>
<td>160m</td>
</tr>
<tr>
<td>Alpha</td>
<td>Every 7 min</td>
<td>64m</td>
<td>18</td>
<td>0m</td>
<td>119m</td>
</tr>
<tr>
<td>Elutriation</td>
<td>Every 30 min</td>
<td>-</td>
<td>14</td>
<td>0m</td>
<td>390m</td>
</tr>
</tbody>
</table>

*Saccharomyces Genome Database Description*

DREAM (Dialogue for Reverse Engineering Assessments and Methods) projects [18] yeast network is built according to Teresa Reguly’s paper (2006). The full LC dataset in the paper is available at the BioGRID and SGD. The BioGRID (Biological General Repository for Interaction Datasets) is a curated biological database of protein-protein and genetic interactions for all major model organism species while attempting to remove redundancy to create a single mapping of interactions. The BioGRID is hosted in Toronto, Ontario, Canada and is partnered with the *Saccharomyces* Genome Database. Teresa Reguly described a comprehensive database of genetic and protein interactions, and associated experimental evidence, for the budding yeast, as manually curated from over 31,793 abstracts and online publications [61].

In general, interactions reported in the literature are reliable: many have been verified by multiple experimental methods or more than one research group. Interactions
reported in reviews or as unpublished data were not considered sufficiently validated. Protein-RNA and protein-DNA associations detected by genome-wide microarray methods were also not included. Of the initial set of 53,117 publications from PubMed about yeast, 21,324 were designated as ‘wrong organism’, usually because of a direct reference to a yeast homolog or to a yeast two-hybrid screen carried out with a non-yeast bait [61].

Physical interactions were divided into eight in vivo categories (affinity capture-mass spectrometry, affinity capture-western, affinity capture-RNA, co-fractionation, co-localization, co-purification, fluorescence resonance energy transfer(FRET), two-hybrid) and six in vitro categories(biochemical activity, co-crystal structure, far western, protein-peptide, protein-RNA, reconstituted complex). Genetic interactions were divided into eight categories(dosage growth defect, dosage lethality, dosage rescue, phenotypic enhancement, phenotypic suppression, synthetic growth defect, synthetic lethality, synthetic rescue).

Figure 18. The number of interactions annotated for each experimental method. In this panel and all subsequent figures, each dataset is color coded as follows: LC-PI, blue; HTP-PI, red; LC-GI, aquamarine; HTP-GI, pink [61].
As Figure 18 shows, LC-PI and LC-GI networks exhibited an unexpectedly high degree of overlap, at 12% of all protein interactions and 17% of all genetic interactions. It is interesting to note that half of their interactions that do overlap in the LC-PI and LC_GI datasets mapped back to the same publication as each other, suggesting that investigators may often test specific interactions in order to support initial observations.

Relationship in *Saccharomyces* Genome Database falls under two categories, which is Genetic regulation and Physical regulation [63].

*Genetic regulation*

- **Synthetic rescue**

  In this type of experiment, mutation/deletion of one gene rescues the lethality or growth defect of a strain mutated/deleted for another gene.

- **Synthetic lethality**

  In this type of experiment, mutations or deletions in separate genes, each of which alone causes a minimal phenotype, result in lethality when combined in the same cell under a given condition.

- **Positive Genetic**

  In this type of experiment, mutations/deletions in separate genes, each of which alone causes a minimal phenotype, result in a less severe fitness defect than expected under a given condition when combined in the same cell.

- **Phenotypic Suppression**

  In this type of experiment, a genetic interaction is inferred when mutation or over expression of one gene results in suppression of any phenotype (other than lethality/growth defect) associated with mutation or over expression of another gene.
• Synthetic growth defect

   In this type of experiment, a genetic interaction is inferred when mutations in separate genes, each of which alone causes a minimal phenotype, result in a significant growth defect under a given condition when combined in the same cell.

• Negative Genetic

   In this type of experiment, mutations/deletions in separate genes, each of which alone causes a minimal phenotype, result in a more severe fitness defect or lethality under a given condition when combined in the same cell.

• Phenotypic Enhancement

   In this type of experiment, a genetic interaction is inferred when mutation or overexpression of one gene results in enhancement of any phenotype (other than lethality/growth defect) associated with mutation or over expression of another gene.

• Dosage Rescue

   In this type of experiment, a genetic interaction is inferred when over expression or increased dosage of one gene rescues the lethality or growth defect of a strain that is mutated or deleted for another gene.

• Dosage Lethality

   In this type of experiment, over expression or increased dosage of one gene causes lethality in a strain that is mutated or deleted for another gene.

• Synthetic growth defect

   In this type of experiment, a genetic interaction is inferred when mutations in separate genes, each of which alone causes a minimal phenotype, result in a significant growth defect under a given condition when combined in the same cell.
**Physical regulation**

- **Affinity capture-MS**

  In this type of experiment, an interaction is inferred when a "bait" protein is affinity captured from cell extracts by either polyclonal antibody or epitome tag and the associated interaction partner is identified by mass spectrometric methods.

- **Affinity capture-RNA**

  In this type of experiment, an interaction is inferred when a "bait" protein is affinity captured from cell extracts by either polyclonal antibody or epitome tag and the associated interaction partner is identified by specific RNA binding.

- **Affinity capture-Western**

  In this type of experiment, an interaction is inferred when a "bait" protein is affinity captured from cell extracts by either polyclonal antibody or epitome tag and the associated interaction partner is identified by Western blotting with a specific polyclonal antibody or second epitome tag.

- **FRET**

  In this type of experiment, an interaction is inferred when close proximity of interaction partners is detected by fluorescence resonance energy transfer between pairs of fluorophore-labeled molecules, such as occurs between CFP and YFP fusion proteins.

- **Co-localization**

  In this type of experiment, an interaction is inferred from co-localization of two proteins in the cell, including co-dependent association of proteins with promoter DNA in chromatin immunoprecipitation experiments.
- Co-fractionation
  
  In this type of experiment, an interaction is inferred from the presence of two or more protein subunits in a partially purified protein preparation.

- Co-purification
  
  In this type of experiment, an interaction is inferred from the identification of two or more protein subunits in a purified protein complex, as obtained by classical biochemical fractionation or affinity purification and one or more additional fractionation steps.

- Biochemical Activity
  
  In this type of experiment, an interaction is inferred from the biochemical effect of one protein upon another, for example, GTP-GDP exchange activity or phosphorylation of a substrate by a kinase.

To better understand the meanings of edges from *Saccharomyces* Genome Database which is used for evaluating network, genes were also selected from the literature related to particular functions in budding yeast cell cycle. 24 genes were selected which exist in the Spellman microarray data set, including 14 genes that encode proteins for organelles, and 9 genes which are cell cycle regulators. The 24 genes are described as Table 2.
### Table 2

**Description of 24 genes selected for gene regulatory network reconstruction**

<table>
<thead>
<tr>
<th>Genes</th>
<th>Locations</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>TUB4</td>
<td>MT nucleation</td>
<td></td>
</tr>
<tr>
<td>SPC98</td>
<td>MT nucleation</td>
<td></td>
</tr>
<tr>
<td>SPC97</td>
<td>MT nucleation</td>
<td></td>
</tr>
<tr>
<td>CNM67</td>
<td>Spacer, anchors OP to CP</td>
<td></td>
</tr>
<tr>
<td>BBP1</td>
<td>SPB core, HB linker to membrane</td>
<td></td>
</tr>
<tr>
<td>BIM1</td>
<td>Spindle Pole Body</td>
<td></td>
</tr>
<tr>
<td>Kar9</td>
<td>Microtubules’ action related in spindle polarity process</td>
<td></td>
</tr>
<tr>
<td>BUD3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KIP2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KIP3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMC1</td>
<td>Sister-chromatid cohesion during G2 and M phase</td>
<td></td>
</tr>
<tr>
<td>SMC3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDS5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLN1</td>
<td>G1 cyclin</td>
<td></td>
</tr>
<tr>
<td>CLN2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLN3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLB5</td>
<td>G1/S cyclin</td>
<td></td>
</tr>
<tr>
<td>CLB6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLB4</td>
<td>S/G2 cyclin</td>
<td></td>
</tr>
<tr>
<td>CLB2</td>
<td>M cyclin</td>
<td></td>
</tr>
<tr>
<td>CDC45</td>
<td>Associate with origin recognition complex (ORC)</td>
<td></td>
</tr>
<tr>
<td>CDC6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TUB3, SPC98, SPC97, CNM67, BBP1 are selected from yeast spindle pole body (SPB). Eukaryotic microtubule organizing centers (MTOC) is required for chromosome segregation during mitosis and meiosis, cellular morphogenesis, cell motility, and intracellular trafficking. SPB is the best-characterized MTOCs. 17 components of the mitotic SPB have been identified. SPBs are embedded in the nuclear envelope throughout the yeast life cycle and grow in diameter from 80nm in G1 to 110nm in mitosis.

BIM1, Kar9, BUD3, Kip2, Kip3 are related to microtubule actions in spindle polarity process.
BIM1p can bind to microtubules and is required for the instability of microtubules before spindle assembly. Kar9p has been implicated in the orientation of functional microtubule attachments into the bud during vegetative growth. Bim1p and Kar9p, appears to provide a functional linkage between the actins and microtubule. Bud3p, a protein for axial budding of haploid cells, accumulates at the bud neck. Kip2p, and Kip3p, are dyeing and the kinesin-like protein and involved in regulating microtubule dynamics, mediating nuclear migration to the bud neck.

SMC1, SMC3, PDS5 and ECO1 are related to sister-chromatic cohesion during G2 and M phases.

The circumference of the cohesion ring largely consists of flexible coiled-coil of the SMC1 and SMC3 subunits, binding each other in head to head and tail-to-tail orientation. The establishment of the cohesion complex is mediated by ECO1 (in S phase) and Pds5.

*Figure 19. Physical relationship from SGD.*
Figure 20. Genetic relationship from SGD.

Physical and Genetic relationships which are identified by SGD is showed Figure 19 and Figure 20. The relationships also founding literatures include SPC98 and TUB4, SPC98 and SPC 97, Kar9 and Bim3, Kip2 and Kip3. SGD shows SPC98 and TUB4 are synthetic lethality, which means SPC98 and TUB4 has similar function, while either one works, the SPBs will be functional and strain will be alive; However if both of them are mutated, the cell will die. SGD shows that SPC98 and SPC97 are dosage rescue, which means SPC98 and SPC97 has similar function, while one over expressed will rescues the growth defect of a strain. SGD shows Kar9 and Bim3 are affinity capture-western, which means Kar9 and Bim3 is able to bind together. SGD shows Kip2 and Kip3 are synthetic rescue, which means the two have similar function. The locations of these proteins are shown in Figure 21.
In conclusion, Saccharomyces Genome Database (SGD, http://www.yeastgenome.org/) is a relatively reliable and abundant resource for the molecular biology and genetics interactions of the yeast Saccharomyces cerevisiae.

Reconstruct GRNs Using Time-Delayed Dynamic Bayesian Network Approach

The budding yeast Saccharomyces cerevisiae cell cycle was used as the benchmark data for performance evaluation of GRN inference due to its characteristic of single eukaryotic cell so that the difference between cells can be ignored. The Spellman experiment was chosen because it provides a comprehensive series of gene expression datasets for budding yeast cell cycle. Four time series dataset were included, three of them were done by synchronized budding yeast cell at different phases of cell cycle, G1, small G1 and mitosis. 35 genes were randomly selected from the gene set, including four transcription factors: CLN3, CLB6, CLB2 and SWI4. These genes were known to be expressed in G1, S, G2, M and M/G1 using a standard Pearson correlation function.

Figure 21. Location of protein components of the spindle pole body [62].
There are 18, 14 and 24 time points in phase 1, phase 2 and phase 3, respectively. The GRNs were inferred using DBN corresponding to different subsets of genes in three phases. The inferred GRNs were compared with the actual GRNs obtained from SGD.

The true biological networks of budding yeast were built from SGD. In SGD, interaction between pairs of genes is defined as physical interactions and genetic interactions. Both interactions are considered to generate true networks which were also used in the DREAM project. The network built from SGD is a directed graph with mutual interactions between nodes as shown in Figure 22.

*Figure 22. True *Saccharomyces cerevisiae* GRNs built from SGD (The *Saccharomyces* Genome Database). (Green: peak at M/G1; Red: peak at G1; Orange: peak at G2/M; Blue: peak at S/G2).* 

The inferred GRNs corresponding to Phase 1 and Phase 3 are given in Figure 23.
Figure 23. Reconstructed GRNs at different cycle phase of *Saccharomyces cerevisiae* (Green: peak at M/G1; Red: peak at G1; Orange: peak at G2/M; Blue: peak at S/G2).

The reconstructed GRNs were compared with the true network in Figure 22. The precision and recall values are given in Table 3.

Table 3

*Comparison of Inferred GRNs at different phases of yeast cell cycle with the true network*

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Phases</th>
<th>Times</th>
<th>Points</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>G1</td>
<td>18</td>
<td></td>
<td>25%</td>
<td>8%</td>
</tr>
<tr>
<td>Cdc15</td>
<td>M</td>
<td>24</td>
<td></td>
<td>14%</td>
<td>4%</td>
</tr>
<tr>
<td>Elu</td>
<td>Small G1</td>
<td>14</td>
<td></td>
<td>23%</td>
<td>6%</td>
</tr>
</tbody>
</table>

It is found that the inference accuracy is relatively low, compared to previous work. DBN is generally higher if a network is sparse. In addition, we have founded that the DBN is sensitive to the datasets and not robust in finding hub genes with a high connectivity. Thus, there is a need to improve the structure learning method of DBN, as describe in Chapter VII.
CHAPTER VII
IMPROVING STRUCTURE LEARNING FOR DBN

In this chapter, we investigated a more sophisticated score metrics for Dynamic Bayesian Network which was first proposed by Nir Friedman for stationary DBNs structures learning of both initial and transition networks but has not been used for reconstruction of Gene Regulatory Networks (GRN) yet. We implemented Friedman’s Bayesian Information Criterion (BIC) score function, modified K2 algorithm that was used to learn Bayesian Network (BN) structure, tested it on reconstruction of both synthetic networks and real yeast networks, and compared its performance with REVEAL in the absence or presence of preprocessed network generated by Zou and Conzen’s algorithm. In the tested cases, Friedman’s score function showed a higher precision and recall than the naive REVEAL algorithm.

Algorithm Description
The Friedman’s Bayesian Information Criterion (BIC) score metrics is based on maximum likelihood, which is elaborated in section 2.1.3, the Algorithm is described as Figure 24.
\textbf{procedure} K2 for DBNs

{Input: A set of $n$ nodes, and upper bound $u$ on the number of parents a node may have, and a database $D$ containing $N_{\text{seq}}$ observations and $N = \sum_{i} N_i$ transactions for $i$ transition slices.)}

{Output: For each node, a printout of the parents of the node}

\textbf{B}_0:

\begin{algorithmic}
\For{$i = 1$ to $n$}
\State $\pi_i = \{\}$
\State $\text{Score \_old} = \text{score}(i, \pi_i[0])$ \{ $\pi_i[0]$ is in time 0\}
\State $\text{OkToProceed} = \text{true}$
\While{$\text{OkToProceed}$ and $|\pi_i[0]| < u$}
\State let $z$ be the node that maximizes $\text{score}(i, \pi_i[0] \cup \{z\})$
\State $\text{Score \_new} = \text{score}(i, \pi_i[0] \cup \{z\})$
\If{$\text{Score \_new} > \text{Score \_old}$}
\State $\text{Score \_new} = \text{Score \_old}$
\State $\pi_i[0] = \pi_i[0] \cup \{z\}$
\EndIf
\EndWhile
\EndFor
\end{algorithmic}

\textbf{B}_t:

\begin{algorithmic}
\For{$i = 1$ to $n$}
\State $\pi_i[t] = \{\}$
\State $\text{OkToProceed} = \text{true}$
\While{$\text{OkToProceed}$ and $|\pi_i[t]| < u$}
\State $\text{Score \_old} = \text{score}(i, \pi_i[t])$ \{ $\pi_i[t]$ is in time $t$ or $t-1$\}
\For{$t = 1$ to $T$}
\State $\text{Score \_new}[t] = \text{score}(i, \pi_i[t] \cup \{z\})$
\EndFor
\If{$\text{Score \_new}[t] > \text{Score \_old} \text{ for all } t$}
\State $\text{Score \_new} = \text{Score \_old}$
\State $\pi_i[t] = \pi_i[t] \cup \{z\}$
\Else $\text{OkToProceed} = \text{false}$
\EndIf
\EndWhile
\EndFor
\end{algorithmic}

\begin{itemize}
\item If $X_i[t] \rightarrow X_j[t+1]$
\item $X_i \rightarrow X_j$
\end{itemize}

\textbf{end}

\textbf{end}(K2);

\textit{Figure 24.} Dynamic Bayesian Network structure learning algorithm using The Friedman’s Bayesian Information Criterion (BIC) score metrics.
Existing Approaches for Comparison

For convenience of performance analysis in the next section, we briefly describe Murphy and Zou’s previous work here and present results in the next section. The widely used DBN implementation developed by Murphy and Mian (called Murphy’s DBN hereafter) is based on REVEAL [12]. Given an unknown structure with full observations, the algorithm learns the parent set for each node independently. There are \( \sum_{k=0}^{n} \binom{n}{k} = 2^n \) such sets, which can be arranged in a lattice for the permutation of factors. The problem is to find the highest score in the lattice. The approach taken by REVEAL is started from the bottom of the lattice, and evaluate the score at all points in the successive level, until a point is found with a score of 1.0. Zou and Conzen [17] proposed a method to generate a preprocessed network for potential regulators by biological interpretation of time course microarray data. It is assumed that the gene with earlier initial up-regulation is the potential regulator of those with later initial up-regulation. This preprocessed network is used to narrow down the searching space for Murphy’s DBN algorithm because it requires excessive time to find a permutation for each node even when imposing a maximum number of parents for the nodes if the network dimension is large.

Synthetic Data Description

We used GeneNetWeaver (GNW), which used in international Dialogue for Reverse Engineering Assessments and Methods(DREAM) competition, to generate the synthetic dataset. GNW is available at http://gnw.sourceforge.net along with its Java source code, user manual and supporting data.

GNW has an intuitive graphical user interface that makes the generation and simulation of gene network models. Network topologies are generated by extracting modules from known \textit{in vivo} gene regulatory network structures such as those of \textit{E.coli}
and *S. cerevisiae*. These structures are then endowed with detailed dynamical models of gene regulations including both transcription and translation processes using a thermodynamic approach accounting for both independent and synergistic interactions. Expression data is generated either deterministically or stochastically to model molecular noise in the dynamics of the networks, and experimental noise is added using a model of noise observed in microarrays.

Both transcription and translation are modeled using a standard thermodynamic approach [68] allowing for both independent and synergistic regulatory interactions. For each gene of a network, the rate of change of mRNA concentration $F^{RNA}_i$ and the rate of change of protein concentration $F^{Prot}_i$ are described by

$$F^{RNA}_i(x, y) = \frac{dx_i}{dt} = m_i \cdot f_i(y) - \lambda^{RNA}_i \cdot x_i \quad (7.1)$$

$$F^{Prot}_i(x, y) = \frac{dy_i}{dt} = r_i \cdot x_i - \lambda^{Prot}_i \cdot y_i \quad (7.2)$$

where $m_i$ is the maximum transcription rate, $r_i$ is the translation rate, $\lambda^{RNA}_i$ and $\lambda^{Prot}_i$ are mRNA and protein degradation rates and $x$ and $y$ are vectors containing all mRNA and protein concentration levels. $f_i$ is the activation function of gene $i$, which computes the relative activation of the gene, which is between 0 (the gene is shut off) and 1 (the gene is maximally activated), given the protein or TF concentrations $y$. The integration of the system of equations defined by (1) and (2) results in noiseless mRNA and protein concentration levels, respectively $x_i(t)$ and $y_i(t)$ for gene $i$.

Hence, random fluctuations affect concentration levels of mRNA and protein, whose expression can be viewed as a stochastic process. Both $F^{RNA}_i$ and $F^{Prot}_i$ are of the form
\[
\frac{dX_t}{dt} = V(X_t) - D(X_t) \quad (7.3)
\]

where \( V(X_t) \) is the production and \( D(X_t) \) the degradation term. The corresponding chemical Langevin equations (CLE) we use to model molecular noise in transcription and translation processes is

\[
\frac{dX_t}{dt} = V(X_t) - D(X_t) + c(\sqrt{V(X_t)\eta_v} + \sqrt{D(X_t)\eta_d}) \quad (7.4)
\]

where \( \eta_v \) and \( \eta_d \) are independent Gaussian white-noise process. \( C \) is multiplicative constant to control the amplitude of the molecular noise. According to this model, a gene that is not activated (\( V(X_t) \) close to zero) has a very low level of noise and it cannot suddenly have a very high transcription rate due to noise. In contrast, a gene that is activated has a higher level of noise [64]. Figure 24 shows an example of the dataset generated by the model.
Figure 25. Generation and simulation of in silico gene network models using GNW. (A) Network structure containing 100 genes and extracted from a regulatory network in yeast. (B) Effects of both molecular and measurement noise on gene expression data. (Top) The integration of the ODE model defined in (1) and (2) leads to noiseless gene expression. (Middle) Molecular noise is introduced by replacing Equations (1) and (2) with stochastic differential equations (SDEs) defined in (4). (Bottom) Superposition of both molecular and experimental noise [64].

Results for Synthetic Data

The synthetic datasets and network were generated using GeneNetWeaver from DREAM (Dialogue for Reverse Engineering Assessments and Methods) projects [18]. We used sub-networks of different sizes (i.e., 10, 20, 50 and 100 genes) with randomly pick-out factors from high-dimensional yeast GRN with 4441 nodes and 12873 edges. A model consisting of ordinary and stochastic differential equations and Gaussian noise was used to generate synthetic gene expression data with a total of 21 time points and 10 replicates for each time slice.

An example of the 10-gene transition network reconstructed using Friedman’s algorithm is shown in Figure 26. This network was converted to a GRN (Figure 7.2) by
forming a relationship between two genes if the two are related in time point t and time point t+1 as the DBN theory suggests.

**Figure 26.** A transition network of 10 genes learned by Friedman score metrics. The left column shows the genes at time point t, and the right column the corresponding gene at the next time slice. (b) The gene regulatory network converted from (a).

The second example is the GRNs with 50 genes as shown in Figure 26 where the dashed lines indicate false positive edges, and solid lines true positive edges. The true network used to generate synthetic data in GeneNetWeave is given in Figure 26 (a). The preprocessed network includes a large number of false positive edges (dashed lines), resulting in a lower accuracy. The GRN reconstructed by Murphy and Zou, as given in Figure 26(c), is a sparse network that has a lower recall, compared to the true network.

The GRN reconstructed by the modified Friedman method (Method 3) without a preprocessed network is a dense network, as given in Figure 27(d). It is noted that the two regulators (YOR383C and YAL051W) were successfully reconstructed and they interacts with 24 and 6 target genes, respectively. The GRN reconstructed by the Friedman method has a much higher structure similarity to the true network than Murphy and Zou (Method
In Method 4, the preprocessed network was also used in the Friedman method to reduce the search space. The reconstructed GRN is also a sparse network with only one regulator gene identified, as demonstrated in Figure 27 (e). It is seen that Zou and Conzen's algorithm can generate a preprocessed network to narrow down the searching space, which is meaningful. While it rules out around 86% edges from the complete network, it is also a relatively loose rule to retain a large network for the next level-learning algorithm. However, when the network size becomes larger, the precision of the preprocessed network (4.0% in the 50-node case and 2.4% in the 100-node case) gradually drops to the random guess precision (4.4% and 1.8%, respectively), as shown in Figure 27.

A complete performance comparison of the four algorithms in terms of precision and recall is given in Figure 27, which shows that Friedman’s method gives higher precision and recall than the method of Murphy and Zou in all four networks. These results demonstrate that Friedman’s method has a great potential in improving the accuracy of GRNs reconstruction.
(a) True network

(b) Zou&Conzon's preprocessing network

Ce: 14 Precision: 3.97% Recall: 13.8%
(c) Kevin Murphy’s algorithm
with Zou&Conzen’s preprocessing network
Ce: 8 Precision: 11.9% Recall: 7.4%

(d) Nir Friedman’s algorithm
Ce: 38 Precision: 23.4% Recall: 36.2%
Figure 27. The 50-gene network reconstructed by different algorithms with dashed lines indicating false positive edges, and solid lines true positive edges. (a) The true network, (b) Zou and Conzen’s prior network algorithm, (c) Murphy’s algorithm, (d) Friedman’s algorithm, (e) Friedman’s algorithm combined with the prior network.

Figure 28. Comparison of performance between different structure learning algorithms using synthetic dataset.
Table 4

Comparison of performance between different structure learning algorithms using synthetic dataset (C_e: Correctly inferred edges; P: Precision; R: Recall)

<table>
<thead>
<tr>
<th>Size</th>
<th>Nir Friedman</th>
<th>Nir Friedman + Zou and Conzon</th>
<th>Kevin Murphy + Zou and Conzon</th>
<th>Zou and Conzon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C_e</td>
<td>P</td>
<td>R</td>
<td>C_e</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.50</td>
<td>0.29</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>0.15</td>
<td>0.17</td>
<td>3</td>
</tr>
<tr>
<td>50</td>
<td>38</td>
<td>0.23</td>
<td>0.36</td>
<td>6</td>
</tr>
<tr>
<td>100</td>
<td>38</td>
<td>0.10</td>
<td>0.22</td>
<td>25</td>
</tr>
</tbody>
</table>

Results for Real yeast benchmark dataset

We also investigated the performance of Friedman’s DBN algorithm in reconstruct of GRNs from real biological datasets. We tested it on the benchmark yeast time series dataset from Spellman’s experiment [19], and compared it with Murphy’s DBN algorithm with Zou’s preprocessed network [17], as well as a modified Probabilistic Boolean Network algorithm [4]. The dataset is from Spellman’s experiment [19], and the interactions are from the SGD database. The networks reconstructed by these three algorithms are showed as Figure 29 and precision and recall are given in Table 5. The results show that the Friedman’s DBN algorithm outperforms Murphy’s DBN algorithms in terms of accuracy and recall. Murphy’s DBN algorithm shows a sparse network structure, compared with the rest. It is also found that the reconstruction accuracy from real biological datasets (Yeast datasets) is higher than that from the synthetic data.
Figure 29. The real yeast network reconstructed by different algorithms (dashed lines indicating false positive edges, and solid lines true positive edges). (a) Murphy + Zou algorithm (b) Probabilistic Boolean Network (c) Friedman’s score metrics.

Table 5

Comparison of performance between different structure learning algorithms using yeast benchmark dataset \((C_e: \text{Correctly inferred edges}; P: \text{Precision}; R: \text{Recall})\)

<table>
<thead>
<tr>
<th></th>
<th>Nir Friedman</th>
<th>Kevin Murphy + Zou and Conzon</th>
<th>Probabilistic Boolean Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>C_e</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>0.76</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Conclusions

In this study, we implemented Friedman’s score metrics for DBNs by our algorithm, and applied the algorithm in reconstruction GRNs using both synthetic time series gene expression data and real yeast benchmark dataset. The algorithm is able to capture the correlation between consecutive time slices in both score function and learning procedure. The results show that Friedman’s score metrics gives a higher
precision and recall than the naive REVEAL algorithm application in the absence or presence of preprocessed network generated by Zou and Conzen’s algorithm. Friedman’s score metrics we implemented has great potential in improving the accuracy of structure prediction for GRN reconstruction with complete synthetic time series data.
CHAPTER VIII
RECONSTRUCTION OF GRNS FOR NEUROTOXICITY

Military-related activities produce many different chemicals, a portion of which inevitably contaminate soil. Neurotoxicity has been associated with the energetic, TNT, RDX, Carbaryl and their degradation products. Monitoring assessing and predicting the risks these chemicals pose when released require fundamental knowledge on how neurotoxicity occurs. A major barrier to development of predictive risk tools is the lack of an appropriate and detailed model of the molecular events leading to neurotoxicity when organisms are exposed to contaminated soils. We are interested to identify and discover how components involved in neurotransmission within the soil organism *Eisenia fetida* interact and are affected by neurotoxicants. Understanding this network of interactions is essential for development of predictive risk models in the future.

As a terrestrial toxicological model organism, *E. fetida* has a simple but efficient nervous system that is an excellent model to study the major neurotransmitters and synaptic pathways. Many neurotransmission pathways are conserved between *E. fetida* and vertebrates. In previous work, Dr. Gong’s research group discovered that at sub-lethal doses, TNT affected genes involved in neurological processes. At appropriate dosages RDX and Carbaryl both exhibit reversible neurotoxicity in *E. fetida*. However, it is unclear whether RDX and Carbaryl have affected other neurotransmission pathways and how genes involved in these pathways interact in a broader network context to compensate for or to cope with the perturbation caused by exposure to a neurotoxicity. Therefore, a system biology approach is used to discover effects of neurotoxicants on neurotransmitter pathways related gene expression in a gene regulatory network (GRN).
Multi-Class Time-Series Earthworm Microarray Dataset

DNA microarray, a maturing genomic technology, has been used extensively as a diagnostic tool to complement traditional approaches such as histopathological examination for various diseases (particularly cancers) because microscopic appearances sometimes can be deceiving. Microarrays have also successfully served as a research tool in discovering novel drug targets and disease- or toxicity-related biomarker genes for cancer classification. In ecological risk assessment, indigenous species such as fish and earthworms are often used as bioindicators for adverse effects caused by environmental contaminants. Previously, Dr. Gong’s research group developed an earthworm (Eisenia fetida) cDNA microarray to analyse toxicological mechanisms for a military-unique explosive compounds 1,3, 5-trinitro-1,3,5-triazacyclohexane (also known as Royal Demolition explosive or RDX) and an insecticide 1-naphthyl methylcarbamate (also known as Carbaryl). These two compounds exhibit distinctive toxicological properties that are accompanied by significantly different gene expression profiles in the earthworm E. fetida [89, 90, 91], which has motivated Dr. Gong’s research group to look further into toxicant- or toxicity-specific signature genes/biomarkers. The second motivation comes from the fact that many diagnostic assays exist for human diseases while very few are available for evaluating impacts on environmentally relevant organisms. Gross survival, growth and reproduction rates are often used as assessment endpoints without reflecting the diseased population of affected animals that is an important part of long-term impact assessment. The last motivation is that computational tools such as machine learning techniques have been widely used in cancer and toxicant
classification with microarray data but rarely applied in microarray data analysis of environmentally relevant organisms [92, 93, 94].

From a regulatory standpoint, there is an increasing and continuous demand for more rapid, more accurate and more predictive assays due to the already large, but still growing, number of man-made chemicals released into the environment [95]. Molecular endpoints such as gene expression that may reflect phenotypic disease symptoms manifested later at higher biological levels (e.g., cell, tissue, organ, or organism) are potentially biomarkers that meet such demands. As a high throughput tool, microarrays simultaneously measure thousands of biologically relevant endpoints (gene expression). However, to apply this tool to animals under field conditions, one critical hurdle to overcome is the separation of toxicity-induced signals from background noise associated with environmental variation and other confounding factors such as animal age, genetic make-up, physiological state and exposure length and route [92,93].

Dr. Gong and co-workers have developed a 43,807-oligonucleotide E. fetida array, and generated a large-scale microarray dataset from a laboratory study where earthworms (E. fetida) were exposed to various concentrations of Carbaryl or RDX for various lengths of time, mimicking field exposure scenarios. The objective of the current study was to identify the mechanism of chemical-induced reversible neurotoxicity through reconstruction of gene regulatory networks in earthworms received different treatments (control, Carbaryl and RDX) during exposure and recovery phases.
Experimental Design and Dataset Generation

A new earthworm array containing 43,803 non-redundant 60-mer probes was used to generate the dataset. The probes were selected from 63,641 previously validated oligonucleotide probes, each targeting a unique *Eisenia fetida* transcript, and 37,439 (59%) of probed targets had meaningful biological annotation [96]. A synchronized earthworm culture (starting from cocoons) was created and mature worms bearing clitellum and weighing 0.4~0.6g was chosen for this experiment. Each worm was transferred from artificial soil-based bedding (culture) and housed in an individual glass vial (115 ml in volume) [97]. These worms were exposed to carbaryl (20 ng/cm$^2$) or RDX (2 µg/cm$^2$) or acetone (solvent control) on moistened filter paper lined up inside the vial. These chemical concentrations were selected because they did not cause lethality. The entire experiment was divided into three phases (Figure 8.2): acclimation (4 days), exposure (6 days) and recovery (7 days). The acclimation phase was necessary for the worms to adapt from soil culture to filter paper, and four samplings were taken to establish the “background” baseline under the control condition. Worms were sampled at 13 and 14 time points for all three treatments (control, RDX and carbaryl) during the exposure phase and the recovery phase, respectively. Sampled worms were measured for conduction velocity of the media giant nerve fiber (MGF) before being sacrificed by snap freezing in liquid nitrogen. All yet-to-be-sampled worms were transferred to new vials at the beginning of the next phase. For instance, at the end of exposure phase, all remaining worms were transferred from exposure vials (containing spiked filter paper) to recovery vials (containing non-spiked clean filter paper). No mortality occurred throughout the
whole experiment. Sampled worms were fixed in RNAlater-ICE to preserve RNA integrity at -80°C.

Total RNA were extracted from at least 5 worms per time point per treatment, except for the 10th time point of RDX treatment in recovery stage (R10-RDX) where only 4 replicates remained after removing an array due to the poor RNA quality in the second replicate. RNA samples were hybridized to the custom-designed 44K-oligo array using Agilent’s one-color Low RNA Input Linear Amplification Kit. After hybridization and scanning, gene expression data were acquired using Agilent’s Feature Extraction Software (v.9.1.3). In this study, a total of 436 good quality arrays were generated, corresponding to 436 worm RNA samples (= 4~8 replicate worms × (1 control treatment × 31 time points + 2 chemical treatments × 27 time points) (see Figure 30). There were 161 untreated controls, 141 carbaryl-treated, and 134 RDX-treated. Three manufacturing batches of arrays were used, so the replicates within the same treatment condition and sampling time point were distributed into three batches in order to minimize batch effects. For example, five replicate worms exposed to RDX were sampled at E01, and 2, 2 and 1 replicate worm was hybridized to arrays of batch 1, 2 and 3, respectively. A multidimensional scaling was used to examine batch effects, and results show that samples are not grouped by batch, suggesting batch had no significant effect in this dataset. Figure 31 shows the sampling scheme and time points of sample collections.
Figure 30. Array distributions to three treatments and 31 time points.

Treatments: Control, RDX (2 µg/cm²).

Worm synchronization:
- Apr 28: started a new worm culture from ca. 600 cocoons
- May 14: surveyed worm weight (avg 0.03g, n=7)
- May 30: surveyed worm wt. (avg 0.17g, n=10)
- Jun 16: surveyed worm wt. and maturity (avg 0.28g, n=11, 40% bearing clitellum)
- Jul 3: surveyed worm wt. (avg 0.29g, n=24) and population size (ca. 3x360=1080)
- Jul 16/17: surveyed worm wt. (avg 0.38g, n=35) and population size (1210)

Worms needed: 3 treatments X 27 time points X 10 worms/sampling + 4 time points (A00~A03) X 10 w/s = 850

Figure 31. Overview of study design and sampling scheme of the time-series earthworm toxicity study using the 44K-probe earthworm microarray [97].
Data Pre-processing

The following data pre-treatment steps were applied prior to further statistical and computational analyses: (1) feature filtering: flag out spots with signal intensity outside the linear range as well as non-uniform spots; (2) conversion: convert signal intensity into relative RNA concentration based on the linear standard curve of spike-in RNAs; (3) normalization: normalize the relative RNA concentration to the median value on each array; and (4) gene filtering: filter out genes appearing in less than 50% of arrays (i.e., present on at least 219 arrays). There were more than 43,000 genes remaining after this procedure.

Feature Filtering by Univariate Statistical Analysis

The Class Comparison Between Groups of Arrays Tool in BRB-ArrayTools v.3.8 software package was used to identify significantly changed genes. The collated earthworm array dataset was imported without any further normalization or transformation. The tool runs a random variance version of the t-test or F-test separately for each gene. It performs random permutations of the class labels and computes the proportion of the random permutations that give as many genes significant at the level set by the user as are found in comparing the true class labels. The following two 2-class comparison analyses were conducted to infer genes differentially expressed in response to Carbaryl or RDX controls vs. Carbaryl, and controls vs. RDX treatments. The following settings were employed: a univariate test random variance model, multivariate permutation tests with 10,000 random permutations, a confidence level of false discovery rate assessment = 99%, and a maximum allowed number of false-positive genes = 10. A total of 7537 unique genes were obtained after combining all significantly changed gene
lists from Carbaryl and RDX treatments. RDX treatment significantly affected 6618 genes, whereas Carbaryl affected 2729 genes, with 1810 genes being affected by both treatments. The frequency of significant gene expression alteration across 31 time points was counted. Only 123 genes affected by Carbaryl were significantly altered at two time points, and alteration of the remaining 2606 genes was observed at one time point. Similarly, 1159 genes affected by RDX were altered significantly at more than one time point, with 12 genes at four time points, 118 genes at three time points, and 1029 genes at two time points. No gene was altered more than four times.

Identification of Significant Pathways

The significantly altered genes were annotated using BLAST2GO. Among them, 686 carbaryl-affected and 1591-RDX affected genes had biologically meaningful functional annotation (E <= 1e-3), with 201 genes affected by both treatments. Using RefNetBuilder, software developed in house, we mapped these annotated and significantly changed genes to KEGG pathways. Figure 32 shows the percentage of genes significantly influenced by RDX or Carbaryl treatment in the total number of genes for each mapped KEGG pathway.
Figure 32. The percentage of genes significantly influenced by RDX or Carbaryl treatment in the total number of genes for each mapped KEGG pathway.
Figure 32. (continued).
Figure 32. (continued).
Figure 32. (continued).
Figure 32. (continued).
Because RDX and Carbaryl are both postulated neurotoxicants [97], we chose to focus on KEGG pathways related to nervous system, including Glutamatergic synapse [PATH: ko04724], GABAergic synapse [PATH: ko04727], Cholinergic synapse [PATH: ko04725], Dopaminergic synapse [PATH: ko04728], Serotonergic synapse [PATH: ko04726], Long-term potentiation [PATH: ko04720], Long-term depression [PATH: ko04730], Retrograde endocannabinoid signaling [PATH: ko04723], Synaptic vesicle cycle [PATH: ko04721], and Neurotrophin signaling pathway [PATH: ko04722]. Table 6 shows the probes (probeid) that were found significantly altered and mapped to above KEGG pathways as well as their mapped ko gene (koid) and the gene annotation. The graphical locations of significant genes (red-highlighted) in these pathways are showed in Figure 32-38.
Table 6

*Significantly influenced probe by RDX treatment that can be mapped to nervous system-related KEGG pathways.*

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Pathway annotation</th>
<th>Ko Id</th>
<th>Annotation</th>
<th>Probe Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>ko04720</td>
<td>Long-term potentiation</td>
<td>K02677</td>
<td>classical protein kinase C</td>
<td>TA2-005067</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04345</td>
<td>protein kinase A</td>
<td>TA2-150155</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-167858</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-041944</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04353</td>
<td>Ras-related protein Rap-1A</td>
<td>TA2-113231</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-013188</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-033516</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04348</td>
<td>protein phosphatase 3, catalytic subunit</td>
<td>TA2-031346</td>
</tr>
<tr>
<td>ko04721</td>
<td>Synaptic vesicle cycle</td>
<td>K04646</td>
<td>clathrin heavy chain</td>
<td>TA2-083920</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K11827</td>
<td>AP-2 complex subunit sigma-1</td>
<td>TA2-005358</td>
</tr>
<tr>
<td>ko04722</td>
<td>Neurotrophin signaling pathway</td>
<td>K06630</td>
<td>14-3-3 protein epsilon</td>
<td>TA2-163392</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-011452</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-096436</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-163091</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-081497</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-154927</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-052092</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-185969</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-169797</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K12462</td>
<td>Rho GDP-dissociation inhibitor</td>
<td>TA2-071316</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04353</td>
<td>Ras-related protein Rap-1A</td>
<td>TA2-113231</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-013188</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-033516</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04513</td>
<td>Ras homolog gene family, member A</td>
<td>TA1-221872</td>
</tr>
<tr>
<td>ko04723</td>
<td>Retrograde endocannabinoid signaling</td>
<td>K04344</td>
<td>CACNA1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td>ko04724</td>
<td>Glutamatergic synapse</td>
<td>K02677</td>
<td>classical protein kinase C</td>
<td>TA2-005067</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04344</td>
<td>CACNA1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K01115</td>
<td>PLD</td>
<td>TA1-135047</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-052397</td>
</tr>
</tbody>
</table>
Table 6 (continued).

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Pathway annotation</th>
<th>Ko Id</th>
<th>Annotation</th>
<th>Probe Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>ko04725</td>
<td>Cholinergic synapse</td>
<td>K04344</td>
<td>CACNA1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04345</td>
<td>PKA</td>
<td>TA2-150155</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-167858</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-041944</td>
</tr>
<tr>
<td>ko04726</td>
<td>Serotonergic synapse</td>
<td>K04344</td>
<td>CACNA1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td>ko04727</td>
<td>GABAergic synapse</td>
<td>K04344</td>
<td>CACNA1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K08341</td>
<td>GABARAP, ATG8, LC3</td>
<td>TA1-007269</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA1-073910</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-018485</td>
</tr>
<tr>
<td>ko04728</td>
<td>Dopaminergic synapse</td>
<td>K02677</td>
<td>classical protein kinase C</td>
<td>TA2-005067</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K03456</td>
<td>protein phosphatase 2 (formerly 2A), regulatory subunit A</td>
<td>TA1-002476</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-194413</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04344</td>
<td>voltage-dependent calcium channel P/Q type alpha-1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04345</td>
<td>protein kinase A</td>
<td>TA2-150155</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-167858</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-041944</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04348</td>
<td>PPP3C, CNA</td>
<td>TA2-031346</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04382</td>
<td>protein phosphatase 2 (formerly 2A), catalytic subunit</td>
<td>TA1-044936</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-152462</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-164070</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-180947</td>
</tr>
<tr>
<td>ko04730</td>
<td>Long-term depression</td>
<td>K02677</td>
<td>classical protein kinase C</td>
<td>TA2-005067</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K03456</td>
<td>protein phosphatase 2 regulatory subunit A</td>
<td>TA1-002476</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K13240</td>
<td>nitric-oxide synthase, brain</td>
<td>TA1-111472</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04344</td>
<td>CACNA1A/voltage-dependent calcium channel P/Q type alpha-1A</td>
<td>TA2-088864</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04961</td>
<td>RYR1/ryanodine receptor</td>
<td>TA1-143048</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K04382</td>
<td>PPP2C/protein phosphatase 2 (formerly 2A), catalytic subunit</td>
<td>TA1-044936</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-152462</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA2-164070</td>
</tr>
</tbody>
</table>
Among the above 10 pathways, we focus on three of them, i.e., Dopaminergic synapse [ko:ko04728], GABAergic synapse [PATH:ko04727], and Synaptic vesicle cycle [PATH:ko04721]. Because RDX has been shown binding to the GABA$_\text{A}$ receptor convulsant site, and blocking GABA$_\text{A}$ receptor-mediated currents and causing seizures [101]; and there also has been evidence that RDX affects Dopaminergic synapse to cause poisoning [102]. Synaptic vesicle cycle [PATH:ko04721] shows very good consistency and is involved in multiple pathways within the nervous system.

**Analysis of synaptic vesicle cycle pathway**

The cell biological events that orchestrate the release of transmitter at the synapse can be described in terms of a cycle, where the components at the synapse are recycled via a repeating step of trafficking, exocytosis and endocytosis. At synaptic sites, vesicles are loaded with neurotransmitter. Next, loaded vesicles dock near release sites, when vesicles initially dock they are not fusion competent. Vesicles first need to be primed so that they are able to fuse rapidly in response to calcium influx. Primed vesicles fuse very quickly in response to calcium elevations in the cytoplasm. This fusion event is thought to be mediated directly by the SNAREs and driven by the energy provided from SNARE assembly. Synaptic vesicle proteins that have been incorporated into the plasma membrane after fusion are retrieved by endocytosis. After transmitter is released and binds to receptor on the postsynaptic membrane, it must be cleared to permit subsequent signaling. Some transmitters like dopamine are transported back into the neuron using plasma membrane transporters. Other transmitters including acetylcholine and some neuropeptides are broken down in the synaptic cleft.
In our experiment, AP-2 complex subunit alpha and Clathrin light chain A were consistently identified as significantly expressed under both RDX exposure and Carbaryl exposure as showed in Figure 32, where interestingly both of the genes were responsible for endocytosis in synaptic vesicle cycle.

Clathrin is a protein that plays a major role in the formation of coated vesicles. Clathrin was first isolated and named by Barbara Pearse in 1975. It forms a triskelion shape composed of three Clathrin heavy chains and three light chains. When the triskelia interact they form a polyhedral lattice that surrounds the vesicle.

Coat-proteins, like Clathrin, are used to build small vesicles in order to safely transport molecules within and between cells.

AP-2 complex subunit alpha-1 is a protein that in humans is encoded by the AP2A1 gene [103]. The complex is part of the protein coat on the cytoplasmic face of coated vesicles, which links Clathrin to receptors in vesicles.

GABAergic, Dopaminergic and Glutamatergic synapses all involve endocytosis activity. Endocytosis retrieves synaptic vesicle proteins that have been incorporated into the plasma membrane after fusion in synaptic vesicle cycle, which might capture the phenomenon where more vesicles are generated to load with neurotransmitter or vesicles are pathologically recollected after exposure to RDX and Carbaryl.

Analysis of GABAergic synapse pathway

Gamma aminobutyric acid (GABA) is the most abundant inhibitory neurotransmitter in the mammalian central nervous system (CNS). When released in the synaptic cleft, GABA binds to three major classes of receptors: GABA_A, GABA_B, and GABA_C receptors. GABA_A and GABA_C receptors are ionotropic and mediate fast GABA
responses by triggering chloride channel openings, while GABA$_B$ receptors are metabotropic and mediate slower GABA responses by activating G-proteins and influencing second messenger systems. GABA$_A$ receptors, the major sites for fast inhibitory neurotransmission in the CNS, are regulated by phosphorylation mechanisms, affecting both their functional properties and their cell surface mobility and trafficking. GABA release by the pre-synaptic terminal is negatively regulated by GABA$_B$ autoreceptors, and is cleared from the extracellular space by GABA transporters (GATs) located either on the pre-synaptic terminal or neighboring cells.

In our experiment, GABARAP (GABA(A) receptor-associated protein (autophagy-related protein 8)) was identified as a significantly changed gene under RDX and Carbaryl exposure. There is evidence in the literature that RDX binds to the GABA$_A$ receptor convulsant site, blocks GABA$_A$ receptor-mediated currents and causes RDX-induced seizures [101], which is consistent with our analysis.

Analysis of Dopaminergic synapse pathway

Dopaminergic synapse pathway [ko: ko04728] was chosen to reconstruct gene regulatory networks because it has a relatively ideal number of identified gene and an ideal amount of probes for each gene. By experience, networks with 5-15 genes have a relatively stable structure and reasonable density when reconstructed using the DBNs. The existence of multiple probes coding for the same gene provides extra information for the reconstruction algorithm and helps stabilizing reconstructed networks. The DBN model described in Chapter 7 was used to reconstruct GRNs using the expression data of the probes showed in Table 8.1 for RDX, Carbaryl, and control treatments during exposure and recovery stages. And then probes belonging to the same KO gene were
combined and mapped to the respective KEGG pathway in the background. The original reconstructed network and the KEGG Dopaminergic synapse pathway-mapped network are displayed side by side in Figure 39-44 as follows: 39 Control Exposure; 40 Control Recovery; 41 RDX Exposure; 42 RDX Recovery; 43 Carbaryl Exposure; 44 Carbaryl Recovery.

The significant genes altered by RDX exposure in this pathway were identified as voltage-dependent calcium channel P/Q type alpha-1A (CACNA1A), protein phosphatase 2c catalytic subunit (PP2A), protein phosphatase 3 catalytic subunit (PP2B), classical protein kinase C (PKC), and protein kinase A (PKA).

Protein phosphatase 2A is one of the four major Ser/Thr phosphatases, and it is implicated in the negative control of cell growth and division, which also involves Wnt signaling pathway and Long-term depression pathway. CACNA1A is Voltage-sensitive calcium channels (VSCC) mediate the entry of calcium ions into excitable cells and are also involved in a variety of calcium-dependent processes, including muscle contraction, hormone or neurotransmitter release, gene expression, cell motility, cell division and cell death, which are also involved in MAPK signaling pathway, GABAergic synapse, Long-term depression, Morphine addiction, Nicotine addiction. Protein kinase C also known as PKC is a family of protein kinase enzymes that are involved in controlling the function of other proteins through the phosphorylation of hydroxyl groups of serine and threonine amino acid residues on these proteins. PKC enzymes in turn are activated by signals such as increases in the concentration of diacylglycerol (DAG) or calcium ions (Ca2+). Hence PKC enzymes play important roles in several signal transduction cascades. Protein kinase A (PKA) refers to a family of enzymes whose activity is dependent on
cellular levels of cyclic AMP (cAMP). PKA is also known as cAMP-dependent protein kinase. Protein kinase A has several functions in the cell, including regulation of glycogen, sugar, and lipid metabolism.

The Protein kinase C is known as involved in controlling the function of other proteins through the phosphorylation of hydroxyl groups of serine and threonine amino acid residues on these proteins; and is also known to be activated by signals such as increases in the concentration of diacylglycerol (DAG) or calcium ions (Ca\(^{2+}\)). The inferred relationship between PP2A and PKA, PP2A and PP2B, and PP2A and PKC consistently exists under the normal control condition as well as exposure and recovery stages of RDX and Carbaryl treatments. The major difference is voltage-sensitive calcium channels (CACNA1A) ’s regulation of PKC through Ca\(^{2+}\)is broken down during the chemical exposure, and instead calcium channels influences PP2B and PKA directly by Ca\(^{2+}\), but not through PKC regulation. These observations of computational inferences form a new hypothesis that calls for further biological validation experiments. Further wet-lab experiments guided by the in silico GRN inference and bioinformatics data mining are well warranted and can potentially lead to novel and breakthrough discoveries for uncovering RDX-induced reversible neurotoxicity.
Figure 33. Genes altered by RDX treatment mapped to the long-term potentiation pathway [PATH: ko04720].
Figure 34. Genes altered by RDX treatment mapped to the Synaptic vesicle cycle pathway [PATH: ko04721].
Figure 35. Genes altered by RDX treatment mapped to the Neurotrophin Signaling Pathway [PATH: ko04722].
Figure 36. Genes altered by RDX treatment mapped to Glutamatergic Synapse Pathway [PATH: ko04724].
Figure 37. Genes altered by RDX treatment mapped to GABAergic synapse [PATH: ko04727].
Figure 38. Genes altered by RDX treatment mapped to Dopaminergic synapse [PATH: ko04728].
Figure 39. Genes altered by RDX treatment mapped to Long-term depression [PATH: ko04730].
Figure 40. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], Control Exposure.
Figure 41. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], Control Recovery.
Figure 42. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], RDX Exposure.
Figure 43. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], RDX Recovery.
Figure 44. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], Carbaryl Exposure.
Figure 45. Reconstructed GRNs mapped to Dopaminergic synapse [PATH: ko04728], Carbaryl Recovery.
CHAPTER IX

CONCLUSIONS AND FUTURE WORK

Summary and Conclusions

Inference of gene regulatory network from time series gene expression data is a very challenging task for computational biologists. Lots of mathematical algorithms and computational approaches have been proposed for modeling gene regulatory networks, such as Boolean network, differential equations and Bayesian network. There is no so-called golden method, which can generally give us the best performance for any kind of data sets. Some models and approaches can better describe the biological networks such as partial differential equations, but it is difficult to determine the parameters in the equations and the computational time is not acceptable. While other approaches can infer gene regulatory networks from a large-scale data set, i.e., information theory model, they can only infer undirected networks and inference accuracy is very low. In the field of gene regulatory networks, the research goal is to improve the inference accuracy and reduce computational overhead.

In this work, we focus on improving the accuracy and speed of reconstruction of GRNs with Dynamic Bayesian based method. First, we implemented an algorithm for DBN structure learning with Friedman’s score function to replace REVEAL, and tested it on reconstruction of both synthetic networks and real yeast networks and compared it with REVEAL in the absence or presence of preprocessed network generated by Zou and Conzen’s algorithm. The new score metric improves the accuracy and recall of the reconstructed network.
In the improved Dynamic Bayesian algorithm described in Chapter VII, we found that although all nodes are added to the network and kept if the node can increase the maximum likelihood of Dynamic Bayesian Network. However, only maximum 3 nodes can be added to each node of four size (10, 20, 50, 100) networks we tested, and after that no adding nodes can increase the score. And actually this phenomenon is in agreement with the theorems presented by Tian [99] and De Campos et al [98]:

In an optimal Bayesian network based on the MDL scoring function, each variable has at most $\log(2N^{2N})$ parents, where $N$ is the number of records.

This phenomenon provides a good reason to investigate the lattice method, which tested all the permutation and combination of parents with a maximum number of parents. However, the lattice method does not provide a higher accuracy at the end, although it does gain a higher BIC score. Thus, the maximum likelihood provides an overview of the network structure, where the parents influence their children, but higher maximum likelihood with current discretized method does not possibly contribute to the reconstruction gene regulatory networks. Also the feature of maximum likelihood mentioned above is probably a limitation of the algorithm, while a nature gene regulatory network has complexity of more than $\log(2N^{2N})$ parents for each gene.

We also tested the Gaussian Bayesian Network described in Section 5.3.3.3. It does not works better either, although it is likely to interpret the synthetic data generated by ODE(ordinary different equations) using GeneNetWeaver, where for each gene of a network, the rate of change of mRNA concentration $F_i^{RNA}$ and the rate of change of protein concentration $F_i^{Prot}$ are described by[64]
\[ F_i^{RNA}(x, y) = \frac{dx_i}{dt} = m_i \cdot f_i(y) - \lambda_i^{RNA} \cdot x_i \] (9.1)

\[ F_i^{Prot}(x, y) = \frac{dy_i}{dt} = r_i \cdot x_i - \lambda_i^{Prot} \cdot y_i \] (9.2)

Where \( m_i \) is the maximum transcription rate, \( r_i \) is the translation rate, \( \lambda_i^{RNA} \) and \( \lambda_i^{Prot} \) are mRNA and protein degradation rates and \( x \) and \( y \) are vectors containing all mRNA and protein concentration levels. \( f_i \) is the activation function of gene \( i \), which computes the relative activation of the gene, the value is between 0 (the gene is shut off) and 1 (the gene is maximally activated), given the protein or TF concentrations \( y \).

Equation (9.1) is used to generate the gene expression level. Based on the equation, the rate of gene expression is proportion to protein concentration rates \( x_i \) and activation function \( f_i \), which is decided by the translation factors of gene \( i \) (or its parents/regulators). Implementation of function \( f_i \) is not clear, but due to the standard thermodynamic model it using. \( f_i \) is a linear combination of parent’s equations which is proportion to the parents concentration. The equation is much like Gaussian Bayesian Network definition, except that equation (9.1) describes the rate of expression, while Gaussian Bayesian Network describes the amount of expression. Those differences might be the cause of low accuracy. Besides, another reason of the low precision of Guassian Bayesian Network might be that maximum likelihood is not a efficient way to solve linear system.

As in the last part of the work, pre-screen related probes by mapping the sequence to an existing identified model organism pathway proved to be a very promising method to generate relatively reasonable and stable GRNs network structure. From our experience, networks with 5-15 genes have relatively stable structures and reasonable
density by this method. The amount of probes under each gene seems providing extra information for the reconstruction algorithm and helping stabilizing a reconstructed network. From the experiments of network analysis using yeast data described in Chapter 6, the overlap of reconstructed network under different treatment is very low (no more than 10%) although using the same set of probes. The same situation appears when we reconstructed earthworm network even we already selected the probes potentially under the same pathway by the methods in Chapter 8. However, the situation largely improved when we collapse the probes belongs to the same gene. The resulting collapsed network maintains a very good consistency and relatively reasonable structure under similar treatments. The phenomenon might shows the degree of association is more of divergence between genes than within the same gene, which provides a very promising way to find out the gene regulatory network than traditional methods more accurately and scientifically.

Future Directions

There are several possible areas that we can extend the current work in the future. The primary task is to test whether the phenomenon that association and causality is more of divergence between genes than within the same gene is repeatable with benchmark microarray gene expression dataset and network structure that can be verified. Another direction that can be worked on is to use linear system for synthetic data by theory given in section 9.1, and it is a relatively better way than Dynamic Bayesian Networks with synthetic data. Furthermore, although Dynamic Bayesian Networks is proved to be an accurate and reasonable way for GRN reconstruction from time series expression data, it still gains very low accuracy and consistency when the network size becomes larger than
20 genes. The accuracy and consistency drops very quickly when the network become
large, because the association and causality is much harder to capture when the system
becomes complex with limited information, noise data, and feedback regulation.
REFERENCES


[21] Kaern, M.; Elston, T.C.; Blake, W.J.; Collins, J.J. Stochasticity in gene expression:


[59] Nagaraj, SH.; Gasser, RB.; Ranganathan, S. A hitchhiker's guide to expressed sequence tag (EST) analysis. *Briefings in Bioinformatics*. 2007, 1, 6–21.


[77] Falciani, F.; Diab, AM.; Sabine, V.; Williams, TD.; Ortega, F. Hepatic transcriptomic profiles of European flounder (Platichthys flesus) from field sites and computational approaches to predict site from stress gene responses following exposure to model toxicants. *Aquatic Toxicology*. 2008, 90, 92-101.


[82] Poynton, HC.; Zuzow, R.; Loguinov, AV.; Perkins, EJ.; Vulpe, CD. Gene expression profiling in Daphnia magna, part II: validation of a copper specific gene expression


