Building and validating gene regulatory networks (GRN) with delays from multiple data sources

Liu, Wenting

2015

Liu, W. (2015). Building and validating gene regulatory networks (GRN) with delays from multiple data sources. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65383>

<https://doi.org/10.32657/10356/65383>

**BUILDING AND VALIDATING GENE
REGULATORY NETWORKS (GRN) WITH DELAYS
FROM MULTIPLE DATA SOURCES**

**WENTING LIU
SCHOOL OF COMPUTER ENGINEERING
2015**

**BUILDING AND VALIDATING GENE
REGULATORY NETWORKS (GRN) WITH DELAYS
FROM MULTIPLE DATA SOURCES**

WENTING LIU

School of Computer Engineering

A Thesis Submitted to Nanyang Technological University
in Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

2015

Acknowledgments

I would like to express my sincere appreciation to my supervisors Prof. Jagath C. Rajapakse and Dr. Kuiyu Chang. Under their supervision, I have learned not only cutting-edge knowledge on data mining and bioinformatics, but also an inspiring and down-to-earth working attitude.

I am delighted to have the research support from Singapore's Ministry of Education Academic Research Tier 2 fund MOE2010-T2-1-056 and Tier 2 grant of ARC9/11. I would also like to thank Bioinformatics Research Center (BIRC) and BIRC group members that make the years I spent at BIRC a great experience.

I would also like to acknowledge my colleagues at Nanyang Technological University for their friendship and insightful discussions. They include Iti Chaturvedi, Piyushkumar Arjunlal Mundra, Jain Divya, Su Tran To Chinh, Liang Zhao, Qian Liu, Zhenhua Li, Xuchang Ouyang, Peng Yang, Shuoyu Xu, Song Liu, Fransiskus Xaverius Ivan, Guangxia Li, Zhen Hai, Min Wu, Peilin Zhao, Shumo Chu, and Linhong Zhu.

Finally, I am also grateful to my family for their unconditional love and constant moral support.

Contents

Acknowledgments	i
List of Figures	vi
List of Tables	viii
List of Abbreviations	x
List of Notation	xi
Abstract	xii
1 Introduction	1
1.1 Motivation	1
1.2 Research Objectives	4
1.3 Main Contributions of the Thesis	5
1.3.1 Rule-based Method for Deriving Gene Regulations	7
1.3.2 Fusing Evidences from Multiple Data Sources	8
1.3.3 Extracting Rate Changes in Transcriptional Regulation from MED- LINE Abstracts	9
1.3.4 Summary of Technical Contributions	9
1.4 Organization of the Thesis	10
2 Background	11
2.1 Building GRNs from Gene Ontology	11
2.1.1 Functional Modules from GO	12
2.1.2 Inferring GRNs from GO Transitive Relation	15
2.2 Fusion of Evidences from Multiple Data Sources	16
2.2.1 Biological Networks	16
2.2.2 Dynamic Network Modeling	17

2.2.3	Fusion Methods	20
2.3	Extracting Evidences for Time-delay in Gene Regulation from Literature	23
3	Inference of Gene Regulatory Networks based on Gene Ontology An-	
	notations	25
3.1	Background and Motivation	25
3.2	Improved Functional Similarity of Genes by Incorporating GO Enrichment	26
3.2.1	The Enrichment of GO Annotations	27
3.2.2	Functional Similarity of Genes	30
3.3	Gene Ontology Networks	34
3.4	Extracting Functionally Associated Networks	35
3.5	Gene Regulatory Networks from GON	36
3.6	Experimental Results	40
3.6.1	Benchmark Networks	40
3.6.2	Gene Ontology Annotations for Gene Sets	41
3.6.3	Gene Expression Time Series Data	41
3.6.4	GRN Inference from GE Time Series Data	41
3.6.5	Performance Measures	43
3.6.6	Extraction of Associated Networks	44
3.6.7	Building Regulatory Networks	51
3.7	Discussion	53
4	Fusion of Multiple Data Sources in Building Time-delayed Gene Reg-	
	ulatory Networks	55
4.1	Background and Motivation	55
4.2	Variable-order Dynamic Bayesian Network	58
4.2.1	Higher-order Dynamic Bayesian Network	58
4.2.2	Learning Higher-order DBN via <i>GlobalMIT</i> ⁺	59
4.2.3	Learning Variable-order Dynamic Bayesian Network	61
4.3	Fusion of Priors with Higher-order DBN	61
4.3.1	Structure Learning of r -th order DBN with Prior B	62
4.3.2	Learning Fusion Parameter β	65

4.4	Various Priors from Multiple Data Sources	65
4.4.1	Functional Association Network from STRING	65
4.4.2	Gene Association Network from Gene Ontology	65
4.4.3	Regulatory Paths from Gene Ontology Network	66
4.5	Experiments and Results	66
4.5.1	Synthetic Datasets	66
4.5.2	Real Datasets	67
4.5.3	Performance Evaluation	68
4.5.4	Performance without Priors	69
4.5.5	Performance with Various Priors	73
4.6	Discussion	87
5	Extracting Rate Changes in Transcriptional Regulation from MED-	
	LINE Abstracts	89
5.1	Background and Motivation	89
5.2	Text Preprocessing	91
5.2.1	Collecting Abstracts and Trigger Words	91
5.2.2	Word Normalization	92
5.2.3	Sentence Filter	92
5.3	Corpus Annotation	93
5.3.1	Records of Transcriptional Regulation Rate Changing Events . . .	93
5.3.2	Corpus Annotation	94
5.4	Biological Knowledge Driven Ontologies for Rate Changes in Transcrip- tional Regulation Events	95
5.4.1	Biological Knowledge Based Analysis	98
5.4.2	Negative Rules for Transcriptional Regulation	100
5.4.3	Statistical Analysis on Ontologies	101
5.5	Identifying Transcriptional Regulation Rate Changing Events with Rule- based Methods	103
5.5.1	Evaluating Ontologies	103
5.5.2	Deterministic Rule-based Methods	104

5.6	Identifying Direct Evidences for Events by Machine Learning Methods	105
5.6.1	Direct and Indirect Evidence Records for Events	105
5.6.2	Learning Direct Evidences by a Decision Tree	107
5.7	Discussion	110
6	Conclusions and Future Works	112
6.1	Summary & Conclusions	112
6.1.1	GRN from GON	112
6.1.2	Fusion of Multiple Biological Data Sources	113
6.1.3	Deriving Time-delayed Regulatory Events from Literature	114
6.2	Summary of Contributions	115
6.3	Future Works	116
6.3.1	A Novel Gene Interaction Scoring Scheme for Fusing Transitive PPIN and Gene Ontology	116
6.3.2	Fusion of Integrated Gene Interactions with Quantitative Time- Delayed GRN Inference Models	118
	Publications	119
	References	121

List of Figures

1.1	Overall research contributions in this research. Research 1: We first derived both direct and transitive gene regulations by using Gene Ontology (GO). Research 2: Then, a fusion framework was proposed to fuse various quantitative priors, such as protein-protein interactions from STRING, functional similarity from GO, or regulatory paths from GON, into various DBNs (DBN, HDBN, or VDBN) on gene expression time series to infer GRN with delays. Research 3: Lastly, we extracted delayed regulations from MEDLINE abstracts by using a text mining tool. They can be used to validate the inferred GRN with delays.	6
3.1	An example of GON. Circles represent GO terms, genes annotated by the term are shown beside, bold directed edges between terms are GO semantic relations.	35
3.2	The regulatory structures on functionally associated networks and pathways.	45
3.3	The ROC curves of association pathways by methods using various similarity scores on <i>E.coli</i> Nets and FANs.	48
3.4	The ROC curves of association pathways by methods using various similarity scores on yeast Nets and FAPs.	50
4.1	Overall Model: The dependencies among genes across different time points are modeled by VDBN; the quantitative gene interactions/regulations scores from STRING or GO are served as network priors in the fusion model; and the transcriptional factors (TFs) provide the parental constraints of genes since there are regulators.	58
4.2	The regulatory pathways of synthetic network.	67

4.3	The time-delayed regulatory pathways of synthetic network, where the numbers on the edges represent the delays.	67
4.4	The GRNs on <i>E.coli</i> network FAN2. The black lines represent the TP edges, the red lines represent FP edges, the dash black lines represent TN edges, and the numbers on the lines represent the delays.	85
4.5	The GRNs on yeast cell cycle network Net4. The same representation as Figure 4.4. Note that most of FP edges (red lines) are validated from YTRP database.	86
5.1	Transcriptional regulation-rate change ontology. The column of big boxes contain the textual patterns corresponding to the sub-processes of transcriptional regulation rate change events. The smaller boxes contain processes or classes of events. The numbers of instances (sentences) corresponding to the process in the corpus are indicated in the bracket next to each process.	96
5.2	Negative transcriptional regulation ontology. The rightmost columns of big boxes contain textual patterns corresponding to the sub-processes of negative instances. The other boxes contain processes or classes of events. The numbers of instances (sentences) corresponding to the process in the corpus are indicated in the bracket next to each process.	97
5.3	The average cumulative distribution of textual patterns from each ontology, over the number of sentences. The left figure shows transcriptional regulation rate change ontology distribution, and the right figure shows negative transcriptional regulation ontology distribution. The data is taken from the average over 100 random shuffles of the corpus sentences.	102

List of Tables

3.1	Functional evaluation of randomly selected networks and functionally associated networks/pathways	46
3.2	Performance of extraction of association pathways by methods using various similarity scores on <i>E.coli</i> Nets and FANs.	47
3.3	Performance of extraction of association pathways by methods using various similarity scores on yeast Nets and FAPs	49
3.4	Performance on extraction of regulatory pathways on <i>E.coli</i> Nets and FANs.	52
3.5	Performance of regulatory pathways methods on yeast Nets and FAPs. .	52
4.1	Performance of various GRN inference methods solely from GE data on synthetic networks. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network)	70
4.2	Performance of various GRN inference methods solely from GE data on <i>E.coli</i> randomly chosen networks and FANs. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network)	71
4.3	Performance of various GRN inference methods solely from GE data on yeast networks and FAPs. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network; and the parentheses of Pre. in FAP4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)	72
4.4	Performance of various DBNs fused with various priors on synthetic network.	74
4.5	Performance of various DBNs fused with various priors on time-delayed synthetic network.	75
4.6	Performance of various DBNs fused with various priors on <i>E.coli</i> Net1 and FAN1.	76

4.7	Performance of various DBNs fused with various priors on <i>E.coli</i> Net2 and FAN2.	77
4.8	Performance of various DBNs fused with various priors on <i>E.coli</i> Net3 and FAN3.	78
4.9	Performance of various DBNs fused with various priors on yeast Net4 and FAP4 with wild-type time series. (The parentheses of Pre. in Net4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)	79
4.10	Performance of various DBNs fused with various priors on yeast Net4 and FAP4 with mutant time series. (The parentheses of Pre. in FAP4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)	80
4.11	Performance of various DBNs fused with various priors on yeast Net5 and FAP5 with wild-type time series.	81
4.12	Performance of various DBNs fused with various priors on yeast Net5 and FAP5 with mutant time series.	82
5.1	Examples of transcriptional regulation-rate change events – one positive and one negative. Words in bold indicate transcriptional regulations while rate changes in the regulation process are italicized.	94
5.2	Transcriptional regulation-rate change pattern examples.	99
5.3	Negative patterns examples.	100
5.4	Rate change trigger words.	103
5.5	Performance of decision tree prediction using rule sets derived from different ontologies.	104
5.6	Classification performance of various ontology rule-based methods.	105
5.7	Examples of indirect evidence.	106
5.8	Performance of ten-fold cross-validation decision tree methods with various feature sets on identifying direct evidence instances.	109

List of Abbreviations

GRN	Gene Regulatory Network
GO	Gene Ontology
BP	Biological Process
MF	Molecular Function
CC	Cellular Component
GE	Gene Expression
PPIN	Protein-protein Interaction Network
FAN	Functionally Associated Network
FAP	Functionally Associated Pathway
STRING	Search Tool for the Retrieval of Interacting Genes/Proteins
YEAstract	Yeast Search for Transcriptional Regulators And Consensus Tracking
YTRP	Yeast Transcriptional Regulatory Pathway
KEGG	Kyoto Encyclopedia of Genes and Genomes
GEO	Gene Expression Omnibus
DBN	Dynamic Bayesian Network
HDBN	High-order Dynamic Bayesian Network
VDBN	Variable-order Dynamic Bayesian Network
LASSO	
Least Absolute	
Shrinkage and	
Selection Operator	
MCMC	Markov Chain Monte Carlo
TF	Transcription Factor
MIT	Mutual Information Test
MDL	Minimum Description Length
BDe	Bayesian-Dirichlet equivalent
BGe	Bayesian-Gaussian equivalent
ODE	Ordinary Differential Equation
DDE	Delay Differential Equation

List of Notations

$r_1 \wedge r_2$	Rule deduction from relation r_1 and r_2
$\binom{n}{m}$	The number of combinations of m objects from a set with n objects
$(\cdot)^T$	Matrix or vector transpose
Dir	Dirichlet distribution
$ $	The L_2 vector norm

Abstract

Cell functions are highly complex as they involve a concerted activity of many genes and their products (proteins). Each cell activity is typically coordinated by the organization of the genome into sets of genes that co-regulate in concert and share common functions (regulatory modules), common roles (processes), or interactions together in pathways. Gene Regulatory Networks (GRN) organize these related cell activities coherently into a concise network representation. Regulatory interactions are inherently asynchronous and transpire at different times. GRNs with time delays are therefore represent the dynamic regulatory interactions reasonably well, which can be inferred from gene expressions (GE) time-series. Although genome-wide expression profiles provide important information about the cellular processes, due to the paucity of information available in the individual data sources, GRNs are better learned by using multiple types of data sources combined with GE data. In this thesis, we proposed several novel methods to infer and validate GRNs including delays from multiple data sources such as gene ontology databases, literature, GE time-series data, and quantitative protein-protein interaction networks (PPIN).

Gene Ontology (GO) provides well-structured hierarchical vocabularies of biological processes and functional annotations of genes. It is commonly assumed that gene regulations/interactions occur among functionally associated genes. GO enrichment has been widely used to evaluate the functional coherence of a gene list, but has not been incorporated into evaluating similarity between genes. We incorporated GO enrichment to improve the functional similarities of genes and used them to extract functionally associated pathways. Most of previous research predicted gene interactions from GO functional similarities of genes by considering the hierarchical relations of GO terms, without/rarely considering their biological meaning, for example, the “regulate” relation. We proposed

a novel method to infer gene regulations by considering the biological clues of GO “regulate” relation. Transitive regulations were also detected by applying transitive rules of GO relations. To improve the accuracy and biological feasibility of inferred gene regulation, we restricted the transitive regulatory path method within a functionally associated pathway to detect regulatory pathways and construct GRNs. The benchmark networks included five functionally associated networks/pathways and five randomly chosen networks from *E.coli* network in RegulonDB and yeast cell-cycle networks in YEASTRACT database. Experimental results showed that our proposed functional similarity score outperformed both previous functional similarity score and the integrated functional association score from STRING on recovering almost all benchmark networks. It was also experimentally shown that our regulatory path method beat the state-of-the-art GNR inference methods from gene expression time-series data, dynamic Bayesian network (DBN), GeneReg and LASSO, which used graphical or linear regression models to predict various delays, on all the networks. Compared with the randomly selected networks, our regulatory path method performed much better on functionally associated pathways. This demonstrates that restricting the transitive regulatory pathway inference within functionally associated pathway is both useful for inferring GRN and biological plausible.

Time delays are important factors often neglected in gene regulatory network (GRN) inference models. As state-of-the-art method, DBNs have been widely used for modeling dynamics of gene regulation. Previous works have extended DBN to higher-order DBN (HDBN) so that the order can represent the delays. We adopted variable-order DBN (VDBN), which presumes each gene has its own delay. Thus biologically more plausible GRN can be modeled with time-delays by using VDBN. As the order of DBN is increased, the number of parameters in the model increases exponentially and the search space becomes exponentially large. We used *GlobalMIT*⁺, a polynomial algorithm to reduce the complexity of learning the VDBN structure. The optimal time delay for each gene was learned by the variable-order Monte Carlo Markov Chain (MCMC) scheme. We also explored the introduction of appropriate priors to improve the accuracy and reduce the complexity of learning DBNs. Since protein interactions and functional associations provide some evidence for gene regulation, we incorporated information of

these priors from STRING or GO into the process of GRN inference via VDBN from GE time series data. We used a Bayesian fusion framework to take the quantitative priors into account. The parameters for weighting the priors were automatically learned from their consistency with the inferred GRN from GE data. Experimental results showed that VDBN significantly outperformed DBN and HDBN with fixed orders. The prior information significantly improved the performance of all DBNs. Many predicted time-delayed regulations were validated by the literature. This demonstrates that our fusion method quantitatively captures the coherent genes’ dependency relations from multiple data sources, which is both effective and biologically reasonable.

Validating time delays from knowledge bases is a challenge since the vast majority of biological databases do not record temporal information of gene regulations. Biological knowledge and facts on gene regulations are typically extracted from bio-literature with specialized methods that depend on the regulation task. We mined evidence for time delays related to the transcriptional regulation of yeast from the PubMed abstracts. Since the vast majority of abstracts lack quantitative time information, we can only collect qualitative evidence of time delays. Specifically, the speed-up or delay in transcriptional regulation rate can provide evidence for time delays (shorter or longer) in GRN. Thus, we focused on deriving events related to rate changes in transcriptional regulation. A corpus of yeast regulation related abstracts was manually labeled with such events. In order to capture these events automatically, we created an ontology of sub-processes that are likely to result in transcription rate changes by combining textual patterns and biological knowledge. We also proposed effective feature extraction methods based on the created ontology to identify the direct evidence with specific details of the events. Our ontologies outperformed existing state-of-the-art gene regulation ontologies to identify rate-changing events. Experimental results showed that the machine learning method on our proposed features achieved an F1-score of 71.43% on identifying the direct evidence of these events. This demonstrates the effectiveness of our methods on deriving delayed gene regulations from bio-literature.

In summary, our research has resulted in three major contributions: extracting evidence for transitive regulation from structured biological database–GO; deriving delayed transcriptional regulation from unformatted data source–biological literature; and fusing

multiple data sources for a time-delayed GRN. We believe that these contributions will lead to more accurate and biological plausible models of dynamics in gene regulation in silico. The established GRNs provide deeper insight of biological processes and will be well applied in life science, such as disease analysis, clinical studies, etc.

Chapter 1

Introduction

1.1 Motivation

Complex biological cellular processes, including cell differentiation, metabolism, cell cycle, etc., involve a variety of gene expression and protein-protein interactions. Gene regulations refer to the cellular activity that arises when a set of genes interact with one another. Through regulation, genes orchestrate the level of synthesized mRNA and thereby control the expression of other genes and the rate at which proteins are produced, eventually deciding the state of the cell.

A Gene Regulatory Network (GRN) is a collection of genes (nodes) and their explicit interactions (links). To better understand the mechanisms underlying complex biological processes, one needs to construct accurate and biologically plausible GRNs. GRNs are typically built from gene expression (GE) data measured over thousands of genes using microarray technology. Delays in molecular interactions are inherent to GRNs since regulatory interactions are inherently asynchronous and transpire at different times. When gene expressions are collected over time, it is theoretically possible to build a GRN with delay information. Accurate prediction of regulatory networks via computational methods, which are quicker and cheaper than lab experiments, certainly expedites ongoing biotechnological projects.

Nowadays, various techniques such as mutual information [10], neural networks [59], Bayesian networks (BN) [103], and dynamic BN (DBN) [30], ordinary differential equations (ODE) [39], etc., have been developed to reconstruct GRN from GE data. A very important task in GRN modeling studies is to assure the reliability of network models

and validate the biological feasibility of the inferred GRNs. However, ground truths of biological regulatory networks are unknown in most cases, and existing databases of molecular interactions do not contain information about the delays on regulatory events, so building an accurate and biologically plausible GRN with delays remains an open research problem in the field of functional genomics. The inherent information of the input data or public biological knowledge can be used to validate GRN models. These sources of information such as text-mined literature data, protein interaction data, transitive relations, ontology rules, etc., provide some evidence for verifying the biological soundness of the inferred GRNs. In this research, we aim to generate time-delayed GRNs by using multiple data sources, which makes them biologically more plausible and accurate than those derived only from GE time-series data.

The structure of transcriptional networks based on databases of experimentally verified interactions provide direct evidence for GRNs. According to the study of the response times in chains of regulatory reactions, Rosenfeld et al. [121] showed that the response delays were due to the transitive transcription interactions. Thus, the transitive regulations can be used to validate time-delayed regulation. A transitive regulation is a regulation between two genes via one or more transitive genes in the absence of a direct regulation [159]. The vast majority of computational methods do not detect transitive regulations. An exception is that Zhou et al. [159] predicted them based on transitive expression similarity among genes of the same biological pathway. However, the transitive expression similarity of genes only indicates the functional interaction, which may not be the real regulation. Gene ontology (GO) provides a controlled vocabulary and hierarchy of terms to facilitate the annotation of gene functions and molecular attributes. By using the biological meaning of GO “regulate” relation and the transitive rules between GO relations, we can deduce transitive “regulate” relations between GO terms, from which we were motivated to derive transitive gene regulations.

Since the vast majority of existed biological databases are far from complete, and only bits and pieces of evidence could be gathered from literature, microarray experiments, online biological databases, such as GO¹, Kyoto Encyclopedia of Genes and Genomes

¹<http://www.geneontology.org/>

(KEGG)², and Search Tool for the Retrieval of Interacting Genes/Proteins (STRING)³, etc. These evidence can only individually and partially validate the GRN. Clearly, what is lacking is a mathematical framework that systematically fuses the various pieces of evidence into a GRN, with individual regulations quantified with a indicative confidence score. Integrating multiple data sources makes the validation process embedded into the integrating approach. Lubovac et al. [91] inferred a gene interaction network from integrating two sources of biological information: phylogenetic profiles and correlated expression. The inferred network was validated against the third source, a “trusted” network from the known interactions. If the agreement between the networks from different biological sources was significantly high, it was considered as a good inference method for gene interactions. Likewise, the functional association network (FAN) from STRING [97] quantitatively integrated physical interactions and functional associations derived from multiple sources: genomic context, high-throughput experiments, (conserved) co-expressions, and previous knowledge from scientific literature, e.g., PubMed papers. Motivated by this, we proposed a fusion method to combine the FAN from STRING with GE time-series to build and validate time-delayed GRN. To model GRN from GE time-series data, we adopted DBN, a widely used graphical model to capture temporal gene regulations from conditional dependency relationships between genes. It can be easily extended to higher-order DBNs (HDBN) to model time-delays. In our research, we used variable-order DBN (VDBN) to allow each gene to have its own delay, which was more reasonable for temporal genetic interactions. A Bayesian fusion framework was adopted to treat the quantitative evidence from STRING as the network structure prior in VDBN learning process, and the optimally learned graph was the one with maximum consistency between the graph learned from GE via VDBN with the prior.

Existing databases of molecular interactions do not contain information about the delays of gene regulations. This is mainly because either the delays cannot be experimentally measured or the text mining tools used to build these databases did not specifically capture the delay from the literature. Validating time delays from knowledge bases is

²<http://www.genome.jp/kegg/>

³<http://string-db.org/>

still a challenge. Biological knowledge and facts on gene regulation are widely existed in bio-literature, and they are typically extracted with specialized methods that depend on the regulation task. In our research, we aimed to mine evidence for time delays related to the transcriptional regulation of yeast from bio-literature, e.g., the MEDLINE⁴ abstracts.

1.2 Research Objectives

The objective of this research is to build and validate GRNs with delays by incorporating multiple data sources. Specific aims of this research are described as follows.

- Previous methods have used only one or a few data types. There is a need to extract evidence from multiple knowledge bases. We first make full use of existing knowledge to collect evidence for time-delayed GRN from GO databases, biological literature, or PPIN, etc. The collected evidence are assigned by quantitative reliability scores and to be constructed as GRNs. These GRNs are to be validated against some well-known GRNs to ensure that our techniques are indeed sound.
- Time delays are often neglected by previous methods. We aim to fuse multiple data sources into time-delayed GRN inference models from gene expression time-series data. Although each data source (PPIN, KEGG pathways, Biological literature) is incomplete, they collectively provide evidence or useful information on the regulatory mechanism. We fuse the various data sources together to obtain a more comprehensive understanding of the transcription regulatory mechanism. The inferred GRN achieve maximum consistency with prior information of existing knowledge from multiple data sources. This technique leads to more accurate and biological plausible GRNs modeling which makes full use of current knowledge.
- The validation of data-derived regulatory event delays is important as the ground truths of the delays are typically unknown. Incorporating biological rules to confirm the derived regulatory interactions is one possible way to validate GRNs. Moreover, the lack of current knowledge on regulatory mechanism means that we have a very few generic rules to depend on. We aim to derive a comprehensive set of rules

⁴<http://www.ncbi.nlm.nih.gov/pubmed>

to detect delayed regulation from knowledge bases. Further, we extend existing text mining tools to extract the delays of interactions, if available, from biological literature. Specifically, we propose to validate GRN by

- (i) performing gene ontology analysis,
- (ii) incorporating information from other molecular interaction data such as DNA-protein and protein-protein interactions,
- (iii) extracting information on delays, if any, from literature by extending existing text mining tools.

1.3 Main Contributions of the Thesis

The main techniques developed to achieve the aforementioned research objectives are illustrated in Figure 1.1. The overall framework introduced a technique aimed at fusing multiple data sources such as GE data, PPIN, GO, and literature for the purpose of building GRNs with delays.

Figure 1.1: Overall research contributions in this research. Research 1: We first derived both direct and transitive gene regulations by using Gene Ontology (GO). Research 2: Then, a fusion framework was proposed to fuse various quantitative priors, such as protein-protein interactions from STRING, functional similarity from GO, or regulatory paths from GON, into various DBNs (DBN, HDBN, or VDBN) on gene expression time series to infer GRN with delays. Research 3: Lastly, we extracted delayed regulations from MEDLINE abstracts by using a text mining tool. They can be used to validate the inferred GRN with delays.

The first research contribution is deriving both direct and transitive gene regulations by using GO. The direct gene regulations take place among biological similar genes where the biological similarity is derived from semantic similarity of GO annotations. By applying transitivity among genes' GO annotation relations, we induced "regulate" relations among genes to detect transitive regulations. The functionally associated pathways were

detected from GO to constrain the derived GRN to improve the performance.

Second, a fusion framework was proposed to fuse various DBNs including DBN, HDBN or VDBN, with various priors to infer GRN with delays. The various priors such as quantitative protein-protein interactions from STRING, the gene association networks from GO, or regulatory pathways from GO network, were used as structural priors to constrain and validate the predicted time-delayed GRNs from DBNs. The automatically learned GRN with delays achieved maximum consistency between the GE data and priors.

The third contribution leads to deriving delayed regulations from MEDLINE abstracts by using a text mining tool. To extract such events, we constructed two ontologies to summarize the biological reasons and textual patterns for delayed regulation. We also proposed some rules and feature extraction methods by using the ontologies to identify the events. The delayed regulations extracted from previous knowledge in scientific literature can be used to validate time-delayed GRN.

In the following subsections, we summarize the three main contributions. The details of contributions are presented in Chapter 3, 4 and 5, respectively.

1.3.1 Rule-based Method for Deriving Gene Regulations

Gene Ontology enrichment is widely used to evaluate the functional coherence of a gene list, but has not been incorporated in evaluating similarity between genes. We incorporated GO enrichment to improve the functional similarities of genes and used them to extract functionally associated pathways. The most of previous research have focused only on inferring functional coherence of genes by considering the hierarchical relation: i.e., *is – a*, *part – of*, of GO terms into the evaluation of similarity between gene annotation lists. We investigated the biological meaning of *regulate* relation by assuming that the “regulate” relation exists between genes with GO “regulate” relation in the same regulatory pathway. The transitive gene regulations can be detected by applying transitivity among genes’ GO annotation relations. We thus proposed a regulatory path method from GO network to detect both direct and transitive regulations.

The benchmark networks included five functionally associated networks/pathways and five randomly chosen networks from *E.coli* network in RegulonDB and yeast cell-

cycle networks in YEASTRACT database. Experimental results showed that our proposed functional similarity score outperformed both previous functional similarity score and the integrated functional association score from STRING on recovering almost all benchmark networks. Compared with the state-of-the-art GRN inference methods from time-series GE data, DBN, GeneReg [68] and LASSO [102], which used graphical or linear regression models to predict various delays, our regulatory path method performed significantly better on all networks. Compared with the randomly selected networks, our regulatory path method performed much better on functionally associated pathways. This demonstrates that restricting the transitive regulatory pathway inference within functionally associated pathway improves the accuracy for inferring GRN.

1.3.2 Fusing Evidences from Multiple Data Sources

As proteins are the products of genes, the physical interactions (i.e., protein-DNA, protein-protein interactions) provide direct evidence for gene regulatory networks. STIRNG provided functional association networks which quantitatively scored the potential of protein-protein interactions. The improved functional similarity with GO enrichment provided the functional association scores of genes. By using our proposed regulatory path method from GO network, the regulatory pathways were extracted with confidence scores. Our third work involved developing techniques to fuse theses various quantitative priors within various DBNs for building GRN with delays. The structure of DBNs were learned by a polynomial algorithm, *GlobalMIT*⁺ [144]. And the optimal delays in VDBN were learned by a variable-order Monte Carlo Markov Chain (MCMC) scheme [114]. A Bayesian fusion framework was adopted to take into account the priors to improve the accuracy and reduce the complexity for learning DBNs. The parameters for weighting the structural priors were automatically learned by maximizing the consistency of the GE data with priors. Experimental results on synthetic, *E.coli* and yeast cell-cycle networks showed that VDBN significantly outperforms DBN and HDBN with fixed orders. The prior information also improved the performance of all DBNs. This demonstrates the effectiveness and reasonableness of our fusion method on quantitatively capturing the coherent gene regulations from multiple data sources.

1.3.3 Extracting Rate Changes in Transcriptional Regulation from MEDLINE Abstracts

The delay information was rarely collected in existing formatted biological databases. Inspired by the biological facts/events widely existed in literature, we have derived delayed regulations from MEDLINE abstracts with a text mining tool. Since the vast majority of abstracts lack quantitative time information, we can only collect qualitative evidence of time delays. Specifically, the speed-up or delay in transcriptional regulation rate can provide evidence for time delays (shorter or longer) in GRN. Thus, we focused on deriving events related to rate changes in transcriptional regulation. A corpus of yeast regulation related abstracts was manually labeled with such events. In order to capture these events automatically, we developed an ontology of sub-processes that are likely to result in transcription rate changes by combining textual patterns and biological knowledge. We also proposed several effective feature extraction methods based on the created ontology to identify the direct evidence with specific details of these events. These techniques lead to a novel framework of text mining method for automatically detect complex biological events by incorporating relevant knowledge as supervision.

1.3.4 Summary of Technical Contributions

We summarize the technical contributions of our research as follows.

- (i) A method to incorporate GO enrichment to improve the functional similarities of genes; then construct functional associated pathways by clustering genes based on their functional similarities.
- (ii) A novel algorithm to detect “regulatory paths” from GO networks. Specifically, to find the shortest regulatory path among multiple edges by using transitive rules in GO term relations.
- (iii) The variable-order DBN model to represent and fuse multiple data sources to infer GRNs with delays.
- (iv) An effective fusion method to learn time-delayed GRN from VDBN with quantitative priors. The predicted time-delayed regulatory interactions can be validated

from GO information or literature. The fusion framework can be adapted to multiple data sources.

- (v) A corpus of yeast regulation related abstracts was manually labeled with events related to rate changes in transcriptional regulation. In order to capture these events automatically, we created an ontology of sub-processes that are likely to result in transcription rate changes by combining textual patterns and biological knowledge.
- (vi) Several effective feature extraction methods including semantic graph features based on the created ontology to identify the direct evidence with specific details of rate changes in transcriptional regulation events.

1.4 Organization of the Thesis

This rest of this thesis is organized as follows. Chapter 2 summarizes related background works. Chapter 3 describes our GO approach for GRN inference. Chapter 4 presents the fusion approach for time-delayed GRN inference. Chapter 5 describes the literature mining work for rate changes in transcriptional regulation events. Finally, Chapter 6 concludes the thesis and discusses future directions.

Chapter 2

Background

In this chapter, we first review methods on detecting evidence for inferring delayed GRN from public biological knowledge; then survey methods on fusing such evidence from multiple data sources; lastly, examine some text mining approaches for extracting biological events from bio-literature.

2.1 Building GRNs from Gene Ontology

The GO terms are used to describe the properties of genes and their products (i.e., proteins). They cover three domains: biological process (BP), molecular function (MF) and cellular component (CC). Specifically, biological process is a collection of molecular events, molecular function defines gene functions in the biological process, and cellular component describes gene locations within a cell. Providing functional categories of genes, these biological knowledge can help to identify functional similar genes, and guide the GRN inference from gene expression data, making the inferred GRN more biological plausible.

According to [5], GO terms have three kinds of relations: *is - a*, *part - of*, and *regulate* (<http://www.geneontology.org/GO.ontology.relations.shtml>). Previous works considered the hierarchical relations (*is - a* and *part - of*) of GO topology by GO enrichment to evaluate the representativeness of GO categories. We proposed to use *regulate* relation to infer gene regulations from their the corresponding GO terms. There are also some transitive rules that can help to deduce new “regulate” relations among GO terms. Thus, transitive gene regulations can also be deduced from transitive “regulate” relations between GO terms.

2.1.1 Functional Modules from GO

Gene Ontology provides annotations of gene functions and molecular attributes. It has been widely used in the validation procedure of functional genomics experiments to test the functional coherence of group genes, i.e., the representativeness of GO terms [130, 101]. Due to the hierarchical structure of GO, the “true path rule” exists in GO annotations: if a gene is annotated with a term, it is also annotated with the ancestors of that term [136]. Therefore, to compute the representativeness of a GO term/category for a given gene/gene set, one must take the GO enrichment of annotations into consideration by incorporating the possibility of extracting enriched categories. Hypergeometric distribution and GO enrichment have been applied to compute the statistical rank score of GO terms [25, 12, 20, 56, 47, 24].

When multiple tests were carried out, Bonferroni correction and false discovery rate(FDR) were two well-known techniques to control the false positives. The Bonferroni correction assumes that all hypotheses (GO terms) are independent, which ignored their dependent relations. It is usually considered a conservative adjustment by maintaining the probability of getting false positives at a particular alpha level. This can be overly restrictive, and result in lots of false negatives. The alternative approach is FDR, which allows a user to choose a threshold that has an acceptable level of false discovery. GO::TermFinder [20] compared their difference on an example data and showed that using the corrected P-value is conservative, it is likely to result in more false negatives; FDR generated more significant scores. According to [104], FDR will always yield as many significant scores as by using the Bonferroni correction.

To generate an organized classes of related genes, i.e., the biological modules/pathways, Huang et al. [67] used an agglomeration algorithm to condense a list of genes or associated biological terms. Manoli et al. [94] used over-representation analysis, functional class scoring (FCS), or global tests. These methods focused on measuring the probability of a single GO term or combination of GO terms to represent a gene set, but neglect the graph structure of GO. Based on the underlying GO graph topology, Alexa et al. [1] proposed two algorithms to improve GO group scoring: the *elim* method eliminated local dependencies between GO terms, so that the most specific nodes can be identified; the *weight* method compensated the *elim* method by comparing significance scores of

connected nodes (a parent and its child), so that the locally most significant terms in the GO graph could be detected. The p-values of the *elim* and *weight* methods were combined with the classic p-value derived from hypergeometric distribution and GO enrichment. At last, the *combined* score was applied to evaluate the significance of GO terms. Goeman et al. [52] proposed a multiple testing method based on a focus level of specificity of terms. This method preserved the graph structure of GO. But it was difficult to use in practice as one needs to predefine the focus level of specificity of terms. Richards et al. [119] assessed the global functional coherence of gene sets by modeling the enrichment of annotations and the relationships among annotations as a Steiner-tree problem. The algorithm they proposed was NP-hard in essence and thus computational inefficient. The *combined* method in “topGO” [1] improved this efficiency issue. It also took both GO enrichment and GO graph topology into account. We thus adopted it to evaluate the GO representativeness for a gene set.

The semantic similarity of GO annotations has been applied to compare two concepts (GO terms) in the context of a corpus within an ontology. Given a GOA file, due to the aforementioned “true path rule” in the hierarchical ontology, the number of genes annotated to ancestor GO terms exceeds that of descendant terms greatly. So the maximum number is achieved at the root. One can define the probability $p(t)$ of a GO term t as the number of genes annotated by it over the root term. It is obvious that p is monotonically non-decreasing as one moves up from descendant term to ancestor term. That is to say, if t “is-a” a , or t is “part-of” a , then $p(t) < p(a)$. The depth/specificity of two terms in a GO graph depends on their Lowest Common Ancestors (LCA). Resnik et al. [117] defined semantic similarity of two terms via the information content of LCA. Jiang and Conrath’s distance [73] and Lin’s similarity [86] all assessed the closeness between a term and its LCA. Schlicker et al. [125] combined Lin’s similarity and Resnik’s similarity, i.e., *simRel* to derive a new metric, which took both a term’s closeness to LCA and LCA’s specificity into measurement. Wu et al. [151] also defined relative specificity semantic relation by considering the specificity of LCA, the relative closeness of two terms, and their distance to LCA. We applied the *simRel* measure [125] to combine Resnik’s similarity and Lin’s similarity. Our approach was able to incorporate both the specificity of LCA and the relative distance of two terms with respect to their LCA. Chabalier et al. [29]

constructed a weighted term vector for each gene product, where the weight measures the representativeness of terms that annotate the gene product. The semantic similarity between gene products was calculated by assessing the similarity between corresponding term vectors. This method treated GO terms as independent. It ignored the hierarchical relations among GO terms. Pandey et al. [110] proposed a statistically motivated functional similarity measure which considers both the specificity and the distribution of GO terms across entity groups. Inspired by their work, we also incorporated the probability of GO terms being annotated to a given gene set into the definition of functional similarity.

Gene Ontology-based similarities have been widely used to infer genetic interactions, functional interactions, and protein-protein interactions [82, 6, 151, 149, 146]. Most GO-driven algorithms for the inference of gene/protein interactions are based on semantic similarity, machine learning techniques, or their combinations. Semantic similarity of gene annotations can provide some clues of direct gene regulations. Many works focused on the design of accurate measures to evaluate the functional similarity of genes based on the semantic similarity of their GO annotations. Lee et al. [82] accumulated the similarity of all combination of annotation term pairs as the similarity of two genes. A BN was employed to combine semantic similarity knowledge from the three GO hierarchies, biological process (BP), molecular function (MF) and cellular component (CC), to measure the strength of gene functional relationship [149]. The logistic regression model has been used to combine BP, MF and CC-derived measures [57], and it was found that CC-derived measures couldn't contribute to improve the performance of recovering gene interactions. We thus combined the semantic similarities from BP and MF hierarchies as functional similarity of genes for GRN inference.

There are also many works using machine learning techniques to learn the relationship between GO similarity and gene/protein interactions. Among the diverse features related to gene/protein interactions, biological processes and similarity in GO annotation between pairwise genes/proteins have been found to be one of the strongest indicators/dominant performance for discriminating true interactions from noise [57, 92]. Some statistical tests have been proposed to check whether the distribution of GO similarity for a pair of genes is different for the positive and random sets [149, 151]. Chabalier et

al. [29] generated the “transversal networks” from the pairs of gene products that have a high degree of similarity above some similarity threshold. Others used GO similarity to predict functional modules [149], or predict protein-protein interaction [146] via a threshold-based method. Similarly, we inferred direct regulations among gene pairs as those with functional similarity above some threshold.

Hong et al. [66] showed the success of gene-set approach for expression patterns detection of groups of genes instead of those of individual genes. Huttenhower et al. [37] presented a scalable Bayesian framework, Microarray Experiment Functional Integration Technology (MEFIT), for predicting functional relationships in the context of specific biological processes from integrated microarray datasets. In [57], the authors quantitatively assessed the application of GO-derived similarity measures for the characterization of both direct and transitive interactions within human regulatory pathways. It demonstrates that the integration of prior pathway knowledge helps for the GRN modeling. We thus extracted the pathway organization from GO annotations based on their functional coherence and functional similarity before inference of GRN.

2.1.2 Inferring GRNs from GO Transitive Relation

Existing works on semantic similarity extraction mostly focus on capturing the hierarchical relations, i.e., “is-a” and “part-of”, without considering the “regulate” relation [110, 92]. The “regulate” relation of the GO terms should not be ignored as they provide evidence for gene regulations. According to the definition of GO relations(<http://www.geneontology.org/GO.ontology.relations.shtml>), the “regulate” relation between two biological processes t and t' means that if these two processes occur, t always “regulate” t' . In a given GOA file, if these two processes t and t' over-represent two genes i and i' , respectively, then gene i most likely “regulate” gene i' because of the “regulate” relation between the processes they participate in. In our approach, we used GO term representativeness to filter more representative GO annotations for gene regulation inference from GO “regulate” relation.

There are also some transitive rules among the semantic relations. They can be used to deduce transitive “regulate” relation and infer transitive gene regulations. A transitive regulation is a regulation between two genes via one or more transitive genes

in the absence of a direct regulation [159]. To illustrate this, consider gene g_1 regulates gene g_2 directly, and gene g_2 relates to gene g_3 . As there is no direct regulation between g_1 and g_3 , if we can infer a regulatory relation from g_1 to g_3 , we say g_1 transitively regulates g_3 through transitive gene g_2 . The vast majority of previous research has focused on finding direct regulations between genes but rarely considering the transitive regulations. An exception is the work of Zhou et al. [159]. They linked genes of identical biological pathway based on transitive expression similarity among genes. The transitive co-expression genes were determined by applying the shortest path analysis on large-scale yeast microarray expression data. Instead of finding the shortest path based on distance, we detected the shortest path based not only on semantic distance/similarity of GO terms, but also on GO regulatory relations, which can give more reliable transitive regulations, as seen in our experiments. Transitive regulation should be more reliable/reasonable when inferred within biological pathway. We thus constrained our GRN inference within functional coherent sub-networks or pathways.

2.2 Fusion of Evidences from Multiple Data Sources

2.2.1 Biological Networks

Existing biological network databases provided many interaction information among biological entities, which can be used to validate GRNs. As proteins are products of genes, protein-protein interaction network (PPIN) databases including DIP [123], BIND [2], IntAct [4], MIPS [109], MINT [27], HPRD [112], Bio-GRID [129], Reactome [96], Pathway Commons [28] and protein complexes [128], have been widely used to validate GRNs [31]. It is worth noting that these databases were far from complete and under constant development. Many methods have been proposed to combine aforementioned databases to predict protein interactions. For example, MAGIC (Multisource Association of Genes by Integration of Clusters) [137] used a BN that combines evidence from different data sources such as the large-scale two-hybrid screens and multiple microarray data to quantify the functional relationship between any two proteins. Search Tool for the Retrieval of Interacting Genes/Proteins (STRING <http://string-db.org/>) database quantitatively integrated known and predicted protein interactions from four sources: genomic context,

high-throughput experiments, (conserved) co-expressions, previous knowledge from scientific literature, e.g., PubMed papers, and shared information between the organisms where applicable. It provided functional association networks (FAN) for a large number of organisms, which currently covers 5,214,234 proteins from 1,133 organisms. The quantitative weights of FAN edges indicated the confidence of functional interaction between each pair of connected genes/proteins [97].

Biological networks, such as PPIN, indirect (functional) associations, etc., record the interaction structure. But lacking information about the regulator and regulatee, such networks cannot be directly applied to validate GRNs. As transcription factors indicate regulators, they have been widely used to constrain the functional interactions into transcriptional regulations [161, 15, 72]. Kyoto Encyclopedia of Genes and Genomes Pathways (<http://www.genome.jp/kegg/pathway.html>) collected the molecular interaction and reaction networks generated by genome sequencing and other high-throughput experimental technologies. They have been directly used to validate gene regulations [39].

2.2.2 Dynamic Network Modeling

Biological processes are often time-dependent and governed by the dynamics of gene regulation [139, 14]. The most abundant source of information regarding such dynamic cellular activity is gene expression data at multiple time points [8]. By identifying the activated genes, inferring their rates of change and causal effects, the dynamic systems can be modeled from these time-series gene expression data. Some basic dynamical patterns that have been observed in time-series experiments were shown in [8].

Gene regulatory network is a logical way of describing phenomena observed with gene expression data. It has been shown that using GRN from experimental data to reason about their dynamics will increase our understanding of cellular function [42, 22]. In [83], it was demonstrated that the yeast cell-cycle network was extremely stable and robust for its function with the use of a simple dynamical model. Time delay in gene regulation is the time required to complete the sequence of reactions from change in regulator gene expression to yield observable target gene expression. As shown in [93], time delay was both theoretically and experimentally the source of instability and oscillations or

transient dynamics in several biological functions. Bratsun et al. [21] studied delay-induced stochastic oscillations in gene regulation, and they showed that time delay in gene expression caused a system to be oscillatory even when its deterministic counterpart exhibited no oscillations. They also demonstrated that considering such delay-induced instabilities help to reduce the effects of noise.

To estimate time-varying GRNs from time series microarray data, relevance measures such as time-lagged correlation was used in [75]; TimeDelay-ARACNE(Algorithm for the Reconstruction of Accurate Cellular Networks) [160] used time-delayed mutual information; regression based techniques, such as dynamic vector auto-regressive model [48], LASSO (least absolute shrinkage and selection operator) [50] were used to derive the dynamic structures. Mundra et al. [102] used time-lagged correlation first to restrict the possible delays, then adopted LASSO to solve the constraint regression models for each gene. We included this state-of-the-art fusion LASSO method in our experiments for comparison of our methods.

Ordinary differential equations (ODE) have been widely used for modeling the relations among differential expression of genes and the dynamic gene networks in systems biology [105, 76, 145, 68]. To construct time-delayed GRNs, Huang et al. [68] used delay differential equations (DDE), a time delay linear regression model, and provided an R package, GeneReg, to solve the DDE. Specifically, GeneReg used a linear function with two parameters: time delay was the time required to transmit change from regulator gene expression to a change of target gene expression; and the regulation coefficient represented the regulation effect. The optimal parameters were determined by minimizing a fitness function. Generally, it required a great deal of time for genetic network inference methods based on sets of differential equations, as the equations need to be solved many times. To reduce the computational cost, other methods have been proposed, for example, Kimura et al. [78] used discrimination classifiers to predict the signs of the gene expression levels, and GRN inference problem was solved by a linear programming machine.

Researchers have proposed tree-based machine learning methods such as Neural Networks (NN) [59, 100], GENIE3 [71], which was based on feature selection with tree-based ensemble methods, and TIGRESS (Trustful Inference of Gene REgulation using Stability

Selection) [64] to infer GRNs. Tree-based methods modeled each gene’s regulation relations with others as a tree. More generally, network based methods modeled all genes’ regulations in a network. A popular framework for inference of GRNs from gene expression data, BN [103, 152, 77, 55, 11] adopted graphical modeling paradigm to capture the statistical dependence among genes. It modeled the probabilistic relationships among a set of random variables (e.g., genes) in a directed acyclic graph with Markov assumption: each variable was independent of its non-descendants, given its parent in the graph. The joint probability density was expressed as a product of conditional probabilities by applying the chain rule of probabilities and independence. A limitation of BN models is that they cannot contain cycles, i.e., no feedback loops are allowed. The dynamic BN (DBN), which is an extension of BNs, overcomes this limitation. According to the comparison of various GRN inference methods in [58, 126], DBN, NN and state space model (SSM, i.e., DBN model with some hidden variables), significantly outperformed all the other models. We chose DBN-based methods in our GRN inference approaches.

Dynamic BNs were able to capture the dynamic temporal gene regulations by considering the temporal order of gene expression time-series [77, 79, 152, 55, 120, 11, 114, 54]. For example, the first-order DBN modeled the gene expression at current time point as dependent on parental genes expression at the previous one time point. As the time required for the regulator gene to express its protein product and the transcription of the target gene to be affected may be lagged, time delays need to be considered for accurate modeling of GRN. Higher-order DBNs (HDBN) were therefore needed to capture the time-delayed interactions [161]. Furthermore, as time delays may be different for different genes, Rajapakse and Chaturvedi [114] proposed variable-order DBN (VDBN) to model variable time-delays in GRNs, the optimal delay for each gene was learned by an order-varying Markov Chain Monte Carlo (MCMC) scheme. We adopted the VDBN to learn the time-delayed GRN from time-series GE data.

Because of the NP-hard nature of learning BN structure [35], most methods for learning DBN employed either local search such as hill climbing, or a meta stochastic global optimization such as genetic algorithm, simulated annealing and Monte Carlo simulation. Dojer et al. [43] showed that learning can be efficiently done in polynomial time for the globally optimal DBN, when Minimum Description Length (MDL) [80] and the Bayesian-Dirichlet equivalent (BDe) [40] scoring metrics were employed. Vinh et al. [143, 144]

showed that the mutual information test (MIT) [41] combined the goodness of both BDe and MDL, the network quality and speed. The polynomial time algorithm, *GlobalMIT*⁺, was then proposed to learn the globally optimal DBN structure with MIT.

2.2.3 Fusion Methods

As shown in [63], the gene expression data is generally incomplete and noisy for GRN modeling, augmenting it with prior information would help to learn more accurate GRNs. Existing biological network databases provide some evidence for GRN, for example, the interactions of gene products, protein-protein interactions were widely used to validate GRNs [138, 31]. They can be used as structural priors for GRNs [154, 103, 72]. Tu et al. [138] showed that accounting for just the binary protein-protein interactions is not enough, and that the varied interaction probabilities between different protein pairs should be quantified and reflected. Function associated network (FAN) from STRING provided quantitatively integrated scores for protein-protein interactions. The improved functional similarity with GO enrichment provided the functional association scores of genes. The regulatory pathways with confidence scores could be extracted by using our proposed regulatory path method from GO network. We can thus use these various quantitative priors in DBNs learning.

Systematic fusion of multiple data sources for GRN inference remains a key challenge in systems biology [32]. Hasan et al. [63] used a fitness function to incorporate prior information in differential evolution based inference method to capture mutual interaction information among genes. Pers et al. [111] used a multi-layered evidence network to combine heterogeneous data from various data sources such as genome-wide association, protein-protein interactions, disease similarity, linkage studies, and gene expression experiments for disease genes detection. In the works of reverse engineering of gene networks from gene expression data in DREAM3 in-silico network challenge(<http://wiki.c2b2.columbia.edu/dream/index.php/D3c4>), Yip et al. [155] observed different and complementary information about the underlying GRN existed in the given two types of data, they thus merged the predictions from the two types of data; Vignes et al. [142] proposed a simple yet very powerful meta-analysis that combines a

wide panel of methods ranging from BNs to penalized linear regressions, their meta-analysis was ranked first among the 16 teams participating in the Challenge 3A. Lo et al. [89] used linear regression model and formulated GRN construction as variable selection problems. To integrate various types of biological knowledge, they summarized them as an informative prior probability distribution over the candidate regression models.

When using the graphical modeling for inference of GRNs from GE data, a Bayesian fusion framework is contingent upon the assignment of a structural prior, a probability distribution on the network structure that encodes a *priori* biological knowledge either in the form of supplemental data or high-level topological features. Yeang et al. [154] developed physical network models to integrate protein-protein and protein-DNA interactions into gene knock-out expression profiles to infer transcriptional regulation. By annotating the available protein-protein and protein-DNA interactions in a skeleton graph, they modeled the consistency of these physical interactions with GE observations. The consistency of knock-out effect with physical interactions was also measured as potential function. Finally, the configurations of the skeleton graph that achieved the *maximum* consistency fielded the integrated physical network. Nariai et al. [103] incorporated protein-protein interaction knowledge when training a BN with gene expression data to estimate a GRN. In the BN structural learning process, when a proposed relation between genes is also listed among protein-protein interactions, they constructed a corresponding protein complex as a virtual node in the BN by using principal component analysis (PCA); the protein complex was used in the estimated network if it works better than using a single gene. Imoto's fusion method [72] also incorporated protein-protein interaction in their BN; the PPIN was updated whenever the learned BN structure changed. For each inconsistent edge in PPIN, they tested whether changing it (removal, addition, or reversal) would lead to better performance (in terms of increasing the likelihood). The change was allowed if a better likelihood was obtained, and rejected otherwise. As these fusion methods incorporated binary protein-protein interactions, they learned protein complexes or updated protein-protein interaction networks in the learning processes. They in fact treated the *priori* interactions qualitatively, accept or reject, during the learning processes, which obviously may lose some information.

To jointly learn the GRN by incorporating evidence from GE data with the prior from transcription factor (TF) binding location data, Bernard et al. [15] firstly learned

quantitative probability of TF-binding interactions. Werhli et al. [148] also integrated the prior knowledge and/or different experimental conditions quantitatively to learn GRNs. We adopted the Bayesian fusion framework to combine the quantitative prior information within VDBN learning. The parameters for weighting the priors were automatically learned from their consistency with the inferred GRN from GE data.

Since DBNs created from mRNA profiles represented the dependencies among genes, but may not correctly capture gene regulations, for example, it may not be able to model two kinds of dependencies [84]: one was the dependency between regulator genes and regulatee genes, the other was the inter-dependencies among co-regulated genes [62]. Using some priors to constrain the structures of BNs can help filter out these dependencies. For example, genome wide location analysis data were used to filter out co-regulated genes in [15, 148, 84], and transcription factors (TFs) were used to constrain potential regulators in DBNs to help to find regulator-regulatee dependencies in [161]. The constraint-based structure learning algorithms also help to reduce the search space of DBNs, as shown in [133], they generally performed well on sparse graphs like GRNs. Wu et al. [150] proposed an enhanced constraint-based approach for causal structural learning. They used an undirected graphical Gaussian model to extract the independence graph, where the edge set represented pairwise conditional independence relationships. They then learned the BN by using constraints from the independence graph. In our approach, we used TFs as the parental constraints of VDBN structure to not only reduce the search space, but also make sure the inferred dependencies among genes be regulatory relationships with known TFs.

Marbach et al. [95] investigated over 30 network inference methods on *Escherichia coli*, *Staphylococcus aureus*, *Saccharomyces cerevisiae* and in silico microarray data of Dialogue on Reverse Engineering Assessment and Methods (DREAM) project. They observed that no single inference method performed optimally across all data sets; in contrast, integration of predictions from multiple inference methods showed robust and high performance across diverse data sets. Thus, we also implemented this meta-fusion method by combining the inferred GRNs with various priors in our experiments to compare with our fusion method.

2.3 Extracting Evidences for Time-delay in Gene Regulation from Literature

Current formatted biological databases rarely recorded the time delays of gene regulation, thus they couldn't be used for validating the time delays in GRNs. There exists a large amount of untapped gene regulation information that have not been systematically mined and expressed in a known useable format from literature. Since volume of published biomedical research is expanding at an increasing rate, the underlying biomedical knowledge need to be extracted automatically. Text mining and knowledge extraction has been widely used to aid researchers to cope with this information overload and extract biological knowledge bases from literature. Text mining is well applied to named entity recognition (NER), text classification, terminology extraction, relationship extraction. For example, Medscan [106] can extract 90% of all facts described in scientific literature. Several research groups are constructing integrated flexible text-mining systems intended for multiple uses. In our research, we made extensive use of text mining to uncover the events about time-delayed gene regulations from literature.

A number of relation extraction methods have been applied to extract gene regulation events [124, 49, 51, 113, 13, 23, 60, 9]. The simplest approach is the co-occurrence based method [9] that predicts the relation between co-occurring entities at the sentence or abstract resolution. The rationale is that entities that are repeatedly mentioned together are somehow related. Prediction results of the co-occurrence methods typically have a high sensitivity but low specificity. Besides this limitation, details of the regulation such as direction and types cannot easily be determined using this method.

Pattern based extraction methods search for textual patterns of events with the help of predefined rules. Rules can be manually compiled [124] or learned from a training corpora [36, 51, 113]. Rules have been previously developed by biology experts [156] to build pathways from biological association networks (BANs) that have been derived from the facts in scientific literature by a natural language processing tool Medscan [106]. To extract and fire a rule, multiple ontologies have to be consulted in addition to bio-entity vocabularies [99, 36, 113, 13]. Concept ontologies, for example, are constructed from text corpora by using formal concept analysis [36]; the BioInfer project provides an ontology

for diverse biological events [51, 113]; Gene Regulation Ontology [13] is designed to model complex events that are part of the gene regulatory processes. Compared with the co-occurrence based methods, pattern based ones increase the specificity at the expense of a decreasing sensitivity.

Semantics analysis using natural language processing tools can help to extract complex rules from underlying sentences. For example, the RelEx Relation extraction system [49] uses a simple set of rules to extract relation paths from dependency parse trees. A set of hand-compiled rules were proposed for linguistic analysis and conceptual inference over semantic structures [60]. The rule-based system then matched syntactic-semantic patterns to the dependency structures. However, extracting rules manually from semantic structures is time-consuming and cannot be done easily on large-scale corpora. A current trend in feature design is to learn the semantic rules automatically via machine learning methods. For example, previous work investigated the performance of different Automatic Content Extraction (ACE) feature sets for biomedical relation extraction in a supervised learning setting [23]. Positive and negative regulation relations in biomedical pathways were learned from the lexical semantic annotations in [157]. Rich graph-based feature sets were proposed to extract complex biological events [16, 17].

In our approach, we created a corpus of yeast regulation related abstracts from MEDLINE, the events related to delays in transcriptional regulation was manually labeled. By analysing textual patterns and biological knowledge of these events, we developed ontologies for automatically extracting such events. We also proposed several effective feature extraction methods based on the created ontology to identify the direct evidence with specific details of these events.

Chapter 3

Inference of Gene Regulatory Networks based on Gene Ontology Annotations

3.1 Background and Motivation

Gene Ontology (GO) has been widely used to test the functional coherence of group genes [130, 101]. Many previous works evaluated the functional coherence by measuring the over-representation of particular functions or categories (GO terms) of a gene set [25, 12, 20, 56, 47, 24]. To compute the representativeness of a GO term/category, GO enrichment was usually considered by adding the probability of enriched terms based on the GO hierarchy. Alexaet al. [1] also considered the underlying GO graph topology to improve GO group scoring, which we adopted in our approach.

Gene Ontology-based similarities have been widely used to infer genetic interactions, functional interactions, and protein-protein interactions [82, 6, 151, 149, 146]. The interactions were predicted from the pairs of gene products that have a high degree of similarity above some similarity threshold [29, 146]. In this chapter, we also used functional similarity to infer functional associations and regulations among gene pairs via a threshold-based method. Moreover, since most of previous GO-derived similarity measures rarely considered GO enrichment, we modified the functional similarity measure by incorporating the functional coherence with GO enrichment.

Most of earlier works on using GO-derived similarity to predict gene interactions, mainly considered the GO hierarchical relations, “is-a” and “part-of”, without consider-

ing the “regulate” relation [110, 92]. The biological meaning of “regulate” relation GO terms can provide some clues of gene regulations. Applying some transitive rules among the semantic relations will deduce transitive “regulate” relation and infer transitive gene regulations. In this chapter, we proposed an algorithm to detect regulatory path in GO network (GON) to infer transitive regulations.

The transitive regulations were rarely considered in most previous research. Only Zhou et al. [159] linked genes of same biological pathway based on transitive expression similarity among genes. They determined the transitive co-expression genes by applying the shortest path analysis on large-scale yeast microarray expression data [159]. However, the transitive co-expression inferred from the shortest path based on gene expression distance indicates only the expression similarity but maybe not the real regulations of genes. Instead, we detected the shortest path based not only on semantic distance/similarity of GO terms, but also on GO regulatory relations, which gave more reliable transitive regulations, as seen in our experiments.

Inspired by the success of gene-set approach for the detection of expression patterns of group genes instead of those of individual genes in [66], we constrained the GRN inference from GO “regulate” relation within functionally associated pathways, to make sure that inferences are biological acceptable.

3.2 Improved Functional Similarity of Genes by Incorporating GO Enrichment

Gene ontology (GO) is a structured controlled biological vocabulary of various terminologies related to genes and their inter-related functional characteristics. It describes how gene products behave in a cellular context in a species-independent manner. The ontology covers three domains: biological process (BP), molecular function (MF), and cellular component (CC). BP is a collection of molecular events, MF defines gene functions in the biological process, and CC describes gene locations within a cell. A gene is associated with GO terms that describe the properties of its products (i.e., proteins). According to the study of combination of BP, MF and CC-derived measures, Guo et al. [57] found that CC-derived measures couldn’t contribute to improve the performance

of recovering gene interactions. As CC terms are not directly related to gene regulation, we only considered BP and MF terms to infer gene associations and regulations in our approach.

There are three defined semantic relations between these GO terms: *is-a* is used when one GO term is a subtype of another GO term, *part-of* is used to represent part-whole relationship in the GO terms, and *regulate* is used when the occurrence of one biological process directly affects the manifestation of another process or quality [5]. Let $\mathcal{R} = \{is-a, part-of, regulate\}$ denote the set of ontological relations. The GO terms and their relations can be constructed as a hierarchical directed acyclic graph (DAG), where the three domains, BP, MF and CC, are represented as three roots at the topmost level. Nodes/terms near the root of a DAG have broader functions and are hence shared by many genes; leaf nodes/terms on the other hand convey more specific biological functions.

GO annotation (GOA) is the process in which gene or gene products are annotated using GO terms. GOA data can be readily obtained from the GO annotation database (<http://www.geneontology.org/GO.downloads.annotations.shtml>). The GO hierarchical structure allows annotators to assign properties of genes or gene products at different levels, depending on the availability of information about an entity. In general, when inferring information from a gene that is annotated by some hierarchical GO terms, more specific biological functions at the lower levels should be chosen as an inference base due to their richer information content. As such, we need to come up with a measure to filter the more informative GO terms.

3.2.1 The Enrichment of GO Annotations

Given a gene set, GO enrichment analysis is usually performed to find which GO terms are over-represented for that gene set. The representativeness of GO terms for a gene set is the conditional probability of a GO term annotated to the given gene set with a hypergeometric distribution [25, 12, 20, 56, 47]. Because of the hierarchical structure of GO, the ‘true path rule’ exists in GO terms: if a gene is annotated with a term, it is also annotated with the ancestors of the term [136]. Thus, to compute the representativeness

of a GO term/category, one must consider the GO enrichment by adding the possibility of extracting enriched categories.

The GO-significant genes are the set of genes that have a large number of annotations involving a set of GO terms. To determine whether a set of genes is GO-significant for a GO terms set, we computed the p-value (p: probability) of a gene set with respect to a set of GO terms and their enrichment. The *classic* p-value is compared against random associations between the GO terms and the set of genes, and defined as:

$$p = 1 - \sum_{i=1}^{k-1} \frac{\binom{M}{i} \binom{N-M}{n-i}}{\binom{N}{n}} \quad (\text{Eq. 3.1})$$

where N is the total number of genes in the GOA data, M is the total number of genes annotated by the GO terms, n is the size of the specified list of genes, k is the number of genes in the specified list annotated by the GO terms. When the p-value is computed on a set of GO terms, Bernoulli correction [25] is applied on significance values to choose GO-significant gene set.

In our approach, we used the algorithms in topGO [1] to score GO group by also considering the underlying GO graph topology. In [1], the authors proposed *elim* method to investigate the nodes independently in the GO graph bottom-up. Since the *elim* algorithm identified the most specific nodes bottom-up, more significant nodes on higher levels in the graph may be missed. The *weight* method was then proposed as an alternative, to detect the locally most significant terms in the GO graph by comparing the connected nodes (a parent and its child). These two algorithms were combined with the *classic* method in Eq. 3.1 in *combine* method, which reported the p-value as the mean of the p-values obtained with *classic*, *elim* and *weight* algorithm, computed on a log scale. If pValue denotes the vector of the three p-values for a GO term, then the *combine* p-value is defined by

$$\bar{p} = \exp\left(\frac{1}{3} \sum_{i=1}^3 \log(pValue[i])\right) \quad (\text{Eq. 3.2})$$

The p-value for enrichment of GO annotations is used to evaluate the functional coherence of a group of genes/proteins, for example, Zheng et al. [158] used it to evaluate the closeness and strength of connectivity of the proteins in the network. In our approach, we incorporated GO enrichment to infer gene associations and regulations.

Due to the hierarchical structure of GO, the terms at top level may be not specific enough to infer the functional associations of genes, we thus need to filter more specific terms for GRN inference. Hierarchical depth of the term from the GO root [46] is a direct measure to filter specific terms heuristically, according to [3], the information content is an more accurate measure to evaluate the use of available information in a given a GOA file. Specifically, the information content $I(t)$ of a GO term $t \in \mathcal{T}$ is given by

$$I(t) = -\log(p(t)) \quad (\text{Eq. 3.3})$$

where the term probability $p(t) = N(t)/N(\text{root}(t)) = M/N$, $M = N(t)$ is the number of genes annotated by term t (i.e., occurrences of term t), and $N = N(\text{root}(t))$ is the number of genes annotated by $\text{root}(t) \in \{BP, MF\}$ in the given GOA data, which is also the number of genes in the GOA. Given a GOA file, due to the ‘true path rule’ in an ontology, the number of genes annotated to ancestor GO terms is much more than descendant terms, and the root term achieves maximum. By defining the probability $p(t)$ of a GO term t as the number of genes annotated by it over the root term, p is monotonically non-decreasing as one moves up from the descendant term to ancestor term: if t “is-a” a , or t is “part-of” a , then $p(t) < p(a)$. Thus, the information content $I(t)$ also reflects the hierarchical depth information of the term from the GO root.

To combine the representativeness of GO terms for a gene set, we modified the GO term probability in a given gene set by considering the conditional probability of a GO term annotated to the given gene set. Given a gene set $\mathcal{G}$ of n genes and its total term-set $\mathcal{T}$, for term $t \in \mathcal{T}$, suppose k genes in $\mathcal{G}$ are annotated by term t , then its probability is modified as

$$p(t, k, n) = p(k, n|t)p(t) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}} \frac{M}{N}, \quad k \in \{0, 1, \dots, n\} \quad (\text{Eq. 3.4})$$

Note that this definition in fact measures the representativeness of GO terms for a given gene set. Since GO term represents the molecular function or biological process, we in fact obtained the representative pathways of the gene set from it. The more representative pathways make the inference more biologically feasible, and enhance the confidence of inference. We thus included this modified term probability into the inference of confidence score assignment.

3.2.2 Functional Similarity of Genes

Semantic similarity has been previously proposed to compare concepts within an ontology. It can evaluate the specificity of a GO term's underlying concept in a given GO annotation. Resnik similarity [117], Jiang and Conrath's distance [73], and Lin's similarity [86] were three of the most popular semantic similarity measures. Resnik et al. [116, 117] measured the semantic similarity of two terms via the information content of their lowest common ancestors (LCA). A related measure has been developed in [73] and [86], which depended on both the information content of the terms and their LCA. Pandey et al. [110] proposed a statistically motivated functional similarity measure that considers both functional specificity and the distribution of functional attributes across entity groups. Inspired by their success of predicting interaction pairs, we incorporated the GO representativeness of two genes into functional similarity measure.

For each term $t \in \mathcal{T}$, previous semantic similarities were defined based on the term probability $p(t)$ that is the probability of finding the descendants of term t in the GO annotation database. In the previous section, we modified the term probability as the joint probability in Eq. 3.4 by considering the representativeness of the term for genes of interest. When we compared two genes, the joint probability $p(t, k, n)$ in fact accounts the imbalance of terms annotated to only one gene $p(t, 1, 2)$ and both of two genes $p(t, 2, 2)$. If t and t' are two terms and $a(t, t')$ represents the set of parent terms shared by both t and t' , then the similarity between two terms is computed in the following Resnik [117] and Lin's [86] measures, respectively:

$$Sim_{Resnik}(t, t') = -\log(p(LCA(t, t'))) \quad (\text{Eq. 3.5})$$

$$Sim_{Lin}(t, t') = \frac{2 \times \log(p(LCA(t, t')))}{\log(p(t)) + \log(p(t'))} \quad (\text{Eq. 3.6})$$

$$(\text{Eq. 3.7})$$

where $p(LCA(t, t')) = \min_{t^* \in a(t, t')} p(t^*)$.

To consider the functional similarity between two genes, as each gene is annotated by a list of GO terms, we need to consider the semantic similarity between two GO terms set. Let GOScore be the measure of functional similarity between two genes with respect to either their biological process (BPscore) or molecular function (MFscore). Given two

genes i and i' , with annotated GO (BP or MF) terms set T_i and $T_{i'}$, respectively, GO similarity scores of two genes are estimated by the *average* inter-set similarity of terms in T_i and $T_{i'}$, or the *maximum* GO similarity measure [90]. Specifically, in our approach, we adopted the *maximum* Resnik similarity to evaluate the specificity (most specific LCA) of two genes, and used *average* inter-set Lin's similarity to evaluate how close the terms to the LCAs as follows.

$$\begin{aligned}
 BPscore_{Resnik}(i, i') &= \max_{t \in T_i \cap \{BP\}, t' \in T_{i'} \cap \{BP\}} Sim_{Resnik}(t, t') & (Eq. 3.8) \\
 MFscore_{Resnik}(i, i') &= \max_{t \in T_i \cap \{MF\}, t' \in T_{i'} \cap \{MF\}} Sim_{Resnik}(t, t') \\
 BPscore_{Lin}(i, i') &= \frac{1}{|T_i \cap \{BP\}| \cdot |T_{i'} \cap \{BP\}|} \sum_{t \in T_i \cap \{BP\}, t' \in T_{i'} \cap \{BP\}} Sim_{Lin}(t, t') \\
 MFscore_{Lin}(i, i') &= \frac{1}{|T_i \cap \{MF\}| \cdot |T_{i'} \cap \{MF\}|} \sum_{t \in T_i \cap \{MF\}, t' \in T_{i'} \cap \{MF\}} Sim_{Lin}(t, t')
 \end{aligned}$$

Finally, the combined GOscore is calculated from the BPscore and the MFscore of a pair of gene products as follows:

$$\begin{aligned}
 GOscore_{Lin}(i, i') &= \frac{1}{2} \left[\left(\frac{BPscore_{Lin}(i, i')}{\max(BPscore_{Lin})} \right)^2 + \left(\frac{MFscore_{Lin}(i, i')}{\max(MFscore_{Lin})} \right)^2 \right] & (Eq. 3.9) \\
 GOscore_{Resnik}(i, i') &= \frac{1}{2} \left[\left(\frac{BPscore_{Resnik}(i, i')}{\max(BPscore_{Resnik})} \right)^2 + \left(\frac{MFscore_{Resnik}(i, i')}{\max(MFscore_{Resnik})} \right)^2 \right]
 \end{aligned}$$

where $\max(BPscore)$ and $\max(MFscore)$ denote the *maximum* score for biological process and molecular function, respectively. The GOscore ranges from 0 for no functional similarity to 1 for the *maximum* functional similarity.

Using a similar method to simRel [125] to combine the semantic similarity of Lin and Resnik, we combine Lin's and Resnik's biological similarities together as a combined biological similarity as defined in Eq. 3.10.

$$FunSim(i, i') = GOscore_{Lin}(i, i') \times (1 - \exp(-GOscore_{Resnik}(i, i'))) \quad (Eq. 3.10)$$

where $GOscore_{Lin}$ is computed from Lin's semantic similarity, which measures the average relative similarity of the GO terms set of two genes; $GOscore_{Resnik}$ is computed from

Resnik's smantic similarity, which measures the *maximum* common GO terms annotated to two genes. Thus, FunSim score considers both how close terms are to their LCA as well as how specific the LCA is within two genes.

As we modified the term probability Eq. 3.4 by also considering the conditional probability of terms annotating the two genes, the functional similarity measure computed from it by incorporating GO enrichment, denoted by "FS-GOE", is summarized as follows. Firstly, the probability of terms is computed by

$$p(t, k, n) = p(k, n|t)p(t) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}} \frac{M}{N}, \quad k \in \{0, 1, \dots, n\} \quad (\text{Eq. 3.11})$$

Then, denote the number of genes (in given two genes) annotated by term t , t' , and $LCA(t, t')$ as k , k' , and k^* , respectively, and denote the semantic similarities computed from the modified term probability by Resnik similarity [117] as $Sim_1(t, t')$ and Lin's similarity [86] as $Sim_2(t, t')$, they are computed as:

$$\begin{aligned} Sim_1(t, t') &= -\log(p(LCA(t, t'), k^*, n)) \\ Sim_2(t, t') &= \frac{2 \times \log(p(LCA(t, t'), k^*, n))}{\log(p(t, k, n)) + \log(p(t', k', n))} \end{aligned} \quad (\text{Eq. 3.12})$$

and corresponding GO scores are computed by

$$\begin{aligned} BPscore_1(i, i') &= \max_{t \in T_i \cap \{BP\}, t' \in T_{i'} \cap \{BP\}} Sim_1(t, t') \\ MFscore_1(i, i') &= \max_{t \in T_i \cap \{MF\}, t' \in T_{i'} \cap \{MF\}} Sim_1(t, t') \\ BPscore_2(i, i') &= \frac{1}{|T_i \cap \{BP\}| \cdot |T_{i'} \cap \{BP\}|} \sum_{t \in T_i \cap \{BP\}, t' \in T_{i'} \cap \{BP\}} Sim_2(t, t') \\ MFscore_2(i, i') &= \frac{1}{|T_i \cap \{MF\}| \cdot |T_{i'} \cap \{MF\}|} \sum_{t \in T_i \cap \{MF\}, t' \in T_{i'} \cap \{MF\}} Sim_2(t, t') \\ GOscore_1(i, i') &= \frac{1}{2} \left[\left(\frac{BPscore_1(i, i')}{\max(BPscore_1)} \right)^2 + \left(\frac{MFscore_1(i, i')}{\max(MFscore_1)} \right)^2 \right] \\ GOscore_2(i, i') &= \frac{1}{2} \left[\left(\frac{BPscore_2(i, i')}{\max(BPscore_2)} \right)^2 + \left(\frac{MFscore_2(i, i')}{\max(MFscore_2)} \right)^2 \right] \end{aligned} \quad (\text{Eq. 3.13})$$

Finally, the FS-GOE is given by

$$FS - GOE(i, i') = GOscore_2(i, i') \times (1 - \exp(-GOscore_1(i, i'))) \quad (\text{Eq. 3.14})$$

Note that FS-GOE score is computed by using the modified probability of terms $p(t, k, n)$.

For example, to compute FunSim score of MBP1 and MCM1 in Figure 3.1, as $T_{MBP1} = \{GO:0000083, GO:0000122, GO:0006355\}$, $T_{MCM1} = \{GO:0006357, GO:0006355\}$, according to the hierarchical relations in Figure 3.1, $LCA(GO:0000083, GO:0006357)$, $LCA(GO:0000122, GO:0006357)$, and $LCA(GO:0006355, GO:0006357)$ are all $GO:0006357$; then, the *maximum* BPscore of Resnik's semantic similarity is computed as

$$BPscore_{Resnik}(MBP1, MCM1) = \max_{t, t'} -\log(p(LCA(t, t'))) = -\log(p(GO : 0006357)); \quad (\text{Eq. 3.15})$$

Suppose in yeast GOA file, there are M genes annotated by term $GO:0006357$, and N genes annotated by BP terms, then, $p(GO : 0006357) = \frac{M}{N}$. As $GO:0006357$ is annotated only one gene MCM1 in given two genes (MBP1, MCM1), thus the modified term probability of $GO:0006357$ is

$$p(GO : 0006357, 1, 2) = \frac{\binom{M}{1} \binom{N-M}{1}}{\binom{N}{2}} \frac{M}{N} \quad (\text{Eq. 3.16})$$

and the *maximum* BPscore of Resnik's semantic similarity computed by the modified term probability is

$$BPscore_1(MBP1, MCM1) = \max_{t, t'} -\log(p(LCA(t, t'))) = -\log(p(GO : 0006357, 1, 2)); \quad (\text{Eq. 3.17})$$

The average BPscore of Lin's semantic similarity is computed as

$$BPscore_{Lin}(MBP1, MCM1) = \frac{1}{|T_{MBP1}| \times |T_{MCM1}|} \sum_{t, t'} \frac{2 \times \log(p(LCA(t, t')))}{\log(p(t)) + \log(p(t'))}. \quad (\text{Eq. 3.18})$$

and the average BPscore of Lin's semantic similarity computed by the modified term probability is

$$BPscore_2(MBP1, MCM1) = \frac{1}{|T_{MBP1}| \times |T_{MCM1}|} \sum_{t, t'} \frac{2 \times \log(p(LCA(t, t'), k^*, n))}{\log(p(t, k, n)) + \log(p(t', k', n))}. \quad (\text{Eq. 3.19})$$

Similarly, one can calculate the other scores: $MFscore_{Resnik}(MBP1, MCM1)$, $MFscore_{Lin}(MBP1, MCM1)$, $MFscore_1(MBP1, MCM1)$ and $MFscore_2(MBP1, MCM1)$. Then, use Eq. 3.9

and Eq. 3.13 to combine MF ontology and BP ontology for the GOscores: $GOscore_{Resnik}(MBP1, MCM1)$, $GOscore_{Lin}(MBP1, MCM1)$, $GOscore_1(MBP1, MCM1)$ and $GOscore_2(MBP1, MCM1)$.

Lastly, the functional similarity score is given by

$$\begin{aligned} FunSim(MBP1, MCM1) &= GOscore_{Lin}(MBP1, MCM1) & (Eq. 3.20) \\ &\times (1 - \exp(-GOscore_{Resnik}(MBP1, MCM1))) \end{aligned}$$

and FS-GOE score computed from modified term perobabilities is given as

$$\begin{aligned} FS - GOE(MBP1, MCM1) &= GOscore_2(MBP1, MCM1) & (Eq. 3.21) \\ &\times (1 - \exp(-GOscore_1(MBP1, MCM1))) \end{aligned}$$

3.3 Gene Ontology Networks

Consider a set $\mathcal{G} = \{i\}_{i=1}^n$ of n genes with gene i associated to GO term-set T_i . The total term-set $\mathcal{T}$ of all genes is given by $\mathcal{T} = \bigcup_{i=1}^I T_i$. Let $r(t, t') \in \mathcal{R}$ denote a GO relation between terms $t, t' \in \mathcal{T}$. All GO relations between terms in the term-set $\mathcal{T}$ are represented as $\mathcal{E} = \{r(t, t') : t, t' \in \mathcal{T}, \exists r(t, t') \in \mathcal{R}\}$. The pair $(\mathcal{T}, \mathcal{E})$ thus constitute the GO network (GON). An example of GON is shown in Figure 3.1.

Figure 3.1: An example of GON. Circles represent GO terms, genes annotated by the term are shown beside, bold directed edges between terms are GO semantic relations.

Given a gene set, representative GO terms must be chosen to construct the corresponding GON. The representative GO terms are filtered out by using p-value (combined) [1], which are the mean of the P-values obtained with “classic”, “elim” and “weight” algorithm considering both GO enrichment and GO graph structures as shown in Eq. 3.2.

3.4 Extracting Functionally Associated Networks

Guo et al. [57] used GO-derived similarity measures to detect direct and indirect interactions within regulatory pathways. Motivated by the success of gene-set approach in expression patterns detection of group genes instead of individual genes in [66], we integrated the pathway organization in GRN inference. Given a big gene network, one

can extract functionally associated sub-networks based on GO functional coherence. We proposed Algorithm 1 to extract FANs from a given GRN by using the GO group testing. The algorithm starts with a initial gene or gene set G , and do (i) find the over-represented (significant) GO annotations $\bar{T}$ for it using GO group testing described in Section 3.2.1; (ii) find all neighbor genes N of G from the whole network, choose those who are over-represented by $\bar{T}$; then update the new gene set G ; and do (i) and (ii) iteratively until no edges can be added.

Algorithm 1 Extracting Functionally Associated Sub-networks

- Step 0. Given the complete network E , an initial gene g_0 , the significance level α , initialize the sub-network $G = \{g_0\}$
 - Step 1. Find the GO term set T corresponding to genes in G
 - Step 2. For each term $t \in T$, calculate its p-value $p(t)$; Set the significant terms set $T_\alpha = \{t : p(t) < \alpha, t \in T\}$
 - Step 3. Find the set of gene neighbors $N = \{g | \exists (g, g') \in E, \forall g' \in G\}$
 - Step 4. If $N \neq \{\}$, for each gene $g \in N$, set $G' = G \cup \{g\}$
 Do significance Bernoulli correction $\hat{\alpha} = \alpha/|T_\alpha|$, calculate the p-value of term set T_α on G' as $p(T_\alpha, G')$
 If $p(T_\alpha, G') < \hat{\alpha}$, set $G = G'$ and $N = N \setminus \{g\}$
 - Step 5. Go to Step 1 until $N = \{\}$
-

Since the modified FS-GOE score considers the GO representativeness, one can directly use hierarchical clustering based on functionally similarity to extract functional associated pathways. Note that hierarchical clustering implemented on small networks will give more accurate clustering results. Smith et al. [127] studied the stability of hierarchical clustering, and they showed that its stability decreased when the sample size increased. So on big networks, hierarchical clustering results in unstable clusters. Thus, in our experiments, for big network, we used Algorithm 1 to extract FANs only based on functional coherence testing; on smaller networks, we employed hierarchical clustering based on FS-GOE score to also consider semantic similarity.

3.5 Gene Regulatory Networks from GON

Recall that in GON, the *regulate* relationship denotes the occurrence of one biological process that directly affects the manifestation of another process or quality, for instance,

process t regulates process t' means that if both processes occur, t always regulates t' . If two genes i and i' are involved (participate) in the two processes t and t' , then a regulation might happen between the two genes.

Suppose gene i is annotated by GO term t and gene i' is annotated by GO term t' . If GO term t regulates term t' , then we infer that gene i might regulate i' . If this inference comes from more specific terms, gene i should regulate i' with high confidence. In our GRN inference procedure, we therefore choose the most reliable regulate inference path. There are a very few direct *regulate* relations in the existing GO database, making it difficult to infer a GRN. As such, we propose to induce transitive *regulate* relations among GO terms, from which we infer a GRN based on both direct and transitive gene regulations derived from the GON. A transitive regulation is a regulation between two genes via one or more transitive genes in the absence of a direct regulation [159]. For example, suppose gene g_1 regulates gene g_2 directly and gene g_2 relates to gene g_3 . Then since there is no direct regulation between g_1 and g_3 , if we can infer a regulatory relation from g_1 to g_3 , we say g_1 transitively regulates g_3 through transitive gene g_2 .

Consider the transitivity rule:

$$\text{if } t_a \xrightarrow{r_1} t_b \text{ and } t_b \xrightarrow{r_2} t_c, \text{ then } t_a \xrightarrow{r_3} t_c \quad (\text{Eq. 3.22})$$

where $t_a, t_b, t_c \in \mathcal{T}$ and $r_1, r_2, r_3 \in \mathcal{R}$. By using the notations of rule deduction, the above transitive relation is written as

$$r_3 = r_1 \wedge r_2 \quad (\text{Eq. 3.23})$$

According to <http://www.geneontology.org/GO.ontology.relations.shtml>, for any $r \in \mathcal{R} = \{is - a, part - of, regulates\}$, the following four transitivity relations are valid among the GO terms.

$$r = r \wedge is - a \quad (\text{Eq. 3.24})$$

$$r = is - a \wedge r \quad (\text{Eq. 3.25})$$

$$regulates = regulates \wedge part - of \quad (\text{Eq. 3.26})$$

$$part - of = part - of \wedge part - of \quad (\text{Eq. 3.27})$$

Note that $part - of \wedge regulates \neq regulates$ and $regulates \wedge regulates \neq regulates$.

Consider a path $(t_j)_{j=0}^J$ in GON where t_0 and t_J denote the source and destination terms, respectively; and $r(t_j, t_{j+1})$ is the parent-child relation between parent term t_j and child term t_{j+1} . Using parent-child relations $\{ r(t_j, t_{j+1}) \}_{j=1}^{J-1}$, each term t_j can induce a relation from the source term. Denote path $\pi_J = (t_j)_{j=0}^J$, and let $r(\pi_J) = r(t_0, t_J)$ denote the inferred relation along path π_J by applying transitive rules to parent-child relations, we have

$$r(\pi_j) = r(\pi_{j-1}) \wedge r(t_{j-1}, t_j). \quad (\text{Eq. 3.28})$$

We then assigned a confidence score function $\sigma(\pi)$ for each inferred path π by considering both the number of steps and the information content of the terms along the inferred paths. The confidence score $\sigma(\pi)$ should give preference to paths with fewer inference steps and more informative terms, defined as

$$\sigma(\pi_j) = \sigma(\pi_{j-1}) + \Delta_{r(t_{j-1}, t_j)}(t_{j-1}, t_j) \quad (\text{Eq. 3.29})$$

where $\sigma(\pi_j)$ is the score assigned to the inferred path π_j from source t_0 to term $t_j \in \mathcal{T}$ and $\Delta_{r(t, t')}(t, t')$ is the score assigned to relation $r(t, t') \in \mathcal{R}$ between terms $t, t' \in \mathcal{T}$.

The cost for deducing a relation between two terms should facilitate the selection of the most informative inferred path. Semantic similarity of GO terms based on their information content, i.e., Lin's semantic similarity measure [86] and Jiang's semantic distance [73] can be used to define the cost of deducing a relation between two terms. For example, $\Delta_{r(t, t')}(t, t') = 1 - \text{Sim}(t, t')$, where $\text{Sim}(t, t')$ is the semantic similarity of terms t, t' . Jiang's semantic distance can also be directly used as the cost as $\Delta_{r(t, t')}(t, t')$. Lastly, different costs can be assigned to each of the three types of relations $r(t, t')$, e.g., Wang's semantic similarity in [147] define different contribution factors for *is - a* and *part - of* relations are 0.8 and 0.6, respectively.

When there exists no relation $r(t, t')$, the cost is assigned as $\Delta_{r(t, t')}(t, t') = \infty$ to avoid inferring empty relations. Thus, the path with the minimum score σ_π is the path inferred collectively using the fewest steps and along the most informative terms, i.e., the most reliable inferred path. If an inferred path ends with the deduced relation *regulate* at the destination, then there is a regulatory path (RP) between the source and the destination. We proposed an algorithm to detect the most reliable RP between two terms $s, d \in \mathcal{T}$ by using the deduction scores in Dijkstra's shortest path algorithm [74]. For each node

$v \in \mathcal{T}$, we use an indicator vector to represent the deduced relations r_v , and $\sigma(r_v, v)$ is a matrix to record the current minimum distance/cost to deduce r_v at term v along path $\pi(r_v, v)$.

Initially, the source term is assigned an *is - a* relation as it does not change the first transitive relation. Subsequently for each $R \in \mathcal{R}$, S_R denotes the unvisited node set with current relation R , and we iteratively choose $u^* = \arg \min_{u \in S_R} \{\sigma(R, u)\}$ as the starting node of the following iteration.

At each inference step, if $\sigma(r_{u^*}, u^*) + \Delta_{r(u^*, v)}(u^*, v) < \sigma(r_v, v)$, we update the current deduced score for the three relations by $r_v = r_{u^*} \wedge r(u^*, v)$, $\sigma(r_v, v) = \sigma(r_{u^*}, u^*) + \Delta_{r(u^*, v)}(u^*, v)$. The iteration stops when all S_R are empty. Upon termination, if $\sigma(\text{regulate}, d) < +\infty$, $\pi(\text{regulate}, d)$ is the most-reliable RP from term s to d , otherwise no RP from term s to d exists. Given a source term and target term, Algorithm 2 finds the most reliable RP.

Algorithm 2 Regulatory Paths from GON (RP-GON)

- Step 0. Given source term s , target term d , term set $\mathcal{T}$, and inference cost matrix Δ
Step 1. Set $r_s(\text{is} - a) = 1$; $\forall u \in \mathcal{T}, R \in \mathcal{R}$. Set $S_R = \mathcal{T}$, $\pi(R, u) = \{u\}$, $\sigma(R, u) = +\infty$ except $\sigma(\text{is} - a, s) = 0$
Step 2. For $R \in \mathcal{R}$, If $S_R \neq \{\}$
 Choose $u^* = \arg \min_{u \in S_R} \{\sigma(R, u)\}$, set $S_R = S_R \setminus \{u^*\}$
 If $\sigma(R, u^*) \neq +\infty$: for each $(u^*, v) \in \mathcal{E}$, if $\sigma(r_{u^*}, u^*) + \Delta_{r(u^*, v)}(u^*, v) < \sigma(r_v, v)$
 Update $r_v = r_{u^*} \wedge r(u^*, v)$, $\sigma(r_v, v) = \sigma(r_{u^*}, u^*) + \Delta_{r(u^*, v)}(u^*, v)$, $\pi(r_v, v) = \{u^*\} \cup \pi(r_v, v)$
Step 3. If $\sigma(\text{regulate}, d) < +\infty$, return $\pi(\text{regulate}, d)$ as the most-reliable RP
-

Up till now, we have only determined the most reliable RP between GO terms. For genes i, i' with GO annotations $T_i, T_{i'}$ respectively, if there exists a RP from $t \in T_i$ to $t' \in T_{i'}$, we can infer that gene i regulates gene i' . To assign the confidence score for this inferred path, we used the minimum regulatory path score of all regulatory paths (if any) as follows.

$$C(i, i') = \min\{\sigma(\pi(t, t')) | r(\pi(t, t')) = \text{regulate}\} \quad (\text{Eq. 3.30})$$

With the regulatory paths, we can construct a GRN with confidence score based on direct and transitive *regulate* relations among the GO terms. For example, in Figure 3.1, from the shortest regulatory path “GO:0000083 (CLN3) *is - a* GO:0006357 (MCM1)

is – a GO:0006355 (MCM1) *regulate* GO:0006351 (SWI4)”, we could infer that “CLN3 transitively regulates SWI4 via MCM1”.

3.6 Experimental Results

3.6.1 Benchmark Networks

The complete gene network of *E.coli* was obtained from RegulonDB database(<http://regulondb.ccg.unam.mx/>) [122]. We randomly chose three sub-networks, Net1, Net2 and Net3, from it for experiments.

Yang et al. [153] provided yeast transcriptional regulatory pathways (YTRPs) by enumerating all directed shortest paths from YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking, <http://www.yeasttract.com/>) Database [135], which contained all transcriptional regulations derived from literature and TF-binding evidence. Transcriptional regulatory pathways (TRPs) indicate indirect regulations between TFs and target genes. To test the power of our Algorithm 2 to recover direct regulation, especially the transitive regulations, we chose some long TRPs in cell cycle with many common genes, and thereby we mixed them up into a transcriptional regulatory network. A cell cycle is comprised of four phases: (i) Gap 1 (G1) is the checkpoint to ensure that it is ready for division, (ii) Synthesis (S_Phase) involves DNA replication, (iii) Gap 2 (G2) is another checkpoint to ensure that the cell is ready to enter the next phase, and (iv) Mitotic (M_Phase) refers to cell division. Two yeast cell cycle transcriptional regulatory networks (Net4 and Net5) were randomly chosen from YEASTRACT database [153, 135] as benchmark networks for experiments.

To show how well the proposed functional similarity measure, FS-GOE score, is able to recover benchmark networks, we also chose the integrated functional association score from STRING database (<http://string-db.org/>) as a comparison since it quantitatively integrates multiple data sources including genome context, gene expression and literature, for generating functional association network (FAN) [97]. The STRING score combined multiple scores to form a single score that expresses a higher confidence by multiplying the probabilities of associations not predicting a functional interaction.

3.6.2 Gene Ontology Annotations for Gene Sets

Gene Ontology terms and relations corresponding to the genes in the networks were obtained from files associated with the GO annotation database (<http://www.geneontology.org/GO.downloads.annotations.shtml>). The corresponding informative GO terms related to the target genes were also selected from the gene association files. Only GO terms involved in the molecular function and biological process are considered. Each GO annotation is classified into one of 5 descending order of quality categories: experimental, computational, author statement, curator statement, and automatic. Annotations derived through direct experiments are deemed higher quality compared to others [118]. We only considered GO terms with the top two quality levels: computational and experimental.

3.6.3 Gene Expression Time Series Data

The gene expression (GE) time series data for *E.coli* were collected from “GSE6425” in NCBI Gene Expression Omnibus (GEO) database. We chose the transcriptional expression of MG1655 harvested from various time points during aerobic or anaerobic growth in Luria-Bertani medium. The data set contained two replicates, each of 11 observations taken at various intervals with 30-minute, 45-minute, 1 hour and 45 minutes, 4-hour, 6-hour and 8-hour. The GE data for yeast cell cycle are obtained from [107], where we chose the transcriptional expression time-series of wild-type cells. Synchronous populations of early G1 cells were collected by centrifugal elutriation. Cell aliquots were then harvested at 16-minute intervals for 270-minute (equivalent to about two cell cycles in the wild type). It provided results of two independent experiments, each of 15 observations taken at 16-minute interval.

3.6.4 GRN Inference from GE Time Series Data

As comparison of recovering regulatory networks, we adopted state-of-the-art inference method for building GRN from time series gene expression (GE) data: DBN, DDE and fusion correlation with linear regression model, as baselines.

Given time courses of expressions of I genes $\mathcal{X} = \{x_i(t)\}_{i=1, t=1}^{I, T}$, where $x_i(t)$ denotes the value of the expression of gene i at time t , and T denotes the total number of time points which are assumed to be equally spaced.

DBN assumes first-order Markov dependence of gene expressions over time:

$$p(x(t)|x(t-1), \dots, x(1)) = p(x(t)|x(t-1))$$

Thus, in DBN, the observations of parent genes is $T-1$. The DBN is defined by a graph G , a family of conditional probability distributions, and their parameters $\theta = \{\vartheta_i\}$, which together specify the joint distribution over the variables $p(\mathcal{X}|G, \theta)$.

$$p(\mathcal{X}|G, \theta, r) = \prod_{i=1}^I \prod_{t=2}^T p(x_i(t)|\pi_i(t-1), \vartheta_i),$$

where each node $x_i(t)$ depends only on its parent nodes' previous expression $\pi_i(t-1)$. The parameter matrix θ is comprised of I vectors, where each vector ϑ_i specifies the local probability distribution. Since it is NP-hard to learn BN, we employed a polynomial time algorithm, "GlobalMIT", to learn the globally optimal DBN structure [143].

A representation of DDE, GeneReg [68], used B-spline to generate predicted time points based on GE data, and then predicted various delays in GRNs by using linear regression models. It used transcription factors (TFs) as regulator constraints. The time delay linear regression model in GeneReg [68] is defined as:

$$\tilde{x}_i(t) = \sum_{j=1}^{|\pi_i|} a_j \tilde{x}_j(t - \tau_j) \quad (\text{Eq. 3.31})$$

where π_i is restricted as transcription factors (TFs) set; $\tilde{x}_i(t)$ is the relative expression level of gene i at time point t to the baseline; $\tilde{x}_j(t - \tau_j)$ is the relative expression level of TF j at time point $t - \tau_j$, and τ_j is the time delay of the regulation of TF j to gene i ; a_j is the regression coefficient of TF j .

Mundra et al. [102] introduced the two-step procedures to derive time-delayed GRN from time series gene expression data: for a given gene, first, determine the probable time lags of regulations by other genes; then LASSO (least absolute shrinkage and selection operator) regression is used to derive the regulators.

They used time-lagged correlation to determine the probable time lags of regulations. If gene j regulates gene i , the unbiased time-lagged correlation is given as follows.

$$\hat{C}(x_i, x_j, \tau) = \frac{1}{T - \tau} \sum_{t=1}^{T-\tau-1} x_i(t + \tau)x_j(t) \quad (\text{Eq. 3.32})$$

The most probable lag was determined by maximizing absolute time-lagged correlation.

Then, they used LASSO, a penalized regression technique in multivariate auto-regression (MVAR) framework, to derive the regulators for a given gene as follows.

$$x(t) = \sum_{\tau=1}^r \beta^\tau x(t - \tau) + \epsilon(t) \quad (\text{Eq. 3.33})$$

where $\beta^\tau = \{\beta_{i,j}^\tau\}_{i=1,j=1}^{I,I}$ represents the strength (regression coefficients) of interactions between all pairs of genes for a model of order τ , and $\epsilon(t) = (\epsilon_i(t))_{i=1}^I$ denotes residuals which follow a Gaussian distribution with zero mean independently and identically distributed (i.i.d.).

3.6.5 Performance Measures

In the following sections, we evaluated the performance of the GRN inference methods (functional association networks from functional similarity scores, denoted by “Fun-Sim” and “FS-GOE”, and “STRING” score; GRNs learned from GE data, denoted by “DBN”, “GeneReg”, and “LASSO”, respectively; regulatory path method, denoted by “RP-GON”) on the benchmark networks.

Since the performance of methods using functional similarity scores and STRING score were based on thresholds, for their comparison, we also provided the Receiver Operating Characteristic (ROC) curves.

We used the following five measures, Precision (Pre.), Recall (Rec.), F1-score (F1), Accuracy (Acc.) and area under ROC curve (AUC) to evaluate the performances of different methods:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \end{aligned} \quad (\text{Eq. 3.34})$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall},$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}.$$

where TP is number of true positives, FP is number of false positives, FN is number of false negatives, and TN is the number of true negatives between the inferred and target network. As the target networks extracted from the databases are still under development, they were not complete; some FP edges may also be true edges which may need some lab experimental validation. Therefore, TP showed that how many already known edges were predicted from inference models, FP indicates how many incorrect edges or maybe new edges were predicted in inference models, and FN indicates how many known edges were not recovered by inference models.

3.6.6 Extraction of Associated Networks

As the complete gene network for *E.coli* was obtained from RegulonDB database [122], we extracted three functionally associated networks (FANs), FAN1, FAN2 and FAN3, from it by using Algorithm 1. And to extract FANs from the two smaller randomly selected yeast networks, Net4 and Net5, we used hierarchical clustering method based on the FS-GOE score to find their longest functionally associated pathways (FAP) as FAP4 and FAP5, respectively. Figure 3.2 showed the regulatory structures of the three *E.coli* FANs and two yeast networks with their FAPs.

The functional coherence of selected *E.coli* and yeast networks comparing with FANs and FAPs was evaluated by the R package “topGO” [1]. The results were shown in Table 3.1. It can be seen that all the extracted FANs and FAPs were GO-significant (with P-value < 0.05). They were also much more GO-significant than the randomly selected networks. The FANs and FAPs were thus highly functionally coherent.

Table 3.1: Functional evaluation of randomly selected networks and functionally associated networks/pathways

Species	Pathways	over-represented GO annotations	p-value (combined)
<i>E.coli</i>	Net1	GO:0006310 (DNA recombination)	0.1687
<i>E.coli</i>	FAN1	GO:0006261 (DNA-dependent DNA replication)	0.0183
<i>E.coli</i>	Net2	GO:0042330 (taxis)	0.0223
<i>E.coli</i>	FAN2	GO:0019395 (fatty acid oxidation)	0.0003
<i>E.coli</i>	Net3	GO:0022607 (cellular component assembly)	0.1360
<i>E.coli</i>	FAN3	GO:0042330 (taxis)	0.000005
Yeast	Net4	GO:0060074 (synapse maturation)	0.0013
Yeast	FAP4	GO:0051488 (activation of anaphase-promoting complex activity)	0.0012
Yeast	Net5	GO:0008611 (ether lipid biosynthetic process)	0.0956
Yeast	FAP5	GO:0051983 (regulation of chromosome segregation)	0.0030

We compared the three functional similarity scores, FS-GOE score, previous FunSim score and the STRING score, for recovering the benchmark networks via threshold-based method. Gene interactions/associations were derived from gene pairs if their functional similarity scores exceeded certain thresholds. The thresholds were chosen heuristically by maximizing the F1-score on recovering the benchmark networks.

Table 3.2 and 3.3 showed the performance of three similarity scores on recovering the randomly selected networks comparing with FANs of *E.coli* and yeast, respectively. Figure 3.3 and 3.4 showed the ROC curves of STRING score, FunSim score, FS-GOE score on recovering randomly selected networks comparing with FANs of *E.coli* and yeast, respectively. It was observed that FS-GOE score significantly outperformed FunSim score, with 3%-7% improvement on F1-score on almost all networks. The exceptions were Net2, Net3 and FAP4, where FS-GOE score generated slightly lower F1-score than FunSim. However, it was seen from the figures that FS-GOE score in fact achieved very similar and comparable performance with FunSim score on these three networks. FS-GOE

score generally outperformed FunSim score on recovering all benchmark networks. This showed that incorporating GO enrichment helped to improve the functional similarity.

Table 3.2: Performance of extraction of association pathways by methods using various similarity scores on *E.coli* Nets and FANs.

Networks	Similarity Scores	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net1	STRING	0.2692	0.4375	0.3333	0.8571	0.7102
	FunSim	0.1455	0.5000	0.2254	0.7194	0.7438
	FS-GOE	0.1212	0.5000	0.1951	0.6633	0.7399
FAN1	STRING	0.2600	0.7222	0.3824	0.7857	0.8563
	FunSim	0.1935	0.3333	0.2449	0.8112	0.6629
	FS-GOE	0.2500	0.4444	0.3200	0.8265	0.7135
Net2	STRING	0.2917	0.5833	0.3889	0.8044	0.7622
	FunSim	0.1481	0.6667	0.2424	0.5556	0.7697
	FS-GOE	0.1563	0.6250	0.2500	0.6000	0.7666
FAN2	STRING	0.1724	0.6667	0.2740	0.7644	0.8294
	FunSim	0.1129	0.4667	0.1818	0.7200	0.7333
	FS-GOE	0.1296	0.4667	0.2029	0.7556	0.7333
Net3	STRING	0.2727	0.5000	0.3529	0.8044	0.7361
	FunSim	0.4286	0.3750	0.4000	0.8800	0.6647
	FS-GOE	0.2813	0.3750	0.3214	0.8311	0.6527
FAN3	STRING	0.0739	0.9286	0.1368	0.2711	0.9643
	FunSim	0.0833	0.0714	0.0769	0.8933	0.5357
	FS-GOE	0.1667	0.0714	0.1000	0.9200	0.5357

3.3.a: *E.coli* Net1.

3.3.b: The *E.coli* FAN1.

3.3.c: The *E.coli* Net2.

3.3.d: *E.coli* FAN2.

3.3.e: *E.coli* Net3.

3.3.f: *E.coli* FAN3.

Figure 3.3: The ROC curves of association pathways by methods using various similarity scores on *E.coli* Nets and FANs.

Table 3.3: Performance of extraction of association pathways by methods using various similarity scores on yeast Nets and FAPs

Networks	Similarity Scores	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net4	STRING	0.1575	0.6897	0.2564	0.5972	0.7470
	FunSim	0.2671	0.6724	0.3824	0.7813	0.7979
	FS-GOE	0.2713	0.8793	0.4146	0.7500	0.8742
FAP4	STRING	0.6000	0.5455	0.5714	0.7500	0.7600
	FunSim	0.4167	0.9091	0.5714	0.5833	0.7836
	FS-GOE	0.4000	0.9091	0.5556	0.5556	0.7255
Net5	STRING	0.2475	0.6405	0.3570	0.4352	0.7734
	FunSim	0.2974	0.8824	0.4448	0.4608	0.9027
	FS-GOE	0.3342	0.8039	0.4722	0.5600	0.8768
FAP5	STRING	0.5682	0.5952	0.5814	0.4375	0.5097
	FunSim	0.6667	1.0000	0.8000	0.6719	0.7825
	FS-GOE	0.6667	1.0000	0.8000	0.6719	0.7825

3.4.a: Yeast Net4.

3.4.b: Yeast FAP4.

3.4.c: Yeast Net5.

3.4.d: Yeast FAP5.

Figure 3.4: The ROC curves of association pathways by methods using various similarity scores on yeast Nets and FAPs.

It was shown that on the *E.coli* networks FAN1, FAN2, Net3, and FAN3, STRING outperformed FunSim scores due to the higher recall. Note that STRING combined information from multiple data sources (GE, bio-literature, and genomic data, etc.), while functional similarity scores (FS-GOE and FunSim) only computed from GO annotations, which were generally quite sparse in the *E.coli* networks. This is biological plausible since *E.coli* genes are functionally simpler than yeast genes. As an extreme case in FAN3, only 20 BP terms and 16 MF terms were annotated by the 15 genes of FAN3. And only 13 edges were predicted from functional similarity scores (range from 0.0075 to 0.1969) among five genes (flhC, ccmA, flhZ, flgM, flhA) because of their sparse GO annotations. Thus, the data points of functional similarity scores in the ROC curve of

FAN3 were only at the left corner as shown in Figure 3.3.

However, on the other *E.coli* networks, i.e., Net1 and Net2, when more GO information was used, the performances of functional similarity scores were rather comparable with STRING in Table 3.3 and Figure 3.3. And on yeast networks, i.e., Net4, Net5, FAP4 and FAP5, when GO information was much denser, functional similarity scores outperformed the STRING score much more as shown in Table 3.3 and Figure 3.3.

Speaking of network structure, it can be seen from Figure 3.2 that the *E.coli* networks were very sparse, while the yeast networks were much denser. Recall that the association networks recovered by all similarity scores were based on the thresholds which were determined/learned under the target networks' supervision. A dense target network with more training edges would help to determine a more accurate threshold for predicting gene interactions. The observation that functional similarity scores outperform STRING score on dense networks indicates that the former have better capacity to identify functional associations. This is consistent with the conclusion that GO has a good indicative power for genetic interactions [57, 92].

3.6.7 Building Regulatory Networks

To illustrate the efficacy of RP-GON method, we compared it with three state-of-the-art time-delayed GRN inference methods, DBN [143], GeneReg [68], and LASSO [102]. Table 3.4 and Table 3.5 showed the performance of various GRN inference methods on the randomly selected networks comparing with FANs of *E.coli* and FAPs of yeast, respectively.

Table 3.4: Performance on extraction of regulatory pathways on *E.coli* Nets and FANs.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net1	DBN	-	0	-	0.9184	0.5000
	GeneReg	0	0	-	0.9082	0.4944
	LASSO	0	0	-	0.8316	0.4944
	RP-GON	1.0000	0.0625	0.1176	0.9235	0.5312
FAN1	DBN	-	0	-	0.9082	0.5000
	GeneReg	0.1364	0.1667	0.1500	0.8265	0.5807
	LASSO	0	0	-	0.9031	0.5000
	RP-GON	0.8333	0.5556	0.6667	0.9490	0.7762
Net2	DBN	0.2500	0.0417	0.0714	0.8844	0.5131
	GeneReg	0.4000	0.0833	0.1379	0.8889	0.5390
	LASSO	0.0667	0.0417	0.0513	0.8356	0.5157
	RP-GON	0.5000	0.0833	0.1429	0.8933	0.5390
FAN2	DBN	0.3333	0.0667	0.1111	0.9289	0.5311
	GeneReg	0.1111	0.0667	0.0833	0.9022	0.5308
	LASSO	0.1765	0.4000	0.2449	0.8356	0.6878
	RP-GON	0.8571	0.4000	0.5455	0.9556	0.7000
Net3	DBN	0.1667	0.0417	0.0667	0.8756	0.5081
	GeneReg	0.4000	0.0833	0.1379	0.8889	0.5336
	LASSO	0	0	-	0.7867	0.4925
	RP-GON	0.6667	0.1667	0.2667	0.9022	0.5808
FAN3	DBN	0.2500	0.2143	0.2308	0.9111	0.6071
	GeneReg	0	0	-	0.7733	0.5000
	LASSO	0	0	-	0.8844	0.5000
	RP-GON	0.9000	0.6429	0.7500	0.9733	0.8214

Table 3.5: Performance of regulatory pathways methods on yeast Nets and FAPs.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net4	DBN	0.4783	0.1897	0.2716	0.8976	0.5834
	GeneReg	0.1687	0.2414	0.1986	0.8038	0.5754
	LASSO	0.1667	0.1552	0.1607	0.8368	0.5503
	RP-GON	0.2193	0.4310	0.2907	0.7882	0.6469
FAP4	DBN	1.0000	0.1818	0.3077	0.7500	0.5909
	GeneReg	0.4286	0.2727	0.3333	0.6667	0.5855
	LASSO	0.2000	0.0909	0.1250	0.6111	0.4800
	RP-GON	0.7500	0.2727	0.4000	0.7500	0.6364
Net5	DBN	0.8095	0.1111	0.1954	0.7760	0.5515
	GeneReg	0.2923	0.2484	0.2686	0.6688	0.5995
	LASSO	0.3111	0.1830	0.2305	0.7008	0.5793
	RP-GON	0.3431	0.2288	0.2745	0.7040	0.6127
FAP5	DBN	0.7143	0.1190	0.2041	0.3906	0.5108
	GeneReg	0.7000	0.1667	0.2692	0.4063	0.5168
	LASSO	0.5500	0.2619	0.3548	0.3750	0.4957
	RP-GON	0.8462	0.2619	0.4000	0.4844	0.6310

Compared with GRN inference methods from GE time-series data, RP-GON method significantly outperformed DBN, GeneReg and LASSO on all networks. Compared with functional similarity method on sparse networks *E.coli*, RP-GON method significantly improved the performance of recovering the regulatory pathways on FANs with much higher precision (83% on FAN1, 86% on FAN2, 90% on FAN3), slightly higher recall (56% on FAN1, 40% on FAN2, 64% on FAN3), and much higher F1-score (67% on FAN1, 55% on FAN2, and 75% on FAN3). However, compared with functional similarity method, RP-GON method performed not so well on randomly selected networks: on *E.coli* random networks, it generally predicted GRNs with higher precision but lower recall; on yeast networks, i.e., Net4 and Net5, it predicted GRNs with comparable precision, but lower recall. The relative low recall of RP-GON method may come from the lack of GO “regulate” relations, which may not yet recorded in current database, or rarely existed in the randomly selected gene sets. The better recall on FANs showed that constraining the RP-GON method within FANs helped to recover more biological reasonable GRNs.

RP-GON method performed well on FANs/FAPs, with precision ranging from 75% to 90%, and F1-scores ranging from 40% to 75%. Compared with *E.coli* random networks, RP-GON method significantly outperformed on all FANs with all metrics, precision, recall, F1-score, and AUC. Note that the yeast networks (Net4 and Net5) were randomly chosen from cell cycle transcriptional regulatory network, and FAP4 and FAP5 were derived from Net4 and Net5 by clustering based on functional similarity score. It was seen that on FAPs, RP-GON significantly improved the performance on the randomly chosen network, with 11% improvement of F1-score, 53% improvement of precision on Net4; and 13% improvement of F1-score, 50% improvement of precision on Net5. This showed that constraining the RP-GON method within FANs/FAPs can improve the performance of RP-GON method significantly. The effectiveness and rationale of our method were thus verified.

3.7 Discussion

In this chapter, we presented a method to improve the functional similarity measure of genes, i.e., FS-GOE, by considering both GO semantic similarities and GO representativeness. The functionally associated pathways were constructed based on genes’ improved

functional similarities. Within a functionally associated pathway, gene regulations were inferred based on GO “regulate” relation. Moreover, by applying transitive rules in GO relations, transitive regulations were detected from the regulatory path method, such as RP-GON.

Experiments were conducted both on randomly chosen networks and functionally associated networks of *E.coli* derived from RegulonDB [122], and yeast cell cycle transcriptional regulatory networks derived from YEASTRACT [135]. The proposed FS-GOE scoring scheme outperformed the existing functional similarity score, as well as the functional association score from STRING, on recovering almost all benchmark networks. Experimental results showed that our RP-GON method performed better on FANs than randomly chosen networks. This proves that our idea of constraining the regulatory path method within functionally associated pathway is biologically reasonable. Compared with state-of-the-art time-delayed GRN inference methods from GE data, such as DBN, GeneReg and LASSO, our RP-GON method shined on all benchmark networks. This indicates that transitive regulations play an important role in GRN, GO can be used effectively to detect transitive regulations, and their detection can improve the accuracy of the generated GRN.

After the study, we speculate that the errors in predicting networks may come from: (i) the incompleteness of the target networks; (ii) the incompleteness of the GO database; (iii) the erroneous annotations of GO database; and (iv) a limited set of attributes that are allowed by GO to annotate genes and their products. (The scope is limited to the three domains, which is not comprehensive.) Therefore, the GRNs we extracted from GO were in fact based upon partial evidence provided by the current GO database. The false negatives of the networks could be reduced by fusing the GO GRN with additional data sources such as wet-lab data. One extension to the proposed method is to identify ontology terms that are specific to the pathways under consideration, e.g., terms related to cell-cycle functions in our experiments. The GRN developed by our method is useful for validation of networks built by other experimental or computational approaches.

Chapter 4

Fusion of Multiple Data Sources in Building Time-delayed Gene Regulatory Networks

4.1 Background and Motivation

Gene interactions are essentially time-delayed since they are results of translation, folding, nuclear translocation, turnover of the regulatory protein, and elongation of the target gene mRNA [115], Gene expression (GE) time-series data provide temporal information regarding the dynamic activity of gene interactions. They can be used to (i) identify the complete set of activated genes in a biological process; (ii) to infer their rates of change, their order and their causal effects; (iii) and to model dynamic systems in the cell [8].

Various regression models have been used to estimate time-varying GRNs from GE time-series data. Examples include dynamic vector auto-regressive model [48], statistical Auto Regressive Time Varying (ARTIVA) model [81], multivariate auto-regression (MVAR) framework with a penalized regression technique, least absolute shrinkage and selection operator (LASSO) [50], and ordinary differential equations (ODE) [76]. Furthermore, to construct time-delayed GRNs, Huang et al. [68] created an R package named as GeneReg. As a time delay linear regression model, it resembles to Kim's ODE [76]. We chose GeneReg as a comparison method for our approach.

Bayesian networks (BN) model gene regulation as the statistical dependence between gene expressions. They provide a data-driven way to model GRNs. According to the comparison of various GRN inference methods in [58], BN outperformed all competitors.

We thus focus on BN-based methods to build GRNs. Dynamic BNs (DBNs) are able to capture the dynamic temporal gene regulations from time series data by incorporating the temporal order of gene expression [55, 11, 114, 54]. The first-order DBN models the gene expression at current time point by assuming that it depends on parental genes' expression at the previous time point. As the time required for the regulator gene to express its protein product and the transcription of the target gene to be affected may be lagged, one needs to take time delays into consideration to model GRN more accurately. Higher-order DBNs (HDBN) are therefore needed to capture the time-delayed interactions [161]. Furthermore, as time delays may be various for different genes, Rajapakse and Chaturvedi [114] proposed variable-order DBN (VDBN) to model variable time-delays in GRNs. We adopted the VDBN to learn the time-delayed GRN from time series GE data.

As learning BN structure is NP-hard [35], most methods for learning DBN employed local search approach like hill climbing, or meta stochastic global optimization algorithms such as genetic algorithm, simulated annealing, and Markov Chain Monte Carlo (MCMC) simulation. Dojer et al. [43] showed that for certain DBN models, where the scoring function satisfies additivity, splitting, and uniformity, such as Minimum Description Length (MDL) [80] and Bayesian-Dirichlet equivalent (BDe) [40] scoring metrics, the globally optimal DBN can be efficiently learned in polynomial time. Vinh et al. [143, 144] showed that the mutual information test (MIT) [41] combined the goodness of both BDe and MDL: the network quality and speed. We thus used the polynomial time algorithm, *GlobalMIT*⁺ [144], to learn the globally optimal VDBN structure with MIT.

DBNs created from mRNA profiles can represent the dependencies among genes, but may fail to correctly capture gene regulations, such as the dependencies between regulator genes and regulatee genes [84]. Using some priors to constrain the structures of BNs can help to capture these dependencies. In addition, constraint-based structure learning can reduce the search space [133]. As indicators of potential regulators, the transcription factors (TFs) have been widely used to constrain the search space of GRN [15, 72, 161]. In our approach, we also included TFs as parental constraints in VDBN learning process. They not only reduce the search space, but also enhance the biological reasonability of the learned GRN.

The GE data is generally incomplete and noisy for GRN modeling [63]. It is hoped that augmenting GE data with prior information would help to increase the quantity of learned GRNs. Existing biological network databases provide various evidence for GRN. Evidences such as the gene product interactions, the protein interaction data provide structural priors for GRNs [154, 103, 72]. Tu et al. [138] showed that accounting for just the binary protein-protein interactions is not enough, and that the varied interaction probabilities between different protein pairs should be quantified and reflected. Function associated network (FAN) from STRING (Search Tool for the Retrieval of Interacting Genes/Proteins, <http://string-db.org/>) quantitatively integrated known and predicted protein interaction data. The improved functional similarity with GO enrichment provided the functional association scores of genes. By using our proposed regulatory path method from GO network, the regulatory pathways could be extracted with confidence scores. We thus used these various quantitative priors in DBNs learning.

Systematically fuse multiple data sources for GRN inference remains a key challenge in systems biology [32]. The Bayesian fusion model is a popular framework to model GRN from GE data with a structural prior. The prior is a probability distribution on networks. It encodes a *priori* biological knowledge in the form of high-level topological features or supplemental data. Protein-protein interaction knowledge was incorporated when training a BN with GE data to estimate a GRN in [154, 103, 72]. In [15, 34], the transcription factor binding location data was used as the priors for GRN learning. Werhli et al. [148] also integrated prior knowledge and/or different experimental conditions to learn GRNs. In our approach, we adopted the Bayesian fusion method to fuse GE data with the quantitative PPIN from STRING.

Our method diagram is shown in Figure 4.1. In our approach, we integrated the quantitative gene interaction scores from STRING or GO as network priors in DBNs learning from GE time series data. The parameters for weighting the priors were automatically learned from its consistency with the inferred GRN from GE data. The parental constraints of VDBN structure were set as TFs to make the inferred dependencies among genes be the regulator-regulatee interactions.

Figure 4.1: Overall Model: The dependencies among genes across different time points are modeled by VDBN; the quantitative gene interactions/regulations scores from STRING or GO are served as network priors in the fusion model; and the transcriptional factors (TFs) provide the parental constraints of genes since there are regulators.

4.2 Variable-order Dynamic Bayesian Network

4.2.1 Higher-order Dynamic Bayesian Network

Given time courses of expressions of n genes $\mathcal{X} = \{x_i(t)\}_{i=1, t=1}^{I, T}$, where $x_i(t)$ denotes the value of the expression of gene i at time t , and T denotes the total number of time points which are assumed to be equally spaced. The higher-order DBN (HDBN) assumes r -order Markov dependence of gene expressions over time:

$$p(x(t)|x(t-1), \dots, x(1)) = p(x(t)|x(t-1), \dots, x(t-r)) \quad (\text{Eq. 4.1})$$

Thus, in r -order DBN, the observation of each gene is dependent on its parent genes in previous $T - r$ time points. The r -order DBN is defined by a graph G , a family of conditional probability distributions, and their parameters $\theta = \{\vartheta_i\}$, which together specify the joint distribution over the variables $p(\mathcal{X}|G, \theta)$:

$$p(\mathcal{X}|G, \theta, r) = \prod_{i=1}^I \prod_{t=r+1}^T p(x_i(t)|\pi_i(t-r), \vartheta_i), \quad (\text{Eq. 4.2})$$

where each node $x_i(t)$ depends only on its parent nodes' previous expression $\pi_i(t-r)$. The parameter matrix θ is comprised of I vectors, where each vector ϑ_i specifies the local probability distribution.

4.2.2 Learning Higher-order DBN via *GlobalMIT*⁺

To learn the DBN structure, we adopted state-of-the-art polynomial algorithm, *GlobalMIT*⁺ [144], which used information theoretic scoring metric mutual information test (MIT) [41]. To reduce the complexity of structure learning, the highly associated genes were used to constrain the parental candidates. Suppose $\pi_i = \{\pi_{i1}, \dots, \pi_{i, s_i}\}$ be the parents set of x_i with corresponding $\{d_{i1}, \dots, d_{i, s_i}\}$ discrete states, $s_i = |\pi_i|$, then the metric mutual information test (MIT) is defined as:

$$SS_{MIT} = \sum_{i=1}^I \{2(T-r)I(x_i, \pi_i) - \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i(j)}}\} \quad (\text{Eq. 4.3})$$

where $I(x_i, \pi_i)$ is the time-delayed mutual information between x_i and its parents π_i on previous r -order time shift; $(T-r)$ are the number of effective observations, when there are multiple time series, for example, m time series with similar network dynamics, each has T_m time points, then the number of effective observations should be $\sum_m T_m - m \cdot r$, the details can be found in [144]; $\chi_{\alpha, l_{ij}}$ is the value that $p(\chi^2(l_{ij}) \leq \chi_{\alpha, l_{ij}}) = \alpha$ (the χ^2 distribution at significance level $1 - \alpha$), and $l_{i\sigma_i(j)}$ is defined as:

$$l_{i\sigma_i(j)} = \begin{cases} (d_i - 1)(d_{i\sigma_i(j)} - 1) \prod_{k=1}^{j-1} d_{i\sigma_i(k)} & j = 2, \dots, s_i \\ (d_i - 1)(d_{i\sigma_i(j)} - 1) & j = 1 \end{cases} \quad (\text{Eq. 4.4})$$

where $\sigma_i = \{\sigma_i(1), \dots, \sigma_i(s_i)\}$ is any permutation of the index set $\{1, \dots, s_i\}$ of π_i , with the number of states in decreasing order. As stated in [144], maximizing the first item

in Eq. 4.3 can be shown to be equivalent to maximizing the log-likelihood of Eq. 4.2. To avoid the over-fitting of learning BN by using the *maximum* likelihood (ML) principle, there is a need to penalize the complexity of learned network. The second term in Eq. 4.3 penalizes the addition of more variables of parents set. As shown in [144], MIT performs competitively with BIC/MDL and BDe score. In our DBN learning scheme, the r -order DBN structure was determined from $p(G|\mathcal{X}, \theta, r) \propto p(\mathcal{X}, G|\theta, r)$. When we adopted the MIT score, which is then equivalent to maximizing the log-likelihood of joint distribution $p(\mathcal{X}, G|\theta, r) = p(\mathcal{X}|G, \theta, r)p(G)$.

$$\arg \max_G SS_{MIT} = \arg \max_G (\log p(\mathcal{X}|G, \theta, r) + \log p(G)) \quad (\text{Eq. 4.5})$$

According to [43], when there is no need to examine the acyclicity of the graph, the scoring function satisfies additivity, splitting, and uniformity, the BNs can be solved in polynomial time. Additivity means that the scoring function can be decomposed into a sum over the set of random variables of local scores, depending on the values of a variable and its parents in the graph only. When there is no need to examine the of the graph, this assumption allows to compute the parents set of each variable independently. Splitting indicates the scoring function can be decomposed into two components: the accuracy of representing the distribution underlying the data by the network, represented by v ; and the complexity, represented by u . Uniformity requires the complexity component u to be a non-decreasing function in the cardinality of parent sets, i.e., the network gets more complex as more variables are added to the parent sets.

As edges direct only forward in time in DBN, thus, there is no need to examine the acyclicity. Vihn et al. [144] proved that the score in Eq. 4.5 satisfied additivity and splitting. To make score in Eq. 4.5 satisfy uniformity, it requires to satisfy variable uniformity that all variables have the same number of discrete states k . Under these five assumptions, acyclicity, additivity, splitting, variable uniformity and uniformity, Theorem 1 in [144] proved that the Algorithm *GlobalMIT*⁺ applied to each variable could find a globally optimal r -th order DBN under the MIT score. Theorem 2 in [144] showed that *GlobalMIT*⁺ admits a polynomial worst-case time complexity of $O((Ir)^{\log_k(T-r)})$ in the number of variables I and DBN order r .

4.2.3 Learning Variable-order Dynamic Bayesian Network

A variable-order DBN (VDBN) allows each gene to have variable delayed parents where the optimal delay is learned via the variable-order MCMC scheme in [114]. Let $p(r)$ denote the probability of the topology having the order r . The proposal probability of a new order r' from current order r is $Q(r'|r) = \min\{1, \frac{p(r')}{p(r)}\}$. Three types of changes to the order are considered: increase a order with probability $\gamma_r = Q(r+1|r)$, decrease a order with probability $\zeta_r = Q(r-1|r)$, and no change of order with probability η_r so that $\gamma_r + \zeta_r + \eta_r = 1$.

According to [114], the probability of selecting r -ordered subset from T -ordered points is $\frac{r!}{T^r}$. The acceptance mechanism is then $A(r'|r) = \min\{1, \frac{p(G|\mathcal{X}, \theta, r', \beta)p(r')Q(r|r')}{p(G|\mathcal{X}, \theta, r, \beta)p(r)Q(r'|r)}\}$, the details are as follows.

$$A(r'|r) = \begin{cases} \min\{1, \frac{p(G|\mathcal{X}, \theta, r', \beta)}{p(G|\mathcal{X}, \theta, r, \beta)} \cdot \frac{T}{r+1}\} & \text{if } r' = r + 1 \\ \min\{1, \frac{p(G|\mathcal{X}, \theta, r', \beta)}{p(G|\mathcal{X}, \theta, r, \beta)} \cdot \frac{r}{T}\} & \text{if } r' = r - 1 \\ \min\{1, \frac{p(G|\mathcal{X}, \theta, r', \beta)}{p(G|\mathcal{X}, \theta, r, \beta)}\} & \text{if } r' = r \end{cases} \quad (\text{Eq. 4.6})$$

4.3 Fusion of Priors with Higher-order DBN

To fuse the priors into VDBN learning, we adopted the fusion method described in [132, 148] to quantitatively account prior knowledge within Bayesian framework. Suppose biological prior knowledge matrix is B , inferred GRN is given by graph G , the energy $E(G) = \sum_{i,j} |B_{ij} - G_{ij}|$. The prior distribution of networks can be written as

$$p(G|\beta) = \frac{\exp^{-\beta E(G)}}{Z(\beta)} \quad (\text{Eq. 4.7})$$

where $Z(\beta) = \sum_G \exp^{-\beta E(G)}$.

As discussed in [148], the hyper-parameter β can be interpreted as a factor that indicates the strength of the influences of the biological prior knowledge relative to the data. For $\beta \rightarrow 0$, the prior distribution becomes flat and informative about the network structure, conversely, $\beta \rightarrow \infty$. That is, the prior distribution becomes sharply peaked at the network structure with the lowest energy.

Given data $\mathcal{X}$, with integrated functional association score B in FAN, and fixed parameter Q , the DBN structural posterior probability is written as follows:

$$p(G|\mathcal{X}, \theta, r, \beta) \propto p(\mathcal{X}, G|\theta, r, \beta) = p(\mathcal{X}, G|\theta, r)p(G|\beta) \quad (\text{Eq. 4.8})$$

4.3.1 Structure Learning of r -th order DBN with Priori B

The ML structure is to maximize $\log p(\mathcal{X}, G|\theta, r) + \log p(G|\beta)$. According to Eq. 4.5, $\log p(\mathcal{X}, G|\theta, r) \approx SS_{MIT}$. Thus, the ML structure is

$$\arg \max_G p(G|\mathcal{X}, \theta, r, \beta) = \arg \max_G (SS_{MIT} - \beta \sum_{i,j} |B_{ij} - G_{ij}|) \quad (\text{Eq. 4.9})$$

Denote the MIT score with priors by SS'_{MIT} , it is given as follows.

$$SS'_{MIT} = SS_{MIT} - \beta \sum_{i,j} |B_{ij} - G_{ij}| = \sum_{i=1}^I \{2(T-r)I(x_i, \pi_i) - \sum_{j=1}^{s_i} (\chi_{\alpha, l_{i\sigma_i(j)}} + \beta |B_{ij} - G_{ij}|)\} \quad (\text{Eq. 4.10})$$

As score SS_{MIT} satisfies the five assumptions as shown in Section 4.2.2, it can be proved that the score SS'_{MIT} in Eq. 4.10 also satisfies the five assumptions for a given β .

We illustrated the five assumptions as follows.

Assumption 1. (acyclicity) Examination of the graph acyclicity is not required.

Assumption 2. (additivity) The score can be decomposed as $S(G, \mathcal{X}) = \sum_{i=1}^I s(x_i, \pi_i)$.

Assumption 3. (splitting) $s(\pi_i) = u(\pi_i) + v(\pi_i)$ for some non-negative functions u and v , and satisfy $\pi_i \subseteq \pi'_i \Rightarrow u(\pi_i) \leq u(\pi'_i)$.

Assumption 4. (variable uniformity) All variables in $\mathcal{X}$ have same number of discrete states k .

Assumption 5. (uniformity) $|\pi_i| = |\pi'_i| \Rightarrow u(\pi_i) = u(\pi'_i)$.

It is trivial that SS'_{MIT} satisfies acyclicity and additivity.

To satisfy splitting, we decomposed a variant of the score Eq. 4.10 into two components u and v . Let $S_{MIT} = \sum_{i=1}^I I(x_i, \mathcal{X}) - SS'_{MIT}$, as $\sum_{i=1}^I I(x_i, \mathcal{X})$ is a constant, maximizing SS'_{MIT} is equivalent to minimizing S_{MIT} .

$$\begin{aligned} S_{MIT}(\pi_i) &= u(\pi_i) + v(\pi_i) \\ u(\pi_i) &= \sum_{j=1}^{s_i} (\chi_{\alpha, l_{i\sigma_i(j)}} + \beta |B_{ij} - G_{ij}|) \\ v(\pi_i) &= 2(T-r)I(x_i, \mathcal{X}) - 2(T-r)I(x_i, \pi_i) \end{aligned} \quad (\text{Eq. 4.11})$$

It can be proved that u and v are non-negative functions, and satisfy $\pi_i \subseteq \pi'_i \Rightarrow u(\pi_i) \leq u(\pi'_i)$.

Proof: Since all possible parent sets π_i , the full set $\mathcal{X}$ has maximum mutual information with X_i , thus, $v \geq 0$. Since the support of the Chi-square distribution is $\mathbb{R}^+$, i.e., $\chi_{\alpha,\cdot} \geq 0$, for a given π_i and β , the prior penalty part $\sum_{j=1}^{s_i} \beta |B_{ij} - G_{ij}|$ is a positive constant, $\pi_i \subseteq \pi'_i \Rightarrow u(\pi_i) \leq u(\pi'_i)$.

Under the assumption of variable uniformity, u satisfies uniformity.

Proof: If $|\pi_i| = |\pi'_i| = s_i$, then $u(\pi_i) = u(\pi'_i) = \sum_{j=1}^{s_i} (\chi_{\alpha,(k-1)^{2kj-1}} + \beta |B_{ij} - G_{ij}|)$.

We thus modified Algorithm *GlobalMIT*⁺ in [144] as following Algorithm 3 to combine priors.

Algorithm 3 *GlobalMIT*⁺ with priors

Step 0. $\pi_i = \emptyset$

Step 1. For $p = 1$ to Ir

If $u(p) \geq S_{MIT}(\pi_i)$ then return π_i ; Stop.

$P = \arg \min \{S_{MIT}(Y) | Y \in \mathcal{X}^r; |Y| = p\}$

If $S_{MIT}(p) < S_{MIT}(\pi_i)$ then $\pi_i := P$.

Similar to Theorem 1 in [144], we can prove that using Algorithm 3 on each variable, one can also find a globally optimal r -th order DBN with priors under the score in Eq. 4.10.

Proof: The key points is that once a parent set grows to a certain extent, its complexity alone surpasses the total source of a previously found sub-optimal parent set. In fact, all the remaining potential parent sets P omitted by the algorithm has a total score higher than the current best score, i.e., $S_{MIT}(\pi) \geq u(|\pi|) \geq S_{MIT}(\pi_i)$, where π_i is the last sub-optimal parent set found.

Similar to Theorem 2 in [144], Algorithm 3 admits a polynomial worst-case time complexity in the number of variables I and DBN order r .

Proof: To find a number p^* satisfying $u(p^*) \geq S_{MIT}(\emptyset)$, clearly, there is no need to examine any parent set of cardinality p^* and over. In the worst case, the algorithm will have to examine all the possible parent sets of cardinality from 1 to $p^* - 1$. From $u(p^*) \geq S_{MIT}(\emptyset)$, we have

$$\sum_{j=1}^{p^*} \chi_{\alpha, l_{i\sigma_i(j)}} \geq v(\Phi) = 2(T - r)I(x_i, \mathcal{X}) \quad (\text{Eq. 4.12})$$

Since a fundamental property of the mutual information states that $I(U, V) \leq \min\{H(U), H(V)\}$, we have

$$I(x_i, \mathcal{X}) \leq H(x_i) \quad (\text{Eq. 4.13})$$

where $H(x_i)$ is the entropy of x_i estimated from a r -time-unit shifted expression profile, $\{x_{i,r+1}, \dots, x_{i,T}\}$. Otherwise, we can use a universally fixed upper bound for all $H(x_i)$, that is, $\log k$, then

$$I(x_i, \mathcal{X}) \leq \log k \quad (\text{Eq. 4.14})$$

Thus, we have

$$p^* = \arg \min \{p \mid \sum_{j=1}^p \chi_{\alpha, l_{i\sigma_i(j)}} \geq 2(T-r)H(x_i)\} \quad (\text{Eq. 4.15})$$

or

$$p^* = \arg \min \{p \mid \sum_{j=1}^p \chi_{\alpha, l_{i\sigma_i(j)}} \geq 2(T-r) \log k\} \quad (\text{Eq. 4.16})$$

It can be seen that p^* only depends on α , k , and $(T-r)$. Since there are $O(Ir)^{p^*}$ subsets of $\mathcal{X}^r$ with at most p^* parents, Algorithm 3 admits a polynomial worst-case time complexity in the number of variables I and DBN order r . While p^* does not admit a closed-form solution (since $\chi_{\alpha, l_{ij}}$ cannot be analytically calculated), a large over-estimate of p^* can be provided as follows. Note that $\chi_{\alpha, l_{ij}}$ is the value such that $p(\chi^2(l_{ij}) \leq \chi_{\alpha, l_{ij}}) = \alpha$. Since generally $\alpha \gg 0.5$, if we take the mean value (corresponding roughly to $\alpha = 0.5$) of the $\chi^2(l_{ij})$ distribution, i.e., l_{ij} , as an under-estimate for $\chi_{\alpha, l_{ij}}$, then, from $\sum_{j=1}^{p^*} \chi_{\alpha, l_{i\sigma_i(j)}} \geq 2(T-r) \log k$, we have

$$\sum_{j=0}^{p^*-1} (k-1)^2 k^j \geq 2(T-r) \log k \quad (\text{Eq. 4.17})$$

thus,

$$(k-1)(k^{p^*} - 1) \geq 2(T-r) \log k \quad (\text{Eq. 4.18})$$

and

$$p^* \geq \log_k \left(\frac{2(T-r) \log k}{k-1} + 1 \right) \quad (\text{Eq. 4.19})$$

Assuming $T-r \gg \log k$, we can see that $p^* \sim \log_k(T-r)$, and the algorithm admits an overall complexity of $O((Ir)^{\log_k(T-r)})$.

4.3.2 Learning Fusion Parameter β

Werhli et al. [148] suggested a MCMC scheme to learn the parameter β for the integrating of prior knowledge: suppose the prior distribution of hyper-parameters $p(\beta)$ is uniform over a positive interval $[0, \beta_{\max}]$, the proposal distribution over a moving interval of length $2l \ll \beta_{\max}$, centered on the current value of the hyper-parameter β , and the proposed hyper-parameter β' is sampled uniform over the interval $[\beta - l, \beta + l]$ with the constraint $\beta' \in [0, \beta_{\max}]$. The acceptance probability is

$$A(\beta'|\beta) = \min\left\{1, \frac{p(G|\mathcal{X}, \theta, r, \beta')p(\beta')Q(\beta|\beta')}{p(G|\mathcal{X}, \theta, r, \beta)p(\beta)Q(\beta'|\beta)}\right\} = \min\left\{1, \frac{p(G|\mathcal{X}, \theta, r, \beta')}{p(G|\mathcal{X}, \theta, r, \beta)}\right\}$$

4.4 Various Priors from Multiple Data Sources

4.4.1 Functional Association Network from STRING

The function association network (FAN) available in the databases can be used as priors. We obtained FANs from STRING (Search Tool for the Retrieval of Interacting Genes/Proteins, <http://string-db.org/>), a database of known and predicted protein interactions, which currently covers 5,214,234 proteins from 1,133 organisms. It included direct (physical) and indirect (functional) associations derived from four sources: genomic context, high-throughput experiments, (conserved) co-expressions, and previous knowledge from scientific literature, e.g., PubMed papers. It was shown that a positive correlation existed between the genomic associations and the likelihood and strength of interactions [38, 70]. Thus, multiple scores can be combined to form a single score that expresses a higher confidence. STRING [97] quantitatively integrated the scores by multiplying the probabilities of associations not predicting a functional interaction. It showed that combinations of separate scores lead to a higher coverage at a given accuracy. We chose the combined scores from STRING database as prior interaction scores for fusion DBNs modeling.

4.4.2 Gene Association Network from Gene Ontology

In Chapter 3, we proposed a functional similarity score with GO enrichment, FS-GOE, in Eq. 3.14, to extract gene association networks, which showed experimentally outperformed STRING score and previous functional similarity score on recovering functional

association networks. We used the association networks predicted from FS-GOE score as priors in fusion DBNs modeling.

4.4.3 Regulatory Paths from Gene Ontology Network

In Chapter 3, we proposed a novel method, RP-GON Algorithm 2, to detect regulatory paths within functionally associated pathways to generate GRN from gene ontology network (GON). The experimental results showed that our RP-GON method could predicted better GRNs than three state-of-the-art GRN inference methods from GE time-series data. We also used the GRNs predicted from RP-GON method as priors in fusion DBNs modeling.

4.5 Experiments and Results

4.5.1 Synthetic Datasets

To generate synthetic datasets, we employed a commonly used network described in [69]. To generate a delayed network, we added delays that are consistent with the transitive regulations in this network. The structures of these two networks were shown in Figure 4.2 and Figure 4.3.

Then, the time series of discrete observations were generated for each network with time points 20 and 50. The synthetic dataset was generated by initializing the expression data of two regulators (ACE2, CLN2) as random values from a uniform distribution on the interval $[-1, 1]$, then generate the expression values of regulated genes in successive time points according to the regulatory structure (activation, inhibition or combination function). The details can be found in [69]. Let Syn1 and Syn2 denote the synthetic dataset with 20 and 50 time points generated from network as shown in Figure 4.2, respectively; Syn3 and Syn4 denote the synthetic dataset with 20 and 50 time points generated from the delayed network as shown in Figure 4.3.

As the genes in these two networks are same and from yeast, we obtained the prior information about protein-protein interactions for this network from STRING database [97]. The gene association network was also obtained from the FS-GOE score

computed by Eq. 3.14 in Chapter 3. Unfortunately, the RP-GON method failed to predict GRN on this network because no GO “regulate” relation existed among their GO terms.

Figure 4.2: The regulatory pathways of synthetic network.

Figure 4.3: The time-delayed regulatory pathways of synthetic network, where the numbers on the edges represent the delays.

4.5.2 Real Datasets

We used the same benchmark networks described in Chapter 3: the three FANs (FAN1, FAN2 and FAN3) extracted by using Algorithm 1, and three randomly chose networks (Net1, Net2 and Net3) from the complete *E.coli* network in RegulonDB database [122]; two randomly selected networks (Net4 and Net5) for yeast from YEASTRACT database [135],

and their functional associated pathways (FAP4 and FAP5) which were extracted by using hierarchical clustering method based on the functional similarity score with GO enrichment, FS-GOE score.

Time-series GE data were collected as previously described in Section 3.6.3 of Chapter 3: the GE data for *E.coli* was obtained from “GSE6425” in NCBI GEO database, which contained two replicates with each of 11 observations taken at various intervals with 30-minute, 45-minute, 1 hour and 45 minutes, 4-hour, 6-hour and 8-hour; and the GE data for yeast cell cycle was obtained from [107], which provided results of two independent experiments for both wild-type and cyclin-mutant cells, with each of 15 observations taken at 16-minute interval.

The prior information about protein-protein interactions for both *E.coli* and yeast were obtained from STRING database [97] that provides functional association scores among genes/proteins. Transcription factors for *E.coli* and yeast were obtained from RegulonDB database [122] and YEASTRACT database [135], respectively. The gene association networks were obtained from the FS-GOE score computed by Eq. 3.14 in Chapter 3. And the prior GRNs for all these networks were predicted from GON by using RP-GON Algorithm 2 in Chapter 3. The details of these three types of priors compared with benchmark networks were shown in Chapter 3. Experimental results in Chapter 3 showed that STRING score and FS-GOE score could provide higher recall/coverage to predict benchmark networks, but with relatively low precision. They should provide more compensation information of GE data to infer GRNs. The RP-GON method generally predicted benchmark networks with higher precision, while with comparable recall on FANs and lower recall on randomly chosen networks. The high precision from this prior would help to improve the precision and accuracy of GRN inference from GE data.

4.5.3 Performance Evaluation

To evaluate the performance of various methods to recover the networks, the performance metrics including Precision (Pre.), Recall (Rec.), F1-score (F1), Accuracy (Acc.) and AUC were computed by Eq. 3.34 in Chapter 3. As ground truth network has no information about exact time delays, TP and FP were computed by counting all predicted edges of any time delay.

GeneReg [68], a state-of-the-art GRN inference method from time series GE data, used B-spline to generate predicted time points based on GE data, then predicted various delays in GRNs by using linear regression models. Note that GeneReg also used TFs to constrain the regulators. We included its performance on all benchmark networks as baselines for various DBNs.

Constrained LASSO [102], a fusion method to derive time-delayed GRN from time series gene expression data in two-step procedures: determine the probable time lags of regulations from time-lagged correlation first, and then use LASSO regression to derive the confined regulations. We also compared it with our methods on all benchmark networks.

Since there were multiple time-series for the dynamics (two time series for *E.coli* networks; two time series for yeast wild-type cells; two time series for yeast cyclin-mutant cells), all DBNs were learned by combining the multiple time series in *GlobalMIT*⁺ for each dynamic structure [144] with TFs as parental constraints. The significance level parameter α were all set as 0.95. The *maximum* delay were set as 6, the VDBN iterations were set as 2000 to learn the optimal delays, and the inner interactions for learning priors parameter β were set as 200 for each gene node. As suggested in [148], to learn optimal β , the interval l was set as 3 and $\beta_{\max}$ was set as 30. In HDBN, the fixed order was experimentally optimally set by assessing all HDBNs with various orders.

4.5.4 Performance without Priors

The comparison results of GeneReg, LASSO, and various DBNs (DBN, HDBN, VDBN) on predicting GRNs solely from GE data for synthetic, *E.coli*, and yeast networks were shown in Tables 4.1, 4.2 and 4.3, respectively. Table 4.1 showed that GRN inference methods performed better on synthetic networks with 50 time points (Syn2 and Syn4) than 20 time points (Syn1 and Syn3), which was reasonable as more training data (time points) gave more accurate predictions. It also showed that with same time points, VDBN performed slightly better on delayed networks (Syn3 and Syn4) than networks without delays (Syn1 and Syn2) solely from GE data.

Note that all DBNs performed poorly on three *E.coli* networks, Net1, Net2 and FAN1, where no true positive edges were predicted. This may come from two reasons: one is the

lack of target networks which makes the predicted edges not be validated; the other one may be from the lack of GE time-series data where some regulated expression has not been captured. It was shown that on all the other networks, VDBN significantly improved the overall performance of DBN, with F1-score improvement ranging from 7% to 46%. Note that LASSO method [102] fused time-lagged correlation with LASSO regression to infer time-delayed GRN from GE data, it gave better performance than other GRN individual inference methods on Syn3 and *E.coli* FAN2. On the other networks, VDBN outperformed all other methods to predict GRN solely from GE data with TFs as parental constraints. This demonstrates the biological feasibility of modeling time-delays in GRN and the assumption that various delays exist in gene regulations.

Table 4.1: Performance of various GRN inference methods solely from GE data on synthetic networks. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network)

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Syn1	GeneReg	1.0000	0.1818	0.3077	0.9375	0.5909
	LASSO	0.2353	0.3636	0.2857	0.8611	0.6729
	DBN	0.6000	0.2727	0.3750	0.9306	0.6309
	HDBN	0.6667	0.1818	0.2857	0.9306	0.5878
	VDBN	0.3636	0.3636	0.3636	0.9028	0.6599
Syn2	GeneReg	1.0000	0.0909	0.1667	0.9306	0.5455
	LASSO	0.5000	0.4545	0.4762	0.9236	0.7245
	DBN	0.3158	0.5455	0.4000	0.8750	0.7334
	HDBN	0.5000	0.1818	0.2667	0.9236	0.5820
	VDBN	0.2759	0.7273	0.4000	0.8333	0.7878
Syn3	GeneReg	1.0000	0.0556	0.1053	0.8819	0.5278
	LASSO	0.3750	0.3333	0.3529	0.8472	0.6623
	DBN	0.5000	0.1111	0.1818	0.8750	0.5472
	HDBN	0.6667	0.1111	0.1905	0.8819	0.5511
	VDBN	0.5000	0.2222	0.3077 ($\uparrow$ 0.12)	0.8750	0.5979
Syn4	GeneReg	-	-	-	-	-
	LASSO	0.7143	0.2778	0.4000	0.8958	0.6389
	DBN	0.5000	0.1667	0.2500	0.8750	0.5712
	HDBN	0.5714	0.2222	0.3200	0.8819	0.6023
	VDBN	0.5000	0.4444	0.4706 ($\uparrow$ 0.22)	0.8750	0.6892

Table 4.2: Performance of various GRN inference methods solely from GE data on *E.coli* randomly chosen networks and FANs. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network)

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net1	GeneReg	0	0	-	0.9082	0.4944
	LASSO	0	0	-	0.8316	0.4944
	DBN	-	0	-	0.9184	0.5000
	HDBN	-	0	-	0.9184	0.5000
	VDBN	-	0	-	0.9184	0.5000
FAN1	GeneReg	0.1364	0.1667	0.1500	0.8265	0.5807
	LASSO	0	0	-	0.9031	0.5000
	DBN	-	0	-	0.9082	0.5000
	HDBN	-	0	-	0.9082	0.5000
	VDBN	-	0	-	0.9082	0.5000
Net2	GeneReg	0.4000	0.0833	0.1379	0.8889	0.5390
	LASSO	0.0667	0.0417	0.0513	0.8356	0.5157
	DBN	0.2500	0.0417	0.0714	0.8844	0.5131
	HDBN	-	0	-	0.8933	0.5000
	VDBN	0.2500	0.0417	0.0714	0.8844	0.5131
FAN2	GeneReg	0.1111	0.0667	0.0833	0.9022	0.5308
	LASSO	0.1765	0.4000	0.2449	0.8356	0.6878
	DBN	0.3333	0.0667	0.1111	0.9289	0.5311
	HDBN	0.3333	0.0667	0.1111	0.9289	0.5308
	VDBN	0.3333	0.1333	0.1905 ($\uparrow$ 0.08)	0.9244	0.5589
Net3	GeneReg	0.4000	0.0833	0.1379	0.8889	0.5336
	LASSO	0	0	-	0.7867	0.4925
	DBN	0.1667	0.0417	0.0667	0.8756	0.5081
	HDBN	0.4000	0.0833	0.1379	0.8889	0.5340
	VDBN	0.2308	0.1250	0.1622 ($\uparrow$ 0.10)	0.8622	0.5364
FAN3	GeneReg	0	0	-	0.7733	0.5000
	LASSO	0	0	-	0.8844	0.5000
	DBN	0.2500	0.2143	0.2308	0.9111	0.6071
	HDBN	0.1538	0.1429	0.1481	0.8978	0.5714
	VDBN	0.8889	0.5714	0.6957 ($\uparrow$ 0.46)	0.9689	0.7857

Table 4.3: Performance of various GRN inference methods solely from GE data on yeast networks and FAPs. (The parentheses of F1 in VDBN show the improvement of F1-score from DBN on the same network; and the parentheses of Pre. in FAP4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net4 Wild-type	GeneReg	0.1687(0.3012)	0.2414	0.1986	0.8038	0.5754
	LASSO	0.1667(0.2407)	0.1552	0.1607	0.8368	0.5503
	DBN	0.4783(0.8696)	0.1897	0.2716	0.8976	0.5834
	HDBN	0.5500(0.9500)	0.1897	0.2821	0.9028	0.5871
	VDBN	0.4773(0.8636)	0.3621	0.4118 ($\uparrow$ 0.14)	0.8958	0.6626
FAP4 Wild-type	GeneReg	0.4286(0.8571)	0.2727	0.3333	0.6667	0.5855
	LASSO	0.2000(0.4000)	0.0909	0.1250	0.6111	0.4800
	DBN	1.0000(1.0000)	0.1818	0.3077	0.7500	0.5909
	HDBN	0.5000(1.0000)	0.0909	0.1538	0.6944	0.5236
	VDBN	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5673
Net4 Mutant	GeneReg	0.1322(0.2810)	0.2759	0.1788	0.7448	0.5740
	LASSO	0.0968(0.2553)	0.1034	0.1000	0.8125	0.5227
	DBN	0.5000(0.8125)	0.1379	0.2162	0.8993	0.5623
	HDBN	0.6429(0.8571)	0.1552	0.2500	0.9063	0.5727
	VDBN	0.4412(0.8235)	0.2586	0.3261 ($\uparrow$ 0.11)	0.8924	0.6123
FAP4 Mutant	GeneReg	0.5000(0.7500)	0.3636	0.4211	0.6944	0.6218
	LASSO	0.2500(0.5000)	0.1818	0.2105	0.5833	0.5145
	DBN	1.0000(1.0000)	0.2727	0.4286	0.7778	0.6364
	HDBN	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5709
	VDBN	0.8000(1.0000)	0.3636	0.5000 ($\uparrow$ 0.07)	0.7778	0.6655
Net5 Wild-type	GeneReg	0.2923	0.2484	0.2686	0.6688	0.5995
	LASSO	0.3111	0.1830	0.2305	0.7008	0.5793
	DBN	0.8095	0.1111	0.1954	0.7760	0.5515
	HDBN	0.5714	0.0523	0.0958	0.7584	0.5198
	VDBN	0.7241	0.2745	0.3981 ($\uparrow$ 0.20)	0.7968	0.6207
FAP5 Wild-type	GeneReg	0.7000	0.1667	0.2692	0.4063	0.5168
	LASSO	0.5500	0.2619	0.3548	0.3750	0.4957
	DBN	0.7143	0.1190	0.2041	0.3906	0.5108
	HDBN	0.8889	0.1905	0.3137	0.4531	0.5725
	VDBN	0.7586	0.5238	0.6197 ($\uparrow$ 0.41)	0.5781	0.6169
Net5 Mutant	GeneReg	0.2969	0.2484	0.2705	0.6720	0.6032
	LASSO	0.2727	0.1569	0.1992	0.6912	0.5700
	DBN	0.8571	0.1176	0.2069	0.7792	0.5567
	HDBN	0.7500	0.0588	0.1091	0.7648	0.5273
	VDBN	0.7660	0.2353	0.3600 ($\uparrow$ 0.15)	0.7952	0.6075
FAP5 Mutant	GeneReg	0.5217	0.2857	0.3692	0.3594	0.5011
	LASSO	0.6667	0.2381	0.3509	0.4219	0.5249
	DBN	1.0000	0.1667	0.2857	0.4531	0.5833
	HDBN	0.8000	0.0952	0.1702	0.3906	0.5271
	VDBN	0.8421	0.3810	0.5246 ($\uparrow$ 0.24)	0.5469	0.6320

4.5.5 Performance with Various Priors

Marbach et al. [95] showed that integration of predictions from multiple inference methods would give robust and high performance over 30 network inference methods in DREAM project. To compare our fusion method, we employed the meta-fusion method that was commonly used to mine consensus information from multiple data sources, and integrate predictions from various GRN inference methods. Given two networks, the meta-fusion method conserved the edges that were both in the two networks and eliminated other inconsistent edges. In the experiments, the meta-fusion networks were generated from various inferred GRNs and prior networks by using meta-fusion method.

Table 4.4 and 4.5 showed the performance on synthetic networks. It was seen that on all the four synthetic data sets, the fusion methods performed better than meta-fusion methods and single GRN inference methods. The higher AUC with about 0.82 was achieved by fusion methods showed the power of our fusion methods. As seen from these two tables, with same time points, all fusion DBNs methods on delayed networks (Syn3 and Syn4) improved much performance on the networks without delays (Syn1 and Syn2). This reflects the power of DBNs on capturing the time-delayed gene regulation. Compared with the slight improvement solely from GE data in Table 4.1, the fusion methods outperformed single GRN inference methods. This indicated that the fusion methods captured more biologically reasonable information from priors, and improved the time-delayed GRNs inference from GE data.

Tables 4.6, 4.7 and 4.8 showed the comparison performance of DBNs with various prior scores from STRING, our FS-GOE score, and regulatory pathways from RP-GON method on the *E.coli* randomly selected networks (Net1, Net2, Net3) and FANs (FAN1, FAN2, FAN3), respectively. Table 4.9, 4.10, 4.11 and 4.12 showed the comparison results of DBNs with these various priors on randomly chosen yeast cell cycle networks (Net4 and Net5) and there FAPs (FAP4 and FAP5) by using wild-type or mutant GE time-series data. As seen from these tables, our fusion methods significantly outperformed all meta-fusion methods on all networks. This reflects the quantitative fusion framework is able to utilize and capture more consistent information during the GRN learning process.

Table 4.4: Performance of various DBNs fused with various priors on synthetic network.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Syn1:meta-fusion	GeneReg and STRING	1.0000	0.0909	0.1667	0.9306	0.5455
	LASSO and STRING	0.2727	0.2727	0.2727	0.8889	0.6323
	DBN and STRING	0.5000	0.1818	0.2667	0.9236	0.5848
	HDBN and STRING	0.5000	0.0909	0.1538	0.9236	0.5420
	VDBN and STRING	0.3333	0.1818	0.2353	0.9097	0.5772
	GeneReg and FS-GOE	1.0000	0.1818	0.3077	0.9375	0.5909
	LASSO and FS-GOE	0.2353	0.3636	0.2857	0.8611	0.6729
	DBN and FS-GOE	0.6000	0.2727	0.3750	0.9306	0.6309
	HDBN and FS-GOE	0.6667	0.1818	0.2857	0.9306	0.5878
	VDBN and FS-GOE	0.3636	0.3636	0.3636	0.9028	0.6599
Syn1:fusion	DBN with STRING	0.1429	0.3636	0.2051	0.7847	0.6141
	HDBN with STRING	0.1786	0.4545	0.2564	0.7986	0.6623
	VDBN with STRING	0.1250	0.2727	0.1714	0.7986	0.5742
	DBN with FS-GOE	0.3226	0.9091	0.4762	0.8472	0.8671
	HDBN with FS-GOE	0.2759	0.7273	0.4000	0.8333	0.7929
	VDBN with FS-GOE	0.2286	0.7273	0.3478	0.7917	0.7505
Syn2:meta-fusion	GeneReg and STRING	1.0000	0.0909	0.1667	0.9306	0.5455
	LASSO and STRING	0.5714	0.3636	0.4444	0.9306	0.6818
	DBN and STRING	0.3333	0.4545	0.3846	0.8889	0.6951
	HDBN and STRING	0.3333	0.0909	0.1429	0.9167	0.5373
	VDBN and STRING	0.3333	0.5455	0.4138	0.8819	0.7303
	GeneReg and FS-GOE	1.0000	0.0909	0.1667	0.9306	0.5455
	LASSO and FS-GOE	0.5000	0.4545	0.4762	0.9236	0.7245
	DBN and FS-GOE	0.3158	0.5455	0.4000	0.8750	0.7334
	HDBN and FS-GOE	0.5000	0.1818	0.2667	0.9236	0.5820
	VDBN and FS-GOE	0.2759	0.7273	0.4000	0.8333	0.7878
Syn2:fusion	DBN with STRING	0.2250	0.8182	0.3529	0.7708	0.8134
	HDBN with STRING	0.1714	0.5455	0.2609	0.7639	0.6692
	VDBN with STRING	0.1944	0.6364	0.2979	0.7708	0.7239
	DBN with FS-GOE	0.2308	0.8182	0.3600	0.7778	0.7915
	HDBN with FS-GOE	0.2105	0.7273	0.3265	0.7708	0.7375
	VDBN with FS-GOE	0.2326	0.9091	0.3704	0.7639	0.8271

Table 4.5: Performance of various DBNs fused with various priors on time-delayed synthetic network.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Syn3:meta-fusion	GeneReg and STRING	1.0000	0.0556	0.1053	0.8819	0.5278
	LASSO and STRING	0.5000	0.2222	0.3077	0.8750	0.6067
	DBN and STRING	0.3333	0.0556	0.0952	0.8681	0.5194
	HDBN and STRING	1.0000	0.1111	0.2000	0.8889	0.5556
	VDBN and STRING	0.6000	0.1667	0.2609	0.8819	0.5750
	GeneReg and FS-GOE	1.0000	0.0556	0.1053	0.8819	0.5278
	LASSO and FS-GOE	0.3750	0.3333	0.3529	0.8472	0.6623
	DBN and FS-GOE	0.5000	0.1111	0.1818	0.8750	0.5472
	HDBN and FS-GOE	0.6667	0.1111	0.1905	0.8819	0.5511
	VDBN and FS-GOE	0.5000	0.2222	0.3077	0.8750	0.5979
Syn3:fusion	DBN with STRING	0.4000	0.5556	0.4651	0.8403	0.7152
	HDBN with STRING	0.3913	0.5000	0.4390	0.8403	0.6920
	VDBN with STRING	0.3500	0.3889	0.3684	0.8333	0.6490
	DBN with FS-GOE	0.4138	0.6667	0.5106	0.8403	0.7509
	HDBN with FS-GOE	0.4138	0.6667	0.5106	0.8403	0.7504
	VDBN with FS-GOE	0.4231	0.6111	0.5000	0.8472	0.7451
Syn4:meta-fusion	GeneReg and STRING	-	-	-	-	-
	LASSO and STRING	0.7500	0.1667	0.2727	0.8889	0.5833
	DBN and STRING	0.6667	0.1111	0.1905	0.8819	0.5516
	HDBN and STRING	0.7500	0.1667	0.2727	0.8889	0.5796
	VDBN and STRING	0.5556	0.2778	0.3704	0.8819	0.6226
	GeneReg and FS-GOE	-	-	-	-	-
	LASSO and FS-GOE	0.7143	0.2778	0.4000	0.8958	0.6389
	DBN and FS-GOE	0.5000	0.1667	0.2500	0.8750	0.5712
	HDBN and FS-GOE	0.5714	0.2222	0.3200	0.8819	0.6023
	VDBN and FS-GOE	0.5000	0.4444	0.4706	0.8750	0.6892
Syn4:fusion	DBN with STRING	0.3548	0.6111	0.4490	0.8125	0.7244
	HDBN with STRING	0.3077	0.4444	0.3636	0.8056	0.6526
	VDBN with STRING	0.3462	0.5000	0.4091	0.8194	0.6894
	DBN with FS-GOE	0.4412	0.8333	0.5769	0.8472	0.8177
	HDBN with FS-GOE	0.3548	0.6111	0.4490	0.8125	0.7119
	VDBN with FS-GOE	0.3636	0.6667	0.4706	0.8125	0.7319

Table 4.6: Performance of various DBNs fused with various priors on *E.coli* Net1 and FAN1.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net1:meta-fusion	GeneReg and STRING	0	0	-	0.9133	0.4972
	LASSO and STRING	-	0	-	0.9184	0.5000
	DBN and STRING	-	0	-	0.9184	0.5000
	HDBN and STRING	-	0	-	0.9184	0.5000
	VDBN and STRING	-	0	-	0.9184	0.5000
	GeneReg and FS-GOE	0	0	-	0.9082	0.4944
	LASSO and FS-GOE	0	0	-	0.8316	0.4944
	DBN and FS-GOE	-	0	-	0.9184	0.5000
	HDBN and FS-GOE	-	0	-	0.9184	0.5000
	VDBN and FS-GOE	-	0	-	0.9184	0.5000
	GeneReg and RP-GON	-	0	-	0.9184	0.5000
	LASSO and RP-GON	-	0	-	0.9184	0.5000
	DBN and RP-GON	-	0	-	0.9184	0.5000
	HDBN and RP-GON	-	0	-	0.9184	0.5000
	VDBN and RP-GON	-	0	-	0.9184	0.5000
Net1:fusion	DBN with STRING	0.5000	0.3125	0.3846	0.9184	0.6455
	HDBN with STRING	0.5556	0.3125	0.4000	0.9235	0.6455
	VDBN with STRING	0.5000	0.3125	0.3846	0.9184	0.6455
	DBN with FS-GOE	0.6000	0.3750	0.4615	0.9286	0.6767
	HDBN with FS-GOE	0.5000	0.2500	0.3333	0.9184	0.6142
	VDBN with FS-GOE	0.6000	0.3750	0.4615	0.9286	0.6767
	DBN with RP-GON	0.5000	0.3125	0.3846	0.9184	0.6455
	HDBN with RP-GON	0.5556	0.3125	0.4000	0.9235	0.6455
	VDBN with RP-GON	0.5000	0.3125	0.3846	0.9184	0.6455
FAN1:meta-fusion	GeneReg and STRING	0.2500	0.1667	0.2000	0.8776	0.5833
	LASSO and STRING	0	0	-	0.9031	0.5000
	DBN and STRING	-	0	-	0.9082	0.5000
	HDBN and STRING	-	0	-	0.9082	0.5000
	VDBN and STRING	-	0	-	0.9082	0.5000
	GeneReg and FS-GOE	0.1364	0.1667	0.1500	0.8265	0.5807
	LASSO and FS-GOE	0	0	-	0.9031	0.5000
	DBN and FS-GOE	-	0	-	0.9082	0.5000
	HDBN and FS-GOE	-	0	-	0.9082	0.5000
	VDBN and FS-GOE	-	0	-	0.9082	0.5000
	GeneReg and RP-GON	1.0000	0.1111	0.2000	0.9184	0.5556
	LASSO and RP-GON	-	0	-	0.9082	0.5000
	DBN and RP-GON	-	0	-	0.9082	0.5000
	HDBN and RP-GON	-	0	-	0.9082	0.5000
	VDBN and RP-GON	-	0	-	0.9082	0.5000
FAN1:fusion	DBN with STRING	0.6154	0.4444	0.5161	0.9235	0.7082
	HDBN with STRING	0.6154	0.4444	0.5161	0.9235	0.7082
	VDBN with STRING	0.6154	0.4444	0.5161	0.9235	0.7082
	DBN with FS-GOE	0.6667	0.5556	0.6061	0.9337	0.7631
	HDBN with FS-GOE	0.6429	0.5000	0.5625	0.9286	0.7349
	VDBN with FS-GOE	0.6667	0.5556	0.6061	0.9337	0.7631
	DBN with RP-GON	0.6667	0.5556	0.6061	0.9337	0.7631
	HDBN with RP-GON	0.6667	0.5556	0.6061	0.9337	0.7631
	VDBN with RP-GON	0.6667	0.5556	0.6061	0.9337	0.7631

Table 4.7: Performance of various DBNs fused with various priors on *E.coli* Net2 and FAN2.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net2:meta-fusion	GeneReg and STRING	0.4000	0.0833	0.1379	0.8889	0.5390
	LASSO and STRING	0.3333	0.0417	0.0741	0.8889	0.5208
	DBN and STRING	0	0	-	0.8889	0.4975
	HDBN and STRING	-	0	-	0.8933	0.5000
	VDBN and STRING	0	0	-	0.8889	0.4975
	GeneReg and FS-GOE	0.4000	0.0833	0.1379	0.8889	0.5390
	LASSO and FS-GOE	0	0	-	0.8756	0.4975
	DBN and FS-GOE	0.3333	0.0417	0.0741	0.8889	0.5157
	HDBN and FS-GOE	0	0	-	0.8933	0.5000
	VDBN and FS-GOE	0.3333	0.0417	0.0741	0.8889	0.5157
	GeneReg and RP-GON	-	0	-	0.8933	0.5000
	LASSO and RP-GON	1.0000	0.0417	0.0800	0.8978	0.5208
	DBN and RP-GON	0	0	-	0.8889	0.4975
	HDBN and RP-GON	-	0	-	0.8933	0.5000
	VDBN and RP-GON	0	0	-	0.8889	0.4975
Net2:fusion	DBN with STRING	0.2000	0.1667	0.1818	0.8400	0.5512
	HDBN with STRING	0.1111	0.0833	0.0952	0.8311	0.5062
	VDBN with STRING	0.2000	0.1667	0.1818	0.8400	0.5510
	DBN with FS-GOE	0.4000	0.3333	0.3636	0.8756	0.6399
	HDBN with FS-GOE	0.2778	0.2083	0.2381	0.8578	0.5793
	VDBN with FS-GOE	0.3684	0.2917	0.3256	0.8711	0.6193
	DBN with RP-GON	0.4000	0.3333	0.3636	0.8756	0.6405
	HDBN with RP-GON	0.3684	0.2917	0.3256	0.8711	0.6201
	VDBN with RP-GON	0.3636	0.3333	0.3478	0.8667	0.6325
FAN2:meta-fusion	GeneReg and STRING	0.1111	0.0667	0.0833	0.9022	0.5308
	LASSO and STRING	0.2222	0.4000	0.2857	0.8667	0.6878
	DBN and STRING	0.5000	0.0667	0.1176	0.9333	0.5311
	HDBN and STRING	0.5000	0.0667	0.1176	0.9333	0.5308
	VDBN and STRING	0.4000	0.1333	0.2000	0.9289	0.5589
	GeneReg and FS-GOE	0.1111	0.0667	0.0833	0.9022	0.5308
	LASSO and FS-GOE	0.1923	0.3333	0.2439	0.8622	0.6565
	DBN and FS-GOE	0.3333	0.0667	0.1111	0.9289	0.5311
	HDBN and FS-GOE	0.3333	0.0667	0.1111	0.9289	0.5308
	VDBN and FS-GOE	0.3333	0.1333	0.1905	0.9244	0.5589
	GeneReg and RP-GON	-	0	-	0.9333	0.5000
	LASSO and RP-GON	-	0	-	0.9333	0.5000
	DBN and RP-GON	0.5000	0.0667	0.1176	0.9333	0.5333
	HDBN and RP-GON	0	0	-	0.9289	0.5000
	VDBN and RP-GON	0.5000	0.0667	0.1176	0.9333	0.5333
FAN2:fusion	DBN with STRING	0.2857	0.1333	0.1818	0.9200	0.5575
	HDBN with STRING	0.2857	0.1333	0.1818	0.9200	0.5568
	VDBN with STRING	0.4000	0.2667	0.3200	0.9244	0.6214
	DBN with FS-GOE	0.4545	0.3333	0.3846	0.9289	0.6537
	HDBN with FS-GOE	0.2500	0.1333	0.1739	0.9156	0.5541
	VDBN with FS-GOE	0.4545	0.3333	0.3846	0.9289	0.6537
	DBN with RP-GON	0.5455	0.4000	0.4615	0.9378	0.6889
	HDBN with RP-GON	0.4444	0.2667	0.3333	0.9289	0.6225
	VDBN with RP-GON	0.5000	0.4000	0.4444	0.9333	0.6875

Table 4.8: Performance of various DBNs fused with various priors on *E.coli* Net3 and FAN3.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net3:meta-fusion	GeneReg and STRING	0.5000	0.0833	0.1429	0.8933	0.5363
	LASSO and STRING	0	0	-	0.8578	0.4950
	DBN and STRING	-	0	-	0.8933	0.5000
	HDBN and STRING	1.0000	0.0833	0.1538	0.9022	0.5417
	VDBN and STRING	0.6667	0.0833	0.1481	0.8978	0.5392
	GeneReg and FS-GOE	0.4000	0.0833	0.1379	0.8889	0.5336
	LASSO and FS-GOE	0	0	-	0.8578	0.4925
	DBN and FS-GOE	0.2500	0.0417	0.0714	0.8844	0.5133
	HDBN and FS-GOE	0.6667	0.0833	0.1481	0.8978	0.5390
	VDBN and FS-GOE	0.4286	0.1250	0.1935	0.8889	0.5523
	GeneReg and RP-GON	0	0	-	0.8889	0.4975
	LASSO and RP-GON	-	0	-	0.8933	0.5000
	DBN and RP-GON	-	0	-	0.8933	0.5000
	HDBN and RP-GON	-	0	-	0.8933	0.5000
	VDBN and RP-GON	-	0	-	0.8933	0.5000
Net3:fusion	DBN with STRING	0.1538	0.0833	0.1081	0.8533	0.5131
	HDBN with STRING	0.2500	0.1250	0.1667	0.8667	0.5392
	VDBN with STRING	0.2000	0.1250	0.1538	0.8533	0.5318
	DBN with FS-GOE	0.2857	0.1667	0.2105	0.8667	0.5574
	HDBN with FS-GOE	0.3077	0.1667	0.2162	0.8711	0.5601
	VDBN with FS-GOE	0.3333	0.2500	0.2857	0.8667	0.5935
	DBN with RP-GON	0.3846	0.2083	0.2703	0.8800	0.5822
	HDBN with RP-GON	0.3636	0.1667	0.2286	0.8800	0.5651
	VDBN with RP-GON	0.4000	0.2500	0.3077	0.8800	0.6012
FAN3:meta-fusion	GeneReg and STRING	0	0	-	0.7733	0.5000
	LASSO and STRING	0	0	-	0.7733	0.5000
	DBN and STRING	0.2857	0.1429	0.1905	0.9244	0.5714
	HDBN and STRING	0.2500	0.1429	0.1818	0.9200	0.5714
	VDBN and STRING	1.0000	0.5000	0.6667	0.9689	0.7500
	GeneReg and FS-GOE	0	0	-	0.9200	0.5000
	LASSO and FS-GOE	0	0	-	0.9200	0.5000
	DBN and FS-GOE	0.3333	0.0714	0.1176	0.9333	0.5357
	HDBN and FS-GOE	0	0	-	0.9333	0.5000
	VDBN and FS-GOE	0.5000	0.0714	0.1250	0.9378	0.5357
	GeneReg and RP-GON	-	0	-	0.9378	0.5000
	LASSO and RP-GON	-	0	-	0.9378	0.5000
	DBN and RP-GON	-	0	-	0.9378	0.5000
	HDBN and RP-GON	1.0000	0.1429	0.2500	0.9467	0.5714
	VDBN and RP-GON	0.8000	0.2857	0.4211	0.9511	0.6429
FAN3:fusion	DBN with STRING	0.9091	0.7143	0.8000	0.9778	0.8571
	HDBN with STRING	0.9000	0.6429	0.7500	0.9733	0.8214
	VDBN with STRING	0.9091	0.7143	0.8000	0.9778	0.8571
	DBN with FS-GOE	0.9091	0.7143	0.8000	0.9778	0.8571
	HDBN with FS-GOE	0.9000	0.6429	0.7500	0.9733	0.8214
	VDBN with FS-GOE	0.9091	0.7143	0.8000	0.9778	0.8571
	DBN with RP-GON	0.9091	0.7143	0.8000	0.9778	0.8571
	HDBN with RP-GON	0.9000	0.6429	0.7500	0.9733	0.8214
	VDBN with RP-GON	0.9091	0.7143	0.8000	0.9778	0.8571

Table 4.9: Performance of various DBNs fused with various priors on yeast Net4 and FAP4 with wild-type time series. (The parentheses of Pre. in Net4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net4 Wild-type:meta-fusion	GeneReg and STRING	0.2340(0.3404)	0.1897	0.2095	0.8559	0.5758
	LASSO and STRING	0.2813(0.3750)	0.1552	0.2000	0.8750	0.5677
	DBN and STRING	0.4737(0.8421)	0.1552	0.2338	0.8976	0.5680
	HDBN and STRING	0.5000(0.9286)	0.1207	0.1944	0.8993	0.5541
	VDBN and STRING	0.5000(0.8529)	0.2931	0.3696	0.8993	0.6319
	GeneReg and FS-GOE	0.1687(0.3012)	0.2414	0.1986	0.8038	0.5754
	LASSO and FS-GOE	0.1667(0.2407)	0.1552	0.1607	0.8368	0.5503
	DBN and FS-GOE	0.4783(0.8696)	0.1897	0.2716	0.8976	0.5834
	HDBN and FS-GOE	0.5500(0.9500)	0.1897	0.2821	0.9028	0.5871
	VDBN and FS-GOE	0.4773(0.8636)	0.3621	0.4118	0.8958	0.6626
	GeneReg and RP-GON	0.2941(0.3529)	0.0862	0.1333	0.8872	0.5332
	LASSO and RP-GON	0.5455(0.7273)	0.1034	0.1739	0.9010	0.5481
	DBN and RP-GON	0.7500(1.0000)	0.0517	0.0968	0.9028	0.5249
	HDBN and RP-GON	0.7500(1.0000)	0.0517	0.0968	0.9028	0.5249
	VDBN and RP-GON	0.6154(0.9231)	0.1379	0.2254	0.9045	0.5653
Net4 Wild-type:fusion	DBN with STRING	0.4359(0.8205)	0.2931	0.3505	0.8906	0.6275
	HDBN with STRING	0.5000(0.8529)	0.2931	0.3696	0.8993	0.6275
	VDBN with STRING	0.4902(0.8627)	0.4310	0.4587	0.8976	0.6947
	DBN with FS-GOE	0.4615(0.8205)	0.3103	0.3711	0.8941	0.6369
	HDBN with FS-GOE	0.4545(0.8485)	0.2586	0.3297	0.8941	0.6141
	VDBN with FS-GOE	0.4630(0.8333)	0.4310	0.4464	0.8924	0.6927
	DBN with RP-GON	0.4063(0.8125)	0.2241	0.2889	0.8889	0.5942
	HDBN with RP-GON	0.5333(0.9000)	0.2759	0.3636	0.9028	0.6258
	VDBN with RP-GON	0.4902(0.8431)	0.4310	0.4587	0.8976	0.6948
FAP4 Wild-type:meta-fusion	GeneReg and STRING	0.7500(1.0000)	0.2727	0.4000	0.7500	0.6364
	LASSO and STRING	0.3333(0.6667)	0.0909	0.1429	0.6667	0.5018
	DBN and STRING	1.0000(1.0000)	0.1818	0.3077	0.7500	0.5909
	HDBN and STRING	0.5000(1.0000)	0.0909	0.1538	0.6944	0.5236
	VDBN and STRING	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5673
	GeneReg and FS-GOE	0.4286(0.8571)	0.2727	0.3333	0.6667	0.5855
	LASSO and FS-GOE	0.2000(0.4000)	0.0909	0.1250	0.6111	0.4800
	DBN and FS-GOE	1.0000(1.0000)	0.1818	0.3077	0.7500	0.5909
	HDBN and FS-GOE	0.5000(1.0000)	0.0909	0.1538	0.6944	0.5236
	VDBN and FS-GOE	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5673
	GeneReg and RP-GON	-	0	-	0.6944	0.4996
	LASSO and RP-GON	-	0	-	0.6944	0.4996
	DBN and RP-GON	-	0	-	0.6944	0.4996
	HDBN and RP-GON	-	0	-	0.6944	0.4996
	VDBN and RP-GON	-	0	-	0.6944	0.4996
FAP4 Wild-type:fusion	DBN with STRING	0.7500(1.0000)	0.5455	0.6316	0.8056	0.7218
	HDBN with STRING	0.7500(1.0000)	0.5455	0.6316	0.8056	0.7218
	VDBN with STRING	0.7500(1.0000)	0.5455	0.6316	0.8056	0.7218
	DBN with FS-GOE	0.7500(1.0000)	0.5455	0.6316	0.8056	0.7218
	HDBN with FS-GOE	0.6250(0.8750)	0.4545	0.5263	0.7500	0.6473
	VDBN with FS-GOE	0.7000(0.9000)	0.6364	0.6667	0.8056	0.7309
	DBN with RP-GON	0.7500(1.0000)	0.5455	0.6316	0.8056	0.7218
	HDBN with RP-GON	0.6667(0.8889)	0.5455	0.6000	0.7778	0.6909
	VDBN with RP-GON	0.7273(0.9091)	0.7273	0.7273	0.8333	0.7745

Table 4.10: Performance of various DBNs fused with various priors on yeast Net4 and FAP4 with mutant time series. (The parentheses of Pre. in FAP4 show the improved precision by considering the validated edges from YTRP database as true positive edges.)

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net4 Mutant:meta-fusion	GeneReg and STRING	0.1605(0.2963)	0.2241	0.1871	0.8038	0.5770
	LASSO and STRING	0.0816(0.2222)	0.0690	0.0748	0.8281	0.5137
	DBN and STRING	0.6667(0.8889)	0.1034	0.1791	0.9045	0.5489
	HDBN and STRING	0.7500(0.8750)	0.1034	0.1818	0.9063	0.5499
	VDBN and STRING	0.5789(0.8421)	0.1897	0.2857	0.9045	0.5875
	GeneReg and FS-GOE	0.1322(0.2810)	0.2759	0.1788	0.7448	0.5740
	LASSO and FS-GOE	0.0968(0.2553)	0.1034	0.1000	0.8125	0.5227
	DBN and FS-GOE	0.5000(0.8125)	0.1379	0.2162	0.8993	0.5623
	HDBN and FS-GOE	0.6429(0.8571)	0.1552	0.2500	0.9063	0.5727
	VDBN and FS-GOE	0.4412(0.8235)	0.2586	0.3261	0.8924	0.6123
	GeneReg and RP-GON	0.2353(0.3529)	0.0690	0.1067	0.8837	0.5220
	LASSO and RP-GON	0.4167(0.6250)	0.0862	0.1429	0.8958	0.5367
	DBN and RP-GON	0.4545(0.7273)	0.0862	0.1449	0.8976	0.5384
	HDBN and RP-GON	0.6667(0.6667)	0.0690	0.1250	0.9028	0.5326
	VDBN and RP-GON	0.4545(0.7273)	0.1724	0.2500	0.8958	0.5759
Net4 Mutant:fusion	DBN with STRING	0.5484(0.8710)	0.2931	0.3820	0.9045	0.6344
	HDBN with STRING	0.6154(0.7692)	0.2759	0.3810	0.9097	0.6288
	VDBN with STRING	0.5122(0.8049)	0.3621	0.4242	0.9010	0.6645
	DBN with FS-GOE	0.4286(0.7857)	0.2069	0.2791	0.8924	0.5888
	HDBN with FS-GOE	0.4615(0.7692)	0.2069	0.2857	0.8958	0.5897
	VDBN with FS-GOE	0.4681(0.8085)	0.3793	0.4190	0.8941	0.6668
	DBN with RP-GON	0.5172(0.8621)	0.2586	0.3448	0.9010	0.6174
	HDBN with RP-GON	0.5417(0.8333)	0.2241	0.3171	0.9028	0.6015
FAP4 Mutant:meta-fusion	VDBN with RP-GON	0.5278(0.8889)	0.3276	0.4043	0.9028	0.6486
	GeneReg and STRING	0.6000(0.8000)	0.2727	0.3750	0.7222	0.6218
	LASSO and STRING	0.5000(0.7500)	0.1818	0.2667	0.6944	0.5745
	DBN and STRING	1.0000(1.0000)	0.0909	0.1667	0.7222	0.5455
	HDBN and STRING	0.5000(1.0000)	0.0909	0.1538	0.6944	0.5273
	VDBN and STRING	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5745
	GeneReg and FS-GOE	0.5000(0.7500)	0.3636	0.4211	0.6944	0.6218
	LASSO and FS-GOE	0.2500(0.5000)	0.1818	0.2105	0.5833	0.5145
	DBN and FS-GOE	1.0000(1.0000)	0.2727	0.4286	0.7778	0.6364
	HDBN and FS-GOE	0.6667(1.0000)	0.1818	0.2857	0.7222	0.5709
	VDBN and FS-GOE	0.8000(1.0000)	0.3636	0.5000	0.7778	0.6655
	GeneReg and RP-GON	1.0000(1.0000)	0.0909	0.1667	0.7222	0.5455
	LASSO and RP-GON	1.0000(1.0000)	0.0909	0.1667	0.7222	0.5455
	DBN and RP-GON	1.0000(1.0000)	0.0909	0.1667	0.7222	0.5455
	HDBN and RP-GON	1.0000(1.0000)	0.0909	0.1667	0.7222	0.5455
	VDBN and RP-GON	1.0000(1.0000)	0.1818	0.3077	0.7500	0.5909
FAP4 Mutant:fusion	DBN with STRING	1.0000(1.0000)	0.3636	0.5333	0.8056	0.6818
	HDBN with STRING	0.8000(1.0000)	0.3636	0.5000	0.7778	0.6655
	VDBN with STRING	1.0000(1.0000)	0.4545	0.6250	0.8333	0.7273
	DBN with FS-GOE	1.0000(1.0000)	0.3636	0.5333	0.8056	0.6818
	HDBN with FS-GOE	0.8000(1.0000)	0.3636	0.5000	0.7778	0.6655
	VDBN with FS-GOE	0.7143(1.0000)	0.4545	0.5556	0.7778	0.6945
	DBN with RP-GON	1.0000(1.0000)	0.2727	0.4286	0.7778	0.6364
	HDBN with RP-GON	0.7500(1.0000)	0.2727	0.4000	0.7500	0.6182
	VDBN with RP-GON	0.8000(1.0000)	0.3636	0.5000	0.7778	0.6655

Table 4.11: Performance of various DBNs fused with various priors on yeast Net5 and FAP5 with wild-type time series.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net5 Wild-type:meta-fusion	GeneReg and STRING	0.3056	0.2157	0.2529	0.6880	0.5895
	LASSO and STRING	0.3529	0.1569	0.2172	0.7232	0.5692
	DBN and STRING	0.7857	0.0719	0.1317	0.7680	0.5329
	HDBN and STRING	0.6364	0.0458	0.0854	0.7600	0.5187
	VDBN and STRING	0.7045	0.2026	0.3147	0.7840	0.5880
	GeneReg and FS-GOE	0.2923	0.2484	0.2686	0.6688	0.5995
	LASSO and FS-GOE	0.3111	0.1830	0.2305	0.7008	0.5793
	DBN and FS-GOE	0.8095	0.1111	0.1954	0.7760	0.5515
	HDBN and FS-GOE	0.5714	0.0523	0.0958	0.7584	0.5198
	VDBN and FS-GOE	0.7241	0.2745	0.3981	0.7968	0.6207
	GeneReg and RP-GON	0.4063	0.0850	0.1405	0.7456	0.5405
	LASSO and RP-GON	0.1429	0.0131	0.0240	0.7392	0.5065
	DBN and RP-GON	1.0000	0.0261	0.0510	0.7616	0.5131
	HDBN and RP-GON	1.0000	0.0065	0.0130	0.7568	0.5033
	VDBN and RP-GON	1.0000	0.0588	0.1111	0.7696	0.5294
Net5 Wild-type:fusion	DBN with STRING	0.7959	0.2549	0.3861	0.8016	0.6181
	HDBN with STRING	0.7391	0.2222	0.3417	0.7904	0.5996
	VDBN with STRING	0.7813	0.3268	0.4608	0.8128	0.6501
	DBN with FS-GOE	0.8163	0.2614	0.3960	0.8048	0.6218
	HDBN with FS-GOE	0.7609	0.2288	0.3518	0.7936	0.6027
	VDBN with FS-GOE	0.8429	0.3856	0.5291	0.8320	0.6821
	DBN with RP-GON	0.6744	0.1895	0.2959	0.7792	0.5810
	HDBN with RP-GON	0.6000	0.1373	0.2234	0.7664	0.5549
	VDBN with RP-GON	0.6667	0.3007	0.4144	0.7920	0.6284
FAP5 Wild-type:meta-fusion	GeneReg and STRING	0.6667	0.1429	0.2353	0.3906	0.5032
	LASSO and STRING	0.5294	0.2143	0.3051	0.3594	0.4665
	DBN and STRING	0.5000	0.0476	0.0870	0.3438	0.4762
	HDBN and STRING	0.8333	0.1190	0.2083	0.4063	0.5363
	VDBN and STRING	0.6500	0.3095	0.4194	0.4375	0.4962
	GeneReg and FS-GOE	0.7000	0.1667	0.2692	0.4063	0.5168
	LASSO and FS-GOE	0.5500	0.2619	0.3548	0.3750	0.4957
	DBN and FS-GOE	0.7143	0.1190	0.2041	0.3906	0.5108
	HDBN and FS-GOE	0.8889	0.1905	0.3137	0.4531	0.5725
	VDBN and FS-GOE	0.7586	0.5238	0.6197	0.5781	0.6169
	GeneReg and RP-GON	1.0000	0.0476	0.0909	0.3750	0.5238
	LASSO and RP-GON	1.0000	0.0952	0.1739	0.4063	0.5476
	DBN and RP-GON	1.0000	0.0476	0.0909	0.3750	0.5238
	HDBN and RP-GON	1.0000	0.0714	0.1333	0.3906	0.5357
	VDBN and RP-GON	1.0000	0.1429	0.2500	0.4375	0.5714
FAP5 Wild-type:fusion	DBN with STRING	0.8947	0.4048	0.5574	0.5781	0.6483
	HDBN with STRING	0.9375	0.3571	0.5172	0.5625	0.6629
	VDBN with STRING	0.9167	0.5238	0.6667	0.6563	0.7154
	DBN with FS-GOE	0.8000	0.3810	0.5161	0.5313	0.5985
	HDBN with FS-GOE	1.0000	0.3810	0.5517	0.5938	0.6905
	VDBN with FS-GOE	0.8571	0.5714	0.6857	0.6563	0.7110
	DBN with RP-GON	0.7143	0.3571	0.4762	0.4844	0.5465
	HDBN with RP-GON	0.8947	0.4048	0.5574	0.5781	0.6645
	VDBN with RP-GON	0.8000	0.5714	0.6667	0.6250	0.6688

Table 4.12: Performance of various DBNs fused with various priors on yeast Net5 and FAP5 with mutant time series.

Networks	Methods	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>	<i>Acc.</i>	<i>AUC</i>
Net5 Mutant:meta-fusion	GeneReg and STRING	0.2745	0.1830	0.2196	0.6816	0.5756
	LASSO and STRING	0.2656	0.1111	0.1567	0.7072	0.5498
	DBN and STRING	0.8889	0.0523	0.0988	0.7664	0.5251
	HDBN and STRING	0.8000	0.0523	0.0982	0.7648	0.5241
	VDBN and STRING	0.7333	0.1438	0.2404	0.7776	0.5639
	GeneReg and FS-GOE	0.2969	0.2484	0.2705	0.6720	0.6032
	LASSO and FS-GOE	0.2727	0.1569	0.1992	0.6912	0.5700
	DBN and FS-GOE	0.8571	0.1176	0.2069	0.7792	0.5567
	HDBN and FS-GOE	0.7500	0.0588	0.1091	0.7648	0.5273
	VDBN and FS-GOE	0.7660	0.2353	0.3600	0.7952	0.6075
	GeneReg and RP-GON	0.5238	0.0719	0.1264	0.7568	0.5350
	LASSO and RP-GON	0.3500	0.0458	0.0809	0.7456	0.5219
	DBN and RP-GON	1.0000	0.0392	0.0755	0.7648	0.5196
	HDBN and RP-GON	1.0000	0.0131	0.0258	0.7584	0.5065
	VDBN and RP-GON	1.0000	0.0719	0.1341	0.7728	0.5359
Net5 Mutant:fusion	DBN with STRING	0.8108	0.1961	0.3158	0.7920	0.5914
	HDBN with STRING	0.7949	0.2026	0.3229	0.7920	0.5939
	VDBN with STRING	0.8448	0.3203	0.4645	0.8192	0.6514
	DBN with FS-GOE	0.8810	0.2418	0.3795	0.8064	0.6167
	HDBN with FS-GOE	0.8250	0.2157	0.3420	0.7968	0.6017
	VDBN with FS-GOE	0.8382	0.3725	0.5158	0.8288	0.6763
	DBN with RP-GON	0.8095	0.2222	0.3487	0.7968	0.6043
	HDBN with RP-GON	0.6857	0.1569	0.2553	0.7760	0.5685
	VDBN with RP-GON	0.7846	0.3333	0.4679	0.8144	0.6539
FAP5 Mutant:meta-fusion	GeneReg and STRING	0.4118	0.1667	0.2373	0.2969	0.4280
	LASSO and STRING	0.5000	0.1190	0.1923	0.3438	0.4551
	DBN and STRING	1.0000	0.0952	0.1739	0.4063	0.5476
	HDBN and STRING	0.8000	0.0952	0.1702	0.3906	0.5271
	VDBN and STRING	0.8000	0.2857	0.4211	0.4844	0.5790
	GeneReg and FS-GOE	0.5217	0.2857	0.3692	0.3594	0.5011
	LASSO and FS-GOE	0.6667	0.2381	0.3509	0.4219	0.5249
	DBN and FS-GOE	1.0000	0.1667	0.2857	0.4531	0.5833
	HDBN and FS-GOE	0.8000	0.0952	0.1702	0.3906	0.5271
	VDBN and FS-GOE	0.8421	0.3810	0.5246	0.5469	0.6320
	GeneReg and RP-GON	0.6667	0.0476	0.0889	0.3594	0.5238
	LASSO and RP-GON	1.0000	0.0476	0.0909	0.3750	0.5238
	DBN and RP-GON	1.0000	0.0476	0.0909	0.3750	0.5238
	HDBN and RP-GON	1.0000	0.0238	0.0465	0.3594	0.5119
	VDBN and RP-GON	1.0000	0.0714	0.1333	0.3906	0.5357
FAP5 Mutant:fusion	DBN with STRING	0.8889	0.3810	0.5333	0.5625	0.6602
	HDBN with STRING	0.8947	0.4048	0.5574	0.5781	0.6634
	VDBN with STRING	0.9048	0.4524	0.6032	0.6094	0.6959
	DBN with FS-GOE	0.8333	0.3571	0.5000	0.5313	0.6293
	HDBN with FS-GOE	0.8500	0.4048	0.5484	0.5625	0.6466
	VDBN with FS-GOE	0.8696	0.4762	0.6154	0.6094	0.6916
	DBN with RP-GON	0.8333	0.3571	0.5000	0.5313	0.6293
	HDBN with RP-GON	0.8750	0.3333	0.4828	0.5313	0.6255
	VDBN with RP-GON	0.8846	0.5476	0.6765	0.6563	0.7354

As seen from these tables, with prior information, all automatically learned DBNs (DBN, HDBN, VDBN) significantly outperformed the DBNs without priors in Tables 4.1, 4.2 and 4.3, with higher Precision, Recall, F1-score and AUC. This shows all these three quantitative priors in the fusion framework are able to capture more information for predicting GRNs than all DBNs (DBN, HDBN, VDBN) solely from GE data. The DBNs with various priors also outperformed the same priors on recovering benchmark networks that were shown in Chapter 3. It demonstrates the effectiveness of our fusion method on capturing multiple data sources for GRN inference, which were better than GRNs inferred from each single data.

As shown in these tables, all VDBN with various priors performed better than both HDBN and DBN with the same prior on all networks. That means, VDBN with prior learned the best time-delayed GRNs that achieved the *maximum* consistency with the prior structures.

Comparing the various priors in fusion models, various DBNs with prior information from GO, either FS-GOE or RP-GON, performed better than prior from STRING on all networks. This indicates either GO functional similarity or regulatory path from GON can provide more compensation information than the integrated information from STRING. Note that all GRN inference methods performed better on FANs/FAPs than the randomly selected networks. This is biologically reasonable since gene regulation more likely takes place among functionally associated genes than randomly chosen genes.

To compare the generated time-delayed GRN from VDBN and fusion VDBN model, we chose *E.coli* network FAN2 as a representation. Since VDBN with FS-GOE or RP-GON all performed better than VDBN with STRING, we only showed the predicted structure from VDBN with STRING model. Figure 4.4 showed the network structure of FAN2 from benchmark, VDBN, and VDBN with STRING. To analyse the predicted time-delayed GRN structure on yeast networks, we chose Net4 as a representation, and showed the predicted networks from VDBN with STRING model on wild-type or mutant cells in Figure 4.5. In these figures, three types of edges were shown: the true positive (TP) edges those were validated from benchmark networks, are drawn as black lines; the false positive (FP) edges those were predicted by fusion model but not recorded by current benchmark networks, were represented by red lines; and the false negative (FN)

edges those were recorded in benchmark networks but not predicted by our fusion model, were represented as dash lines. The predicted various time delays were also shown on the corresponding predicted edges.

Figure 4.4 showed that the fusion model (VDBN with STRING) predicted more accurate structures than VDBN: two more TP edges were recovered ($\text{fadR} \xrightarrow{\text{regulate}} \text{fadD}$; and a self-regulation on gene iclR); less FP edges (red lines) and less FN edges (dash lines). These observations indicated that the precision, recall and F1-score were all improved.

Figure 4.5 showed that the fusion model (VDBN with STRING) predicted various time delays in GRNs, and the structure predicted from wild-type and mutant time-series data were slightly different. The common structures included all the TP edges (black lines), and some FP edges (red lines). However, some FP edges (red lines) and time delays were different in the two predicted GRNs. For example, the GRN from wild-type GE data had more FP edges than the one from mutant data. This demonstrates that our models are able to capture the dynamics and the dynamic structure may change with different experimental conditions, for example, in our experiments they were different in wild-type cells and mutant cells.

4.5.a: Yeast Net4.

4.5.b: Yeast Net4 Predicted by VDBN with STRING from Wild-type Time-series.

4.5.c: Yeast Net4 Predicted by VDBN with STRING from Mutant Time-series.

Figure 4.5: The GRNs on yeast cell cycle network Net4. The same representation as Figure 4.4. Note that most of FP edges (red lines) are validated from YTRP database.

As seen from the figures, some FP (red) edges were detected but without validation from benchmark networks. Note that the benchmark networks only represented some regulatory relations extracted from literature, without recording the corresponding time delays. Transitive regulations can provide some evidence for time delays since they may result from the delays in gene regulation. Yang et al. [153] provided yeast transcriptional regulatory pathways (YTRP) database, which recorded the possible transitive regulations. However, these YTRP edges are only predicted transitive regulation from shortest path method, but not validated by laboratory experiments. As some of the YTRP edges may not be real gene regulations, we couldn't directly use them as ground-truth networks. We only used them to validate the inferred time-delayed GRNs: if the predicted edges were also consistent with transitive regulations from YTRP, we considered they were true positive edges. On the selected yeast networks, we found that some FP edges in Net4 and FAP4 were validated by the TRP edges. The new precision of these two networks were then computed by counting these validated edges from YTRP as TP edges, and they were shown in the parentheses of Pre. columns of Net4 and FAP4 in the tables. As shown in the tables, the precision was significantly improved on all models, with up to 40% improvement on Net4 and FAP4. This reflects the incompleteness of current benchmark networks without counting the transitive regulations and time-delayed regulation. The very high new precision (almost all above 80%) of fusion models means that, in Figure 4.5, almost all FP edges (red lines) in Net4 were also validated from transitive regulations in YTRP. This shows that our VDBN fusion approach can generate more compensation structures compared with current GRN databases such as RegulonDB [122] and YEASTRACT [135].

4.6 Discussion

In this chapter, we fused various DBNs with various quantitative priors to infer time-delayed GRNs. A polynomial algorithm was adopted to learn the globally optimal structure of these DBNs. The optimal delays in VDBN were learned by a variable-order MCMC method. A Bayesian fusion framework was employed to integrate quantitative structural priors from STIRNG or GO with GE time series to build GRN with delays.

The parameters for weighting the priors were automatically learned by maximizing the consistency of the GE data with the priors.

Experiments were implemented on two synthetic networks, six *E.coli* networks from RegulonDB [122], and four yeast cell-cycle networks from YEASTRACT database [135] in Chapter 3. We implemented the various DBNs (DBN, HDBN, and VDBN), and also fused them with various priors (STRING score, FS-GOE score, and RP-GON confidence score) on all networks. From the comparison results, VDBN outperformed DBN and HDBN with fixed orders. The prior information also significantly improved the performance of all DBNs. Experimental results showed that compared with STRING prior, the GO methods could provide more compensation information of GE time-series data.

There are some possible reasons for the errors of the predicted networks shown in experiments. One reason is that the target networks used for evaluation are only structures without delays. However, what our methods inferred/recovered were the time-delayed gene regulations in given time series under some condition. Thus, it is reasonable that some gene regulations may be not captured in given time series. The other reason is the limitation of current knowledge, and some gene regulations maybe not recorded in gold-standard networks. This was also demonstrated in our experiments, many false positive edges were in fact validated by transitive regulations from yeast transcriptional regulatory pathways.

Our VDBN with priors method can be adapted on continuous data with Bayesian-Gaussian equivalent (BGe) scores [55], and more prior information, such as pathways from KEGG [39], or time-delayed transcriptional regulations from literature [88].

Chapter 5

Extracting Rate Changes in Transcriptional Regulation from MEDLINE Abstracts

5.1 Background and Motivation

Living cells are products of gene expression processes involving regulated transcription over thousands of genes. Transcription is controlled by regulatory proteins binding to specific promoter sequences. The biochemical process by which a collection of regulatory proteins associates with genes across a genome can be described abstractly through a transcriptional regulatory network. There exists a time delay or lag involved in the process where the change of expression of the regulator is transmitted to the change in the target gene expression. Time delay in gene regulation is due to intermediate processes between the gene product and repressor/inducer. Time delay is thus an important element in the construction of a transcriptional regulatory network [121]. However, the validation of time delays involved in gene regulation is a challenge as existing biological databases always lack time-related information of gene regulation. In this chapter, we address the problem of validation of time delays in gene regulation.

There is an abundance of biological knowledge and facts in scientific articles [19, 44, 141, 33]. Due to ever-increasing corpus of scientific articles, it is difficult, if not impossible, to manually extract biological knowledge in literature. Text mining tools help to automatically extract biological events from literature. For example, a number

of relation extraction methods have been applied to extract gene regulation events in molecular interactions [124, 49, 51, 113, 13, 23, 60, 9].

The simplest approach is the co-occurrence based method [9], that predicts the relation between co-occurring entities at the resolution of sentence or abstract. The rationale is that entities that are repeatedly mentioned together are somehow related. Prediction results of the co-occurrence methods typically have a high sensitivity but low specificity. Besides this limitation, details of the regulation such as direction and types are not easily determined by using this method.

Pattern based extraction methods search for textual patterns of events with the help of predefined rules. Rules can be manually compiled [124] or learned from a training corpora [36, 51, 113]. To extract and fire a rule, multiple ontologies have to be consulted in addition to bio-entity vocabularies [99, 36, 113, 13]. Concept ontologies, for example, are constructed from text corpora by using formal concept analysis [36]; the BioInfer project provides an ontology for diverse biological events [51, 113]; Gene Regulation Ontology [13] is designed to model complex events that are part of the gene regulatory processes. Compared with the co-occurrence based methods, pattern based ones increase the specificity at the expense of a decreasing sensitivity.

Semantics analysis using natural language processing tools can help to extract complex rules from underlying sentences. For example, the RelEx relation extraction system [49] uses a simple set of rules to extract relation paths from dependency parse trees. A set of hand-compiled rules was proposed for linguistic analysis and conceptual inference over semantic structures [60]. The rule-based system then matches syntactic-semantic patterns to the dependency structures. However, extracting rules manually from semantic structures is time-consuming and cannot be done easily on large-scale corpora. A current trend in feature design is to learn the semantic rules automatically via machine learning methods. For example, previous work investigated the performance of different Automatic Content Extraction (ACE) feature sets for biomedical relation extraction in a supervised learning setting [23]. Positive and negative regulation relations in biomedical pathways were learned from the lexical semantic annotations in [157]. Rich graph-based feature sets have been proposed to extract complex biological events [18, 85].

In this chapter, we propose a text mining approach to detect regulation delay events from scientific literature. The method is illustrated by constructing a corpus by extracting transcriptional regulation rate changing events for yeast from PubMed abstracts. To automatically extract such events, we need to extract the related textual patterns, but there rarely exist obvious textual patterns about rate changes in transcriptional regulation. We thus create two ontologies from the corpus, based on biological analysis of sub-processes that may result in rate changes in the transcriptional regulation, and the summarization of the positive/negative textual patterns/rules for describing such sub-processes. We then adopt the effective information retrieval (IR) tools to detect target events. Both automatic rule learning method and simple decision rule based method are used to test the effectiveness of the generated ontologies on identifying rate changes in transcriptional regulation. We also extend the ACE-Style features [23] and graph-based features [18, 85] based on our ontologies to obtain a set of conceptual, semantic and sentence graph structure features which can be used to learn the direct evidence for detecting transcriptional regulation rate changing events. The effectiveness of the proposed features will be verified experimentally. A decision tree trained with the proposed features achieves 71.43% F1-score over ten-fold cross-validation.

5.2 Text Preprocessing

5.2.1 Collecting Abstracts and Trigger Words

We downloaded 181,517 abstracts related to yeast from PubMed (<http://www.ncbi.nlm.nih.gov/pubmed>) on September 30, 2012. The downloaded abstracts contain any of the terms “yeast”, “*Saccharomyces cerevisiae*”, or “*S. cerevisiae*” in (i) title, abstract, article fields, or (ii) the head of a Medical Subject Headings (MeSH, a controlled vocabulary for manually annotating PubMed articles) term for the article.

We collected the vocabulary of words related to gene regulations from the BioInfer relation ontology [113] and used heuristics to manually create a list of trigger words relating to rate changes. We then extracted only those sentences from the abstracts that contain both trigger words about gene regulation and rate changes. To find out which trigger words are commonly used in PubMed, we randomly chose hundreds of sentences

containing different trigger words and manually pruned them. The trigger word lists were pruned by removing relation words unrelated to regulation and extending the rate change word list with more words.

5.2.2 Word Normalization

To identify all the genes of yeast, we obtained a list of synonyms and identifiers from the Saccharomyces Genome Database (<http://www.yeastgenome.org>). We also manually created a dictionary to cluster semantically similar trigger words together, using a stemming method like in [49]. We then performed word normalization by (i) replacing all gene names with a placeholder word “genename” to eliminate the differences between diverse genes; and (ii) replacing all keywords in the dictionary with their stemmed counterparts, i.e., the words in the first column of the dictionary. Word normalization helps removing the diversity of textual information, especially the article keywords. It is therefore helpful for summarizing textual patterns or rules for events.

5.2.3 Sentence Filter

Since information extraction methods work poorly on irrelevant raw data, we need text filtering to help reduce the variability of textual data. It is commonly assumed that the desired information is local to a sentence [36, 51, 113, 49]. We thus constrain the event extraction at the sentence resolution.

Using gene regulation and rate change trigger words as filters, relevant sentences were filtered from yeast gene regulation and time related PubMed abstracts. The MEDLINE abstracts were first segmented into individual sentences by using Natural Language Toolkit (NLTK, <http://nltk.org/>) sentence tokenizer. Afterwards, the sentences were tokenized by NLTK word tokenizer into word tokens. The final set of sentences must satisfy all of the following conditions: (i) there are at least two occurrences of gene/protein names tagged as noun in the sentence; (ii) the sentence matches many words from the “regulate” keyword list; and (iii) the sentence matches many words from the rate change keyword list.

The first filter is easy to implement. To implement the other two filters, we used the cosine similarity to quantify the occurrence of keywords in the sentence. Cosine

similarity uses normalized counts that help to remove the bias caused by the overlong sentences. Thus, it can capture more informative sentences than by using occurrence counts. Specifically, each trigger word list (i.e., “regulate” or temporal information) is written as a unit vector $\mathbf{y}$. Each sentence is written as a binary vector $\mathbf{x}$ which has a one in the dimension corresponding to a matched word in the sentence from the trigger word list. Vectors $\mathbf{x}$ and $\mathbf{y}$ have the same dimension as the number of keywords in the trigger list. The cosine similarity of the sentence with the keywords list is then computed as

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

where $\mathbf{x} \cdot \mathbf{y}$ represents the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$; and $\|\cdot\|$ represents the L_2 vector norm.

There are altogether 114,375 sentences that score above zero in both two similarity scores. Since it is infeasible to manually label all of them, we choose only sentences whose cosine similarity to the “regulate” list exceeds 0.1 and whose similarity to the temporal list exceeds 0.15. The final corpus includes 1309 sentences, which are then used as inputs to the next information extraction task.

5.3 Corpus Annotation

5.3.1 Records of Transcriptional Regulation Rate Changing Events

To the best of our knowledge, we are the first group to extract time-delayed gene regulation evidence from literature. Since specific/quantitative time-related information is not common in literature, much less in abstracts, our aim is to detect the events of rate changes in transcriptional regulation instead. The detected rate changes provide qualitative evidence (i.e., shorter or longer) for inferring time delays in transcriptional regulation. Table 5.3.1 gives two examples about transcriptional regulation rate changing events.

Table 5.1: Examples of transcriptional regulation-rate change events – one positive and one negative. Words in bold indicate transcriptional regulations while rate changes in the regulation process are italicized.

Negative Instance	As the translational control in the expression of cycle proteins Cdc13 and Cdc25 constructs positive feedback loops , the dynamical activities of the key components undergoes a <i>rapid rising</i> after a preliminary stage of <i>slow increase</i> . (PMID: 20303984)
Positive Instance	In contrast, degradation of MTH1 is <i>reinforced</i> by glucose repression of MTH1 expression: disappearance of MTH1 is <i>slowed</i> when glucose repression of MTH1 expression is <i>prevented</i> , and this results in a <i>delay</i> in induction of HXT3 expression in response to glucose . (PMID: 16400179)

The positive instance shows that “when glucose repression of MTH1 expression is prevented”, degradation of MTH1 is slowed, and a delay occurs in “induction of HXT3 expression in response to glucose”. Thus it contains a transcriptional regulation rate changing event. As a comparison, the negative instance indicates that “dynamical activities of the key components” undergoes rate changing, but the words “translational control in the expression of cycle proteins Cdc13 and Cdc25” indicates that it is in translational process instead of transcriptional process.

Note that the rate-changing regulation events are different from positive/negative regulations, which have been studied by others [23, 60]. The positive regulation (GO:0010628) and negative regulation (GO:0010629) defined in the Gene Ontology (<http://www.geneontology.org/>) correspond to biological processes that modulate gene expression to increase or decrease, respectively. Both positive and negative regulations belong to gene regulation process. In contrast, the rate-changing regulation studied here refers to a change in the speed or duration of the regulation process. As such, both positive and negative regulations may have rate changes as well.

5.3.2 Corpus Annotation

We annotated the corpus by manually labeling sentences containing transcriptional regulation rate changing events as positive instances and others as negative instances. For positive instances, we identified trigger words that indicate mentions of transcriptional

regulation processes or rate changes of the processes. These words were annotated to facilitate the creation of our time-delay (transcriptional regulation rate change) ontology. In the negative class, the sentence may only include information about gene regulation without rate changes or about a biological process other than transcriptional regulation.

Both direct and indirect evidence exist in the positive instances. We thus further annotate the positive class with two types of events: (i) events with specific information about regulator, regulatee and rate changes in transcription regulation, and (ii) indirect evidence for transcription regulation rate changing events.

A biological practitioner was employed to annotate the corpus. Since there is a single annotator, we are unable to report the inter-annotator agreement score about our annotation. However, during the ontology extraction process, whenever we found annotations inconsistent with the extracted ontologies (biology knowledge), the annotator was asked to double check and correct all annotations (especially the positive class) again.

5.4 Biological Knowledge Driven Ontologies for Rate Changes in Transcriptional Regulation Events

By combining statistical analysis on the annotated trigger words with biological knowledge on gene regulation, we were able to create two ontologies. The first ontology, as shown in Figure 5.4, summarizes the textual patterns and biological reasons for rate changes in transcriptional regulation. The second ontology summarizes the negative rules as illustrated in Figure 5.4.

Figure 5.1: Transcriptional regulation-rate change ontology. The column of big boxes contain the textual patterns corresponding to the sub-processes of transcriptional regulation rate change events. The smaller boxes contain processes or classes of events. The numbers of instances (sentences) corresponding to the process in the corpus are indicated in the bracket next to each process.

Figure 5.2: Negative transcriptional regulation ontology. The rightmost columns of big boxes contain textual patterns corresponding to the sub-processes of negative instances. The other boxes contain processes or classes of events. The numbers of instances (sentences) corresponding to the process in the corpus are indicated in the bracket next to each process.

In order to make our ontologies consistent with previous work, we also provided logical relations between subclasses, events, or processes. Following the convention used in the Gene Ontology database, *is - a* represents the subtype relation and *part - of* is part-whole relation. Following the Gene Regulation Ontology [13], *result - from* refers to the result of some processes or explanations. We also list transitive rules among these relations, which is consistent with the Gene Ontology relation (<http://www.geneontology.org/GO.ontology.relations.shtml>) and Gene Regulation Ontology as follows.

$\forall$ relation $r \in \{is - a, part - of, result - from\}$, we have rules $r \circ is - a \rightarrow r$ and $is - a \circ r \rightarrow r$; moreover,

$$\begin{aligned} part - of \circ result - from &\rightarrow result - from \\ result - from \circ part - of &\rightarrow result - from \\ part - of \circ part - of &\rightarrow part - of \end{aligned}$$

where $r_1 \circ r_2 \rightarrow r_3$ means if A has relation r_1 with B, and B has relation r_2 with C, we can infer that A has relation r_3 with C. Thus, according to these transitive rules, we infer more relations between two terms if an inferred relation path exists between them according to the method described in [87].

5.4.1 Biological Knowledge Based Analysis

The stages in which gene expression is regulated typically include: (i) Chromatin domains; (ii) Transcription; (iii) Post-transcription; (iv) Translation; and (v) Post-translation. The first two stages are mostly related to transcriptional regulation. We thus analyse the causes in these processes that may result in transcriptional regulation rate changes. The textual patterns for transcriptional regulation rate changes related to these two processes are as follows.

(i) Chromatin domains. Unlike prokaryotic, eukaryotic linear DNA molecules coil with nucleosomes tightly and package into chromosomes. When some genes are to be expressed, the region of the activation zone is unpacked, the nucleosome is decomposed, and then transcription related proteins come inside to form transcription complex and perform transcription initialization. Chromosome unpacking is often related to several events such as nucleosomes composition and decomposition, and chromosome structure

modification. Thus, all of these events are the sub-events that result in transcriptional regulation rate changes. Some chemical modifications around histones may change the chromosome structure, which can eventually affect transcription [140]. The processes that modify histones include DNA methylation, microRNA, or DNA-binding of proteins. Example-1 in Table 5.4.1 indicates that gene “Mdm2” ubiquitylates chromatin proteins around the area of target promoter and controls the transcription level of the target gene. The Example-2 in Tabel 5.4.1 indicates that gene “chz1” relates to the ubiquitination of histone H2B and di-methylation. As a result, the rate of binding process of “Sir3p, and Sir4p” is increased.

Table 5.2: Transcriptional regulation-rate change pattern examples.

Example-1	Endogenous Mdm2 is tethered in vivo, presumably via p53, to chromatin comprising the p53-responsive p21(waf1) promoter, and Mdm2 overexpression <i>enhances</i> protein ubiquitylation in the vicinity of a p53 binding site within that promoter. (PMID: 15546622)
Example-2	Deletion of CHZ1 led to <i>reduced</i> ubiquitination of subtelomere-associated H2B, <i>reduced</i> subtelomeric H3K79 dimethylation , and <i>increased</i> binding of Sir3p, and Sir4p at telomere-distal euchromatin regions, correlating with <i>decreased</i> gene expression in subtelomeric regions. (PMID: 20008511)
Example-3	The cofactor npl4-1 and ufd1-2 mutants also exhibit G1 delay and <i>reduced</i> CLN1 promoter activity at 38.5°C, suggesting that Npl4-Ufd1 complex mediates the function of Cdc48 at G1. (PMID: 21526151)

(ii) Transcription. Transcriptional regulation controls when the transcription takes place and how many copies of RNA are synthesized. Transcription factors help RNA polymerase bind to the target region. In this process, protein-DNA and protein-protein interactions are involved. Factors that affect the function of transcriptional factors or RNA polymerase eventually lead to transcriptional regulation rate changes, e.g., promoter activity, RNA polymerase activity, and RNA polymerase binding event. Example-3 in Table 5.4.1 indicates that cofactor genes npl4-1 and ufd1-2 are related to the expression of gene cln1 by affecting the activity of the cln1 promoter.

5.4.2 Negative Rules for Transcriptional Regulation

We have applied the transcription rate change ontology on the corpus with the goal of predicting transcription regulation rate changing events. We obtained satisfactory recall value. But there were many false positive counts. By analysing false positive samples, we discovered some trigger words in the ontology, such as regulation, binding, reducing, and delay, which are widespread in many biological processes, but are not related to transcription. We thus generated the negative transcriptional regulation ontology based on the analysis of trigger words in the false positive sentences. The negative rules for transcriptional regulation are shown in Figure 5.4. They are primarily derived from three types of events, namely the cell cycle regulation, DNA replication, and regulation of other biological processes.

Cell cycle is the process of cell-division, which is sophisticatedly regulated by a series of cell signaling events and gene regulation. Thus, when we apply “regulate” class of rules, we need to combine them with cell cycle negative rules. Example-4 in Table 5.4.2 contains several keywords associated with activation but its target is not transcription. The regulation here refers to the cell-cycle as indicated by the terms “nuclear segregation” and “anaphase I spindle”.

Table 5.3: Negative patterns examples.

Example-4	We show here that the role of these proteins is instead to promote nuclear segregation, including release of the Cdc14 phosphatase required for Cdk1 inactivation and disassembly of the anaphase I spindle. (PMID: 12737807)
Example-5	Here we show that two highly conserved ATP-dependent chromatin-remodeling complexes in <i>Saccharomyces cerevisiae</i> , Isw2 and Ino80, function in parallel to promote replication fork progression. (PMID: 18408730)
Example-6	Of the single codon changes, mutation of the first ATG (ATG1) resulted in the largest increase of the reporter gene PIS1(promoter)-lacZ expression. (PMID: 16997274)

In DNA replication processes, chromosomes must be decomposed and nucleosomes need to re-deposited to form newly-synthesized histones. Thus, when we use rules of histone and chromosomes behaviors, we consider DNA replication based negative

rules. Example-5 in Table 5.4.2 includes keywords about chromosome behaviors such as “chromatin-remodeling”; however, these behaviors actually relate to DNA replication instead of gene expression. The keyword “replication fork” positively confirms this to be a process of DNA replication.

Specific biological processes other than transcription are also included in the negative rules. Example-6 in Table 5.4.2 contains the term “codon”, which indicates the process to be translation process.

The negative transcriptional regulation ontology helps filter out events that are textually similar to transcription regulation events. Note that each sentence in our corpus contains both trigger words for regulation and rate changes. Hence, the negative transcriptional regulation ontology is based on the analysis of sentences that have similar textual patterns with rate-changing regulation events but are in fact not. They can help reduce the number of false positives from using the regulation ontology alone.

5.4.3 Statistical Analysis on Ontologies

To test the coverage of the textual patterns embedded in both ontologies, we shuffle the sentences and plot the average cumulative fraction of textual patterns that occur in randomly ordered sentences in Figure 5.4.3. The figure shows that the first 20% randomly selected sentences contain 60% of the transcriptional regulation rate change rules embedded in the regulation ontology, and 53% negative rules listed in the negative ontology. This demonstrates that our ontologies have a good coverage.

Figure 5.3: The average cumulative distribution of textual patterns from each ontology, over the number of sentences. The left figure shows transcriptional regulation rate change ontology distribution, and the right figure shows negative transcriptional regulation ontology distribution. The data is taken from the average over 100 random shuffles of the corpus sentences.

Figures 5.4 and 5.4 also list the number of matching sentences in each subclass of the two ontologies. For example, the two largest counts in the rate change ontology come from promoter activity (101) and transcription behavior (343). Some sub-classes have just a handful of instances, such as nuclear change (13), chromatin remodeling (12), DNA structure changes (1), and chemical reaction (11). However, due to their important biological implication on the transcriptional regulation or negative events, we also include them into the ontology. This makes our ontology biologically more feasible and complete.

A large number of sentences contain process descriptions that resemble regulation but are in fact non-transcription regulations. This can be seen in the large counts of non-regulation processes. Keywords such as “regulate”, which are used in transcription regulation, are in fact quite generic, i.e., they are used to describe other regulation processes like cell cycle regulation (71), translation (35), DNA replication and repair (49). This demonstrates the necessity of having a negative ontology of rules.

5.5 Identifying Transcriptional Regulation Rate Changing Events with Rule-based Methods

5.5.1 Evaluating Ontologies

We compared the proposed ontologies with two benchmarks: BioInfer ontology [113] and GeneReg ontology [13]. The former has been used to recognize diverse biological events [51, 113]. The latter is designed to model complex events that are part of the gene regulatory processes [13]. We extracted gene regulation rules from the BioInfer ontology. We then extracted trigger words from both ontologies as our regulation rules. To ensure a fair comparison, we extracted trigger words indicating rate changes from our ontologies. The extracted trigger words are listed in Table 5.4.

Table 5.4: Rate change trigger words.

Quicken	activate, accelerate, exert, extend, increase, rise, peak, drastically, rapidly, positive, promote, enhance, assist, acetylase
Delay	inhibit, block, deacetylase, prevent, suppress, abolish, reduce, decrease, degrade, compromise, sustain, alleviate, decline, lower, negative, repress, diminish, limit, fail, lack, delay, late, slow, shut down, less effect, turn off
Change	rate change, affect rate, alter, over period, during period

We used the decision tree from the Weka machine learning package (<http://weka.wikispaces.com/>) as a tool to evaluate the discriminative power of these ontologies on identifying the transcriptional regulation rate changing event. Specifically, we built several decision trees, each of which uses a specific ontology-based rules as rule bases (features). For the BioInfor ontology and the GeneReg ontology, we incorporated them with rate change trigger words as the rule (feature) set. For the proposed regulation rate change ontology and negative rules, besides testing them individually, we also evaluated a combination. These decision trees are trained on a subset of the aforementioned corpus. They actually act as an automatic rule pruning process [134]. After training, the refined rules are tested over another portion of the corpus. By comparing the testing results, we see the discriminative power of various ontologies.

Table 5.5 reports ten-fold cross-validation results of training decision trees with rule sets derived from various ontologies. It can be seen that our ontologies outperform

the two benchmarks on all metrics. The proposed regulation rate change ontology can predict the positive class with high precision and recall. The proposed negative rules can help reduce the false positive rate and improve precision of regulation rate change ontology; the combined ontologies thus achieve the highest F1-score of about 76%. This demonstrates the effectiveness of both our regulation rate change ontology and negative rules.

Table 5.5: Performance of decision tree prediction using rule sets derived from different ontologies.

Model	<i>Prec.(%)</i>	<i>Rec.(%)</i>	<i>F1(%)</i>	<i>Acc.(%)</i>
BioInfer ontology	64.83	62.46	63.62	80.52
GeneReg ontology	68.31	65.83	67.05	82.35
Regulation rate change ontology	72.06	77.31	74.59	85.64
Combined ontologies	73.67	77.59	75.58	86.33

5.5.2 Deterministic Rule-based Methods

The proposed ontologies can be used to derive rules for identifying transcriptional regulation rate changing events. A straightforward rule is to treat sentences in the corpus as positive instances if they contain sub-events from the regulation rate change ontology. We name this as “Regulation-based Rule”. Table 5.6 shows the results of predicting transcriptional regulation rate changing events using this method. Venturing a step further, we also explored the feasibility of combining both the regulation rate change ontology and negative rules based ontology. Specifically, the following rules are used to predict transcriptional regulation rate changing events.

- If a sentence only contains sub-events from the regulation rate change ontology, it is classified as a positive instance.
- If it also contains sub-events from the non-regulation ontology, in addition to containing sub-events from the regulation rate change ontology, we classify it as a negative instance.
- If no sub-events from the regulation rate change ontology exist in the sentence, it is classified as a negative instance.

The performance of the “Combined Rule” approach is shown in Table 5.6.

Table 5.6: Classification performance of various ontology rule-based methods.

	<i>Prec. (%)</i>	<i>Rec. (%)</i>	<i>F1 (%)</i>	<i>Acc. (%)</i>
Regulation-based Rule	60.04	93.84	73.22	81.28
Combined Rule	67.08	90.20	76.94	85.26

From Table 5.6, it can be seen that both methods are able to identify transcriptional regulation rate changing events with reasonable accuracies. By comparing the “Regulation-based Rule” with the “Combined Rule”, we see that non-regulation ontology has successfully improved the precision from 60% to 67%. Both precision and F1-scores are improved, with a slight degradation (3%) in recall. We also notice that the performance of our deterministic rule-based methods are comparable to that of the decision tree approach using “Combined ontologies” as shown in Table 5.5. This demonstrates the effectiveness of our ontology on discriminating transcription regulation rate changing events from other regulation events.

5.6 Identifying Direct Evidences for Events by Machine Learning Methods

The above rule-based methods are capable of identifying transcriptional regulation rate changing events, but fail to identify specific details of the rate-changing regulation events. The details are important as they are indispensable for extracting gene regulatory network (GRN) from literature. To extract detailed information, we need to consider the position of genes and their relations such as part-of-speech (POS) tags and grammatical relations between keywords, etc.

5.6.1 Direct and Indirect Evidence Records for Events

Through manual inspection of the transcription regulation rate changing events, we discovered that there are direct or indirect evidence of transcription regulation rate changing events. The direct evidence provide complete information about the regulator, regulatee

and rate changes of the event. The indirect evidence in the positive class have following two subclasses:

- (i) Subclass I: incomplete regulation information. The sentence provides incomplete or partial regulation information. For example, according to the snippet “a decrease in the amount of HSP12 and HSP104 mRNAs” of Example-7 in Table 5.6.1, we know that the expression of “HSP12 and HSP104” changes. However, we are not sure about the regulation relationship and regulation rate change. Thus, we need more information to confirm it as a regulation rate changing event.
- (ii) Subclass II: genetic engineering manipulation. The sentence may include information about gene regulation and transcription that are unnatural, which means that the gene information included in the sentences are not necessary the original genes in our target organism. As Example-8 shown in Table 5.6.1, “multicopy plasmid” and “Gal-IME1 construct” indicates that the related genes are external or went through some engineering manipulation. Thus, this type of events only provide indirect evidence for our target event; more information is needed to confirm that the genes are indeed original.

Table 5.7: Examples of indirect evidence.

Example-7	The results indicate that during the first hours of microvinification there is an increase in the GPD1 mRNA levels with a maximum about one hour after inoculation, and a decrease in the amount of HSP12 and HSP104 mRNAs, although with differences between them. (PMID: 12086182)
Example-8	Four different conditions were found to cause expression of Ime1 protein in vegetative cultures: elevated transcription levels due to the presence of IME1 on a multicopy plasmid; elevated transcription provided by a Gal-IME1 construct; G1 arrest due to alpha-factor treatment; G1 arrest following mild heat-shock treatment of cdc28 diploids. (PMID: 8483452)

To extract detailed information about transcriptional regulation rate changing events, we need to detect direct evidence from indirect evidence. As indicated by the examples shown in Table 5.6.1, direct and indirect evidence have similar textual patterns. We thus need to formulate semantic structural features to extract the direct evidence.

5.6.2 Learning Direct Evidences by a Decision Tree

We propose a feature-based method that incorporates diverse lexical, syntactic and semantic features to automatically extract transcriptional regulation relations. Inspired by the Automatic Content Extraction (ACE) program features [23] and state-of-the-art rich graph features [18, 85], which have proven useful in detecting relations between entities, we define three classes of features, bag-of-words, graph and shortest path features as follows:

- (i) Keyword-tag: a combination of the keywords defined in our ontologies, and their POS tags, which indicate their grammatical roles in sentences. The keywords in the features are normalized to reduce the diversity of words with the same tags.
- (ii) Word-relation-word: two words concatenated by the name of their dependency relation type. The relation is extracted from the shortest relation path between genes and keywords in the dependency tree derived from the Stanford NLP parser (<http://nlp.stanford.edu/software/lex-parser.shtml>).
- (iii) Gene-keyword-distance: a triplet of gene, keyword, and length of the shortest relation path between them in the dependency tree.

Compared with the classical window scoping and chunking entity recognition approach [23], we extracted more concise and informative word features (keywords) by using our structured ontologies. We also considered the tag types of keywords in addition to considering the entity type [23]. In [23], the full parsed features include the path of phrase and phrase labels between two entity mentions in the parser tree. To consider relational trigger words, they also exploit features indicating whether the top phrase in the parse path between the entity mentioned contains the relation triggers. We also considered these two kinds of features, but express them differently with the help of the informative trigger words. Similar to the full parsed features in [23], we considered the dependence relations between entity genes and the relation trigger words. Specifically, we denoted the shortest path in the full parse tree between them as the “word-relation-word” form. Thus, all phrase labels are in fact shown independently and the phrase words are also included. We summarized the shortest path length between gene and keywords as

the “gene-keyword-distance” feature. Hence, we in fact re-formalized and improve the ACE features in [23] to use our ontology for improved automatic relation extraction.

With the help of the informative keywords provided by our ontologies, the proposed features actually incorporated the state-of-the-art rich graph features [18, 85]. Specifically, “keyword-tag” feature was the combination of “token features” and “POS of tokens” in [18, 85]. Instead of using all tokens in the path, we chose more informative keywords from our ontologies. The “word-relation-word” feature combined the 2-gram consecutive tokens and their “edge feature” (i.e., dependency type of the edge) in the path [18, 85]. And “gene-keyword-distance” feature covered the “frequency feature” which is the length of the shortest path [18, 85]. Thus, the proposed features in fact covered state-of-the-art features [18, 85] and were more informative due to our ontologies. In fact, we also implemented the experiment that adding all tokens in the path and their POS into our features, but it didn’t improve any performance. This indicates that the keywords from our ontologies provide enough information of tokens.

After extracting the graph features, we performed following post-processing to reduce the diversity of features: (i) for the dependency type of edge in the path, we used generic types (i.e., the dependency types in the second level of hierarchy tree of typed dependencies as shown in [26]) instead of more specific ones; and (ii) for the length of the shortest path, we used three nominal features (i.e., near, moderate and far) instead of the specific value. If the length is less than three, it is treated as “near”; if the length ranges from four to six, it is treated as “moderate”; and if the length is larger than six, it is treated as “far”. Moreover, to reduce the word diversity in the features, we also performed word normalization by using their normalized representation instead of original forms in texts. Experimental results show that applying post-processing on extracted features can improve the performance.

There are 357 sentences in total, where 211 sentences belong to direct evidence classes, with the remaining belonging to indirect evidence classes. To evaluate the proposed feature-based approach, we performed ten-fold cross-validation on the rate-changing transcriptional regulation events by using the decision tree as the classification method. The results are reported in Table 5.8 with the row header “Combined Features”.

Table 5.8: Performance of ten-fold cross-validation decision tree methods with various feature sets on identifying direct evidence instances.

Feature sets	<i>Prec.(%)</i>	<i>Rec.(%)</i>	<i>F1(%)</i>	<i>Acc.(%)</i>
Baseline features [23]	59.82	62.09	60.93	52.94
All rules features	59.36	70.62	64.50	54.06
Original combined features	59.12	82.94	69.03	56.02
Combined features	60.26	87.68	71.43	58.54

As a baseline, we implemented the ACE features as described in [23]. They use all words between entities/keywords mentions as word features. Parser related features among entities/keywords are also included. The results of this method are shown in Table 5.8 with the row header “Baseline features”. To show the performance of ontologies on identifying direct evidence, we treated all textual patterns from the two ontologies as features. The results are listed in Table 5.8 with the row header “All rules features”. To verify the effectiveness of post-processing, we implemented the experiment on combined features without applying post-processing. The results are shown in Table 5.8 with the row header “Original combined features”, which also indicate the performance of using graph features in [18, 85]. Benchmarking against these baselines is important as it shows whether the proposed features are more effective than baseline features [23], and in fact necessary to identify direct evidence.

As shown in Table 5.8, the proposed “Combined features” significantly improve the recall (about 26% improvement) compared to the baseline. It demonstrates the importance of incorporating semantic patterns in the features. Besides, the proposed features achieved an F1-score of 71.43%, which is about 11% better than the baseline. This shows that the proposed features are effective in mining patterns from direct evidence instances.

When direct evidence records are detected, the specific information of rate-changing transcriptional events are identified automatically from the regulation, gene/protein name, and rate change related trigger words. The rate change directions are intuitively identified from the class of rate changes related trigger words as shown in Table 5.4. As suggested in [23], we can use the 13 frequent textual patterns between “agent” and “patient” to identify regulator and regulatee from gene/protein names. Besides, the effective sentence simplification rules described in [98] can be adopted to remove the irrelevant information, i.e., the noisy genes or keywords.

5.7 Discussion

In this chapter, we derived a new ontology to summarize the textual patterns of rate-changed transcriptional regulation and proposed a method for identifying time-delayed events in gene regulation. We manually created a corpus containing events of rate-changing transcriptional regulation, which can now be downloaded from <https://sites.google.com/site/wentingntu/data>. By statistically analysing the textual patterns of positive instances, combined with biological reasoning over the transcriptional regulation rate change, we cataloged sub-processes and their textual patterns into a transcriptional regulation rate change ontology. Similarly, we established an ontology to collect negative transcriptional regulation textual patterns. Experimental results show that our ontologies outperform state-of-the-art gene regulation ontologies when used together with our decision tree based rule learning method on our corpus. We also proposed some deterministic decision rules by using two established ontologies to identify the events. Experimental results show that this deterministic rule-based method can achieve comparable performance with the automatic rule learning method. This demonstrates the effectiveness of our ontologies and deterministic decision rules on identifying the transcriptional regulation rate changing events.

Since both direct and indirect evidence exist in the transcriptional regulation rate changing events, we need to figure out direct evidence to confirm the final transcriptional regulatory network with rate changes. However, ontology based rules fail to identify direct evidence due to the similarity of their textual patterns with indirect evidence. We thus proposed effective feature extraction methods based on the ontologies to identify direct evidence of events. Experimental results show that it achieves a 71.43% F1-score for ten-fold cross-validation. This demonstrates the effectiveness of the proposed feature mining methods to identify direct evidence.

The detected transcriptional regulation rate changing events can be used as a guidance on detecting time delays from gene expression data [65]. It is easy to use our corpus and ontologies to collect other related events. As our ontology catalogs sub-events of transcriptional regulation rate changing events, it can be applied for sub-events detection, e.g., promoter activity, expression changing events, etc. How to combine the context

information to help predict the event information from the indirect evidence remains an open problem.

Chapter 6

Conclusions and Future Works

This chapter summarizes the thesis and presents several future directions of the research work.

6.1 Summary & Conclusions

This thesis proposed novel methods to infer and validate GRNs with delays from multiple data sources including gene ontology data, literature mining data, gene expression time series data, and quantitative protein-protein interaction data.

6.1.1 GRN from GON

In Chapter 3, we incorporated GO enrichment to improve the detection of functional similar genes from GO. The functionally associated pathways were constructed based on the functional similarities with GO enrichment among genes. To construct GRN, we developed a novel regulatory path method: gene regulations were inferred based on GO “regulate” relation, furthermore, transitive regulations were detected by applying transitive rules of GO relations within functionally associated pathways.

Our methods were demonstrated by experiments implemented on five functionally associated networks and five randomly chosen networks of *E.coli* and yeast cell cycle. The proposed functional similarity score outperformed both the previous functional similarity score and the functional association score from STRING on recovering almost all of these networks. Experiments showed that our regulatory path method performed better on functionally associated pathways than extracting from randomly chosen networks.

This demonstrates that our approach to constrain regulatory paths within functional association pathways is biological reasonable and valid. Compared with state-of-the-art time-delayed GRN inference methods from GE data, “DBN”, “GeneReg” and “LASSO”, our RP-GON method significantly outperformed them on all benchmark networks. This indicates that transitive regulations play an important role in GRN and their detection improves the accuracy of the generated GRN. We also demonstrated that GO can be used effectively to detect transitive regulations. The errors in the networks were partly due to the incompleteness of the GO databases and target networks. The coverage/recall of the predicted networks could be improved by fusing the GRN deduced from GON with other biological data.

One extension to this work is to identify ontology terms that are specific to the pathways under consideration, for example, terms related to cell-cycle functions in our experiments. The GRN developed by our method could be useful for validation of networks built by other experimental or computational approaches. Also, the methods could be used with other biological data to build more accurate GRNs.

6.1.2 Fusion of Multiple Biological Data Sources

In Chapter 4, we also proposed a fusion method to detect time-delayed gene regulations from gene expression data with various quantitative priors: protein interactions from STRING, functional associations or gene regulations inferred from GO.

We adopted variable-order DBN (VDBN) to model GE data, which allows gene regulations to have various delays, and thus providing biologically plausible models of time-delayed GRN. As the order of DBN increases, the number of parameters in the model and the search space increase exponentially. Therefore, we employed *GlobalMIT*⁺ [144], a polynomial algorithm to learn the globally optimal DBN, in our VDBN structure learning to reduce the complexity. The variable-order Monte Carlo Markov Chain (MCMC) scheme [114] was adopted to learn the optimal time delays of each gene.

Furthermore, we included appropriate priors such as protein interaction score from STRING, functional association score from GO, or GRN from GO network, to improve the accuracy and reduce the complexity of learning DBNs. Transcriptional factors were included to constrain the parental genes in GRN structure learning since they act as

regulators. To incorporate information from various priors into the process of GRN inference via VDBN from gene expression (GE) time series data, we used a Bayesian fusion framework to take into account the quantitative priors. The parameters for weighting the priors were automatically learned by maximizing their consistency with the inferred GRN from GE data.

Experimental results on two synthetic networks and the same ten real networks in Chapter 3 showed that VDBN significantly outperformed DBN and HDBN with fixed orders. The prior information also significantly improved the performance of all DBNs. This demonstrates that our fusion method quantitatively captures the coherent and dependent relations from multiple data sources, which is both effective and biologically closer.

Our fusion model shows the importance of fusing multiple sources of biological evidence to create a useful and reliable GRN. In fact, our framework can be used to fuse more data sources, such as biological pathways (KEGG) and literature mined information, etc.

6.1.3 Deriving Time-delayed Regulatory Events from Literature

In Chapter 5, we developed text mining tools to extract events of gene regulation and protein interaction with time delays. Due to the lack of temporal information in existing biological databases, we derived evidence from literature abstracts for validation of GRN with time delays. We manually created a corpus containing events of rate-changing transcriptional regulation, which is now available in public domain from <https://sites.google.com/site/wentingntu/data>.

By statistically analysing the textual patterns of positive instances, combined with biological reasoning over the transcriptional regulation rate changes, we cataloged sub-processes and their textual patterns into a transcriptional regulation rate change ontology. Similarly, we established an ontology to collect negative transcriptional regulation textual patterns. Experimental results showed that our ontologies outperformed state-of-the-art gene regulation ontologies when used together with our decision tree based rule learning method on our corpus. We also proposed some deterministic decision rules by using two

established ontologies to identify the events. Experimental results showed that this deterministic rule-based method achieved comparable performance with the automatic rule learning method. This demonstrates the effectiveness of our ontologies and deterministic decision rules on identifying the transcriptional regulation rate changing events.

Since both direct and indirect evidence exist in the transcriptional regulation rate changing events, we need to figure out direct evidence to confirm the final transcriptional regulatory network with rate changes. However, ontology based rules fail to identify direct evidence due to the similarity of their textual patterns with indirect evidence. We thus proposed effective feature extraction methods based on the ontologies to identify direct evidence of events. Experimental results showed that it achieved a 71.43% F1-score for ten-fold cross-validation. This demonstrates the effectiveness of the proposed feature mining methods to identify direct evidence.

The detected transcriptional regulation rate changing events can be used as a guidance on detecting time delays from gene expression data [65]. It is also easy to use our corpus and ontologies to collect other related events. As our ontology catalogs sub-events of transcriptional regulation rate changing events, it can be applied for sub-events detection, for example, promoter activity, expression changing events, etc.

6.2 Summary of Contributions

The main highlights of the contributions of our research are listed as follows.

- (i) We incorporated GO enrichment to detect the functionally similar genes; thereby, we constructed functionally associated pathways by clustering genes based on their functional similarities.
- (ii) We proposed a novel algorithm to detect “regulatory paths” from GO networks. Specifically, the shortest regulatory paths were detected from among multiple edges by using transitive rules of GO term relations.
- (iii) We proposed to use variable-order DBN to represent and fuse multiple data sources to infer GRNs with delays.

- (iv) We proposed an automatic and effective fusion method to fuse multiple data sources such as gene expression, transcription factors, and *priori* biological networks, to learn time-delayed GRN. The predicted time-delayed regulatory interactions were shown much more accurate and biological plausible than those predicted solely from single data. The fusion framework can be adapted on nicely articulate multiple data sources.
- (v) A corpus of yeast regulation related abstracts was manually labeled with events related to rate changes in transcriptional regulation.
- (vi) In order to capture delays in gene regulatory events automatically, we created an ontology of sub-processes that are likely to result in transcription rate changes by combining textual patterns and biological knowledge.
- (vii) We proposed several effective feature extraction methods based on the created ontology to identify the direct evidence with specific details of rate changing in transcriptional regulation events.

6.3 Future Works

6.3.1 A Novel Gene Interaction Scoring Scheme for Fusing Transitive PPIN and Gene Ontology

As future of this work, one could do is to generate integrated functional association networks, especially considering the topological properties of biological networks. We'll propose a new scoring scheme for measuring the gene interactions from current biological networks, such as the quantitative PPIN from STRING, and GO functional association network. Since most of existed biological networks are incomplete, inspired by [61], one could use downward random walks to simulate the incompleteness of current annotations to improve the existing similarity measures. One would adopt the random walks to simulate the incompleteness of current biological networks.

One can use the method of [131] to estimate PPI scores by modeling protein interactions as a function of two random variables: (i) the integrated PPI confidence score B from STRING, and (ii) the functional similarity $FunSim$ of the corresponding genes

computed from GO. Let W denote the graph with connection strengths computed from these two variables:

$$W = \alpha B + (1 - \alpha) FunSim \quad (\text{Eq. 6.1})$$

Similarly as [7, 131], the weight of each random variable can be learned from a training set of positive and negative examples.

Random walk models including the Markov Clustering Algorithm [45], PageRank [108], etc., have been successfully used to model the link structure of graphs. Likewise, we can extract the topological structure of our extended PPIN by using a random walk model instead of the commonly used small-world clustering coefficient [53].

Given an undirected graph W , the random walk transition matrix T is defined as follows.

$$T = \begin{cases} c \frac{W_{ij}}{\sum_{j=1}^I W_{ij}} + \frac{1-c}{I} & \text{if } W_{ij} \neq 0 \\ \frac{1-c}{I} & \text{otherwise} \end{cases} \quad (\text{Eq. 6.2})$$

where $c \in (0.5, 1)$ is the fusion parameter (a.k.a. the damping factor in PageRank, typically set to 0.85), which determines how probable the next transition occurs from among one of the outgoing links versus just going to any random link.

Enright et al. [45] has shown that the transition matrix T will converge quadratically to an equilibrium state, which represents the topological connectivity of the graph. The converged matrix is computed by

$$\hat{T} = \lim_{k \rightarrow +\infty} T^k. \quad (\text{Eq. 6.3})$$

The converged matrix shows how likely any protein pair will be related.

Using a method similar to that of [61] to account the incompleteness of GO, the integrated protein interactions/functional association score would be $0.5 * W + 0.5 * \hat{T}$. One can implement some experiments to test the efficacy of the proposed quantitative protein interaction scores compared with STRING score on recovering gene interaction networks. This method would provide a framework on quantitatively integrating current evidence with incompleteness simulation of biological networks. The integrated network could be used in our fusion framework as quantitative priors for inferring time-delayed GRNs, or directly used for validating GRN inference methods.

6.3.2 Fusion of Integrated Gene Interactions with Quantitative Time-Delayed GRN Inference Models

Another potential future work is to study the quantitative models for delayed GRNs, by using delay differential equations (DDE) to model the details of delay mechanisms. In the current work, we mainly investigated the graphical models and various DBNs to build GRNs with delays. The predicted GRNs represent the dependencies (linear or non-linear) among genes. They neglected the detailed information such as positive or negative regulation, and how reliable the predicted edges are. The quantitative GRN models such as DDEs can help to recover these information, thus they can help to understand the underlying behaviors of gene regulation. One would to apply these models on Time-Delayed GRN Inference. The main difficulty of these models comes from the huge number of parameters, the time complexity and instability of generated networks. One thus needs to seek appropriate fusion methods to combine the integrated Gene Interactions from current biological networks for generating more reliable and accurate GRNs. Hasan et al. [63] used S-System model and differential evolution based inference method to capture cellular dynamics using available mutual interaction information among genes. They proposed a new fitness function to effectively incorporate prior information, which guides the inference method to deduce correct skeletal structure of the network with more accurate parameter values. The fitness function mirrors user confidence in the validity of knowledge and helps in narrowing down the search range of the model parameters for highly confident knowledge. One would adopt the fitness function in DDE model to infer quantitative GRNs with various time delays.

Publications

Journals

- (i) Wenting Liu, Kui Miao, Guangxia Li, Kuiyu Chang, Jie Zheng and Jagath C. Rajapakse. Extracting Rate Changes in Transcriptional Regulation from MEDLINE Abstracts. BMC Bioinformatics, 2014, 15(Suppl 2):S4.
- (ii) Wenting Liu, Guangxia Li, James Cheng: Fast PageRank Approximation by Adaptive Sampling. Knowledge and Information Systems (KAIS), 2013, pp.1-20.
- (iii) Guangxia Li, Steven C. H. Hoi, Kuiyu Chang, Wenting Liu, Ramesh Jain: Collaborative Online Multitask Learning. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2013.

Conferences

- (i) Wenting Liu, Kui Miao, Guangxia Li, Kuiyu Chang, Jie Zheng and Jagath C. Rajapakse. Extracting Rate Changes in Transcriptional Regulation from MEDLINE Abstracts. Asia Pacific Bioinformatics Conference (APBC), 2014.
- (ii) Wenting Liu, Kuiyu Chang, Jie Zheng, Jain Divya, Jung-Jae Kim and Jagath C. Rajapakse. Gene Regulatory Networks from Gene Ontology, The International Symposium on Bioinformatics Research and Applications (ISBRA), Lecture Notes in Computer Science, Vol. 7875, 2013, pp. 87-98.
- (iii) Zhen Hai, Gao Cong, Kuiyu Chang, Wenting Liu, Peng Cheng: Coarse-to-Fine Review Selection via Supervised Joint Aspect and Sentiment Model. ACM Special Interest Group on Information Retrieval (SIGIR) Conference, 2014: 617-626.

- (iv) Wenting Liu and Jagath C. Rajapakse. Functional Coherent Network Motifs for Better Inference of Gene Regulatory Network. In EMBO Workshop (SigN), Singapore. 2013. (Poster Presentation)

Submissions in Preparation

- (i) Wenting Liu, Guangxia Li, Kuiyu Chang, Jagath C. Rajapakse: Fusing Gene Ontology and Transitive Protein-Protein Interaction for Gene Regulatory Networks. For submission to PLOS ONE, 2015.
- (ii) Wenting Liu, Jagath C. Rajapakse: Logical Inference of Gene Regulatory Networks based on Gene Ontology Annotations of Genes. For submission to Bioinformatics, 2015.
- (iii) Wenting Liu, Jagath C. Rajapakse: Fusing Functional Associations within Variable-order Dynamic Bayesian Network for Time-delayed Gene Regulatory Networks. For submission to Bioinformatics, 2015.

References

- [1] J. R. Adrian Alexa and T. Lengauer. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, 22(13):1600–1607, 2006.
- [2] C. Alfarano, C. Andrade, K. Anthony, N. Bahroos, M. Bajec, K. Bantoft, D. Betel, B. Bobechko, K. Boutilier, and E. Burgess. The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Res*, 33(Database issue):D418–424, 2005.
- [3] G. Alterovitz, M. Xiang, and M. Mohan. GO PaD: the gene ontology partition database. *NAR*, 35:322–327, 2007.
- [4] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, M. Feuermann, A. Ghanbarian, S. Kerrien, and J. Khadake. The IntAct molecular interaction database in 2010. *Nucleic Acids Res*, 38(Database issue):D525–531, 2010.
- [5] M. Ashburner, C. A. Ball, and J. A. Blake. Gene Ontology: tool for the unification of biology. *Nat Genet*, 25:25–29, 2000.
- [6] F. Azuaje, H. Wang, H. Zheng, O. Bodenreider, and A. Chesneau. Predictive integration of Gene Ontology-driven similarity and functional interactions. In *Proc. IEEE-ICDM 2006 Workshop on Data Mining in Bioinformatics (ICDM’06)*, pages 114–119, 2006.
- [7] J. S. Bader, A. Chaudhuri, and J. M. Rothberg. Gaining confidence in high-throughput protein interaction networks. *Nat Biotechnol*, 2004.

- [8] Z. Bar-Joseph, A. Gitter, and I. Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nat Rev Genet*, 13(8):552–564, 2012.
- [9] A. Barbosa-Silva, T. G. Soldatos, I. L. Magalhaes, G. A. Pavlopoulos, J.-F. Fontaine, M. A. Andrade-Navarro, R. Schneider, and J. M. Ortega. LAITOR - literature assistant for identification of terms co-occurrences and relationships. *BMC Bioinformatics*, 11(70), 2010.
- [10] K. Basso, A. Margolin, and G. Stolovitzky. Reverse engineering of regulatory networks in human b cells. *Nature Genetics*, 37(4):382–390, 2005.
- [11] A. Battle, M. Jonikas, and P. Walter. Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology*, 6(379), 2010.
- [12] T. Beissbarth and T. Speed. GOstat: Find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, 2004.
- [13] E. Beisswanger, V. Lee, J.-J. Kim, D. Rebholz-Schuhmann, A. Splendiani, O. Dameron, S. Schulz, and U. Hahn. Gene Regulation Ontology (GRO): Design principles and use cases. *Studies in Health Technology and Informatics*, 136:9–14, 2008.
- [14] M. R. Bennett and J. Hasty. Microfluidic devices for measuring gene network dynamics in single cells. *Nature Reviews*, 2009.
- [15] A. Bernard and A. J. Hartemink. Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. In *Pacific Symposium on Biocomputing*, pages 459–470, 2005.
- [16] J. Björne, F. Ginter, J. Heimonen, S. Pyysalo, and T. Salakoski. Learning to extract biological event and relation graphs. In *17th Nordic Conference on Computational Linguistics*. Citeseer, 2009.

- [17] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski. Extracting complex biological events with rich graph-based feature sets. In *BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 10–18. Boulder, USA, 2009.
- [18] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski. Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence*, 27(4):541–557, 2011.
- [19] C. Blaschke and A. Valencia. The functional genomics network in the evolution of biological text mining over the past decade. *New Biotechnology*, 30(3), 2013.
- [20] E. I. Boyle, S. Weng, and J. Gollub. GO::TermFinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.
- [21] D. Bratsun, D. Volfson, L. S. Tsimring, and J. Hasty. Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences*, 102(41):14593–14598, 2005.
- [22] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *TRENDS in Biotechnology*, 20(11), 2002.
- [23] E. Buyko, E. Beisswanger, and U. Hahn. Testing different ACE-style feature sets for the extraction of gene regulation relations from MEDLINE abstracts. In *Semantic Mining in Biomedicine*, Turku, Finland, Sept. 2008.
- [24] P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and A. Pascual-Montano. GENECODIS: a web-based tool for finding significant concurrent annotations in gene lists. *Genome Biology*, 8(R3), 2007.
- [25] C. I. Castillo-Davis and D. L. Hartl. GeneMerge—post-genomic analysis, data mining, and hypothesis testing. *Bioinformatics*, 19(7):891–892, 2003.
- [26] M. catherine De Marneffe and C. D. Manning. Stanford typed dependencies manual. http://nlp.stanford.edu/software/dependencies_manual.pdf, 2008.

- [27] A. Ceol, A. C. Aryamontri, L. Licata, D. Peluso, L. Briganti, L. Perfetto, L. Castagnoli, and G. Cesareni. MINT, the molecular interaction database: 2009 update. *Nucleic Acids Res*, 38(Database issue):D532–539, 2010.
- [28] E. Cerami, B. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. Bader, and C. Sander. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Res*, 39(Database issue):D685–690, 2011.
- [29] J. Chabalier, J. Mosser, and A. Burgun. A transversal approach to predict gene product networks from ontology-based similarity. *BMC Bioinformatics*, 8(235), 2007.
- [30] I. Chaturvedi and J. Rajapakse. Detecting robust time-delayed regulation in mycobacterium tuberculosis. *BMC Genomics*, 10(Suppl 3):S28, 2009.
- [31] I. Chaturvedi, M. Sakharkar, and J. Rajapakse. Validation of gene regulatory networks from protein-protein interaction data: Application to cell-cycle regulation. In *Pattern Recognition in Bioinformatics*, pages 300–310, 2007.
- [32] H. Chen and V. VanBuren. A review of integration strategies to support gene regulatory network construction. *The Scientific World Journal*, 453, 2012.
- [33] M. L. Chen Li and D. Rebholz-Schuhmann. Biological network extraction from scientific literature: state of the art and challenges. *Briefings in Bioinformatics*, 2013.
- [34] C. Cheng, R. Alexander, R. Min, and et.al. Understanding transcriptional regulation by integrative analysis of transcription factor binding data. *Genome Research*, 22(9):1658–1667, 2012.
- [35] D. Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics*, pages 121–130, 1996.
- [36] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.

- [37] C. M. Curtis Huttenhower, Matt Hibbs and O. G. Troyanskaya. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897, 2006.
- [38] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, 23, 1998.
- [39] N. Díaz-Díaz, F. Gómez-Vela, and J. A.-R. Domingo S. Rodriguez-Baena. Gene regulatory networks validation framework based in kegg. In *Lecture Notes in Computer Science*, volume 6679, pages 279–286, 2011.
- [40] C. P. de Campos and Q. Ji. Properties of Bayesian dirichlet scores to learn Bayesian network structures. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*, pages 431–436, 2010.
- [41] L. de Campos. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *The Journal of Machine Learning Research*, 7:2149–2187, 2006.
- [42] A. de la Fuente, P. Brazhnik, and PedroMendes. Linking the genes: inferring quantitative gene networks from microarray data. *TRENDS in Genetics*, 18(8), 2002.
- [43] N. Dojer. Learning Bayesian networks does not have to be NP-hard. In *Proceedings of International Symposium on Mathematical Foundations of Computer Science*, pages 305–314, 2006.
- [44] K. Dolinski, A. Chatr-aryamontri, and M. Tyers. Systematic curation of protein and genetic interaction data for computable biology. *BMC Biology*, 11(43), 2013.
- [45] A. J. Enright, S. Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584, 2002.
- [46] L. Franke, H. van Bakel, L. Fokkens, E. D. de Jong, M. Egmont-Petersen, and C. Wijmenga. Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *American Journal of Human Genetics*, 78:1011–1025, 2006.

- [47] R. D.-U. Fátima Al-Shahrour and J. Dopazo. Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. *Bioinformatics*, 21(17):3575–3577, 2005.
- [48] A. Fujita, J. Sato, H. Garay-Malpartida, P. Morettin, M. Sogayar, and C. Ferreira. Time-varying modeling of gene expression regulatory networks using the wavelet dynamic vector autoregressive method. *Bioinformatics*, 23(13):1623–1630, 2007.
- [49] K. Fundel, R. Küffner, and R. Zimmer. RelEx-Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.
- [50] G. Geeven, R. E. van Kesteren, A. B. Smit, and M. C. M. de Gunst. Identification of context-specific gene regulatory networks with gemula gene expression modeling using LASSO. *Bioinformatics*, 28(2):214–221, 2012.
- [51] F. Ginter, S. Pyysalo, J. Björne, J. Heimonen, and T. Salakoski. BioInfer relationship annotation manual. Technical Report 806, Turku Centre for Computer Science, 2007.
- [52] J. J. Goeman and U. Mansmann. Multiple testing on the directed acyclic graph of Gene Ontology. *Bioinformatics*, 24(4):537–544, 2008.
- [53] D. S. Goldberg and F. P. Roth. Assessing experimentally derived interactions in a small world. In *Proceedings of the National Academy of Sciences*, 2003.
- [54] M. Grzegorzcyk and D. Husmeier. Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes. *Bioinformatics*, 27(5):693–699, 2011.
- [55] M. Grzegorzcyk, D. Husmeier, and K. D. Edwards. Modeling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Nucleic Acids Research*, 24(18):2071–2078, 2008.
- [56] F. C. Gunnar Wrobel and M. Primig. goCluster integrates statistical analysis and functional interpretation of microarray expression data. *Bioinformatics*, 21(17):3575–3577, 2005.

- [57] X. Guo, R. Liu, C. D. Shriver, H. Hu, and M. N. Liebman. Assessment of semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics*, 22(8):967–973, 2006.
- [58] H. Hache, H. Lehrach, and R. Herwig. Reverse engineering of gene regulatory networks: A comparative study. *Bioinformatics and Systems Biology*, 2009.
- [59] H. Hache, C. Wierling, and H. Lehrach. Reconstruction and validation of gene regulatory networks with neural networks. In *Foundations of Systems Biology and Engineering*, pages 319–324, 2007.
- [60] U. Hahn, K. Tomanek, E. Buyko, J. jae Kim, and D. Rebholz-schuhmann. How feasible and robust is the automatic extraction of gene regulation events? a cross-method evaluation under lab and real-life conditions. In *Workshop on BioNLP*, pages 37–45, Boulder, Colorado, June 2009.
- [61] T. N. Haixuan Yang and A. Paccanaro. Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, 28(10):1383–1389, 2012.
- [62] A. J. Hartemink, D. K. Gifford, and T. S. Jaakkola. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pacific Symposium on Biocomputing*, 2001.
- [63] M. M. Hasan, N. Noman, and H. Iba. A prior knowledge based approach to infer gene regulatory networks. In *International Conference on Systems Biology*, pages 15–17, Calicut, India., 2010.
- [64] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert. Tigress: Trustful inference of gene regulation using stability selection. *BMC Systems Biology*, 6(145), 2012.
- [65] K. M. Hettne, A. Boorsma, D. A. M. van Dartel, J. J. Goeman, E. de Jong, A. H. Piersma, R. H. Stierum, J. C. Kleinjans, and J. A. Kors. Next-generation text-mining mediated generation of chemical response-specific gene sets for interpretation of gene expression data. *BMC Medical Genomics*, 6(2), 2013.

- [66] M.-G. Hong, Y. Pawitan, P. K. E. Magnusson, and J. A. Prince. Strategies and issues in the detection of pathway enrichment in genome-wide association studies. *Human Genetics*, 126:289–301, 2009.
- [67] D. W. Huang, B. T. Sherman, Q. Tan, J. R. Collins, W. G. Alvord, J. Roayaei, R. Stephens, MichaelWBaseler, H. C. Lane, and R. A. Lempicki. The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biology*, 8(R183), 2007.
- [68] T. Huang, L. Liu, Z. Qian, K. Tu, Y. Li, and L. Xie. Using GeneReg to construct time delay gene regulatory networks. *BMC Research Notes*, 3(142), 2010.
- [69] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.
- [70] M. Huynen, B. Snel, W. Lathe, and P. Bork. Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. *Genome Res.*, 10, 2000.
- [71] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9), 2010.
- [72] S. Imoto, T. Higuchi, and T. Goto. Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, 3(1):1–16, 2006.
- [73] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Conference on Computational Linguistics and Speech Processing*, 1997.
- [74] D. B. Johnson. A note on Dijkstra’s shortest path algorithm. *Journal of the Association for Computing Machinery (ACM)*, 20(3):385–388, 1973.
- [75] W. A. S. Jr., R. M. Raab, and G. Stephanopoulos. Elucidation of gene interaction networks through time-lagged correlation analysis of transcriptional data. *Genome Res*, 14(8):1654–63, 2004.

- [76] S. Kim, J. Kim, and K.-H. Cho. Inferring gene regulatory networks from temporal expression profiles under time-delay and noise. *Computational Biology and Chemistry*, 31:239–245, 2007.
- [77] S. Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235, 2003.
- [78] S. Kimura, S. Nakayama, and M. Hatakeyama. Genetic network inference as a series of discrimination tasks. *Bioinformatics*, 25(7):918–925, 2009.
- [79] H. Lahdesmaki, S. Hautaniemi, and I. Shmulevich. Relationships between probabilistic boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, 86(4):814–834, 2006.
- [80] W. Lam and F. Bacchus. Learning Bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, 10(3):269–293, 1994.
- [81] S. Lèbre, J. Becq, F. Devaux, M. P. Stumpf, and G. Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4, 2010.
- [82] P. Lee and D. Lee. Modularized learning of genetic interaction networks from biological annotations and mRNA expression data. *Bioinformatics*, 21:2739–2747, 2005.
- [83] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 2004.
- [84] F. Li, Y. Yang, and E. Xing. Inferring regulatory networks using a hierarchical Bayesian graphical gaussian. Technical Report CMU-ML-06-117, Carnegie Mellon University, Pittsburgh, PA, 2006.
- [85] L. Li, Y. Wang, and D. Huang. Improving feature-based biomedical event extraction system by integrating argument information. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 109–115, 2013.

- [86] D. Lin. An information theoretic definition in similarity. In *International Conference on Machine Learning*, pages 266–304, 1998.
- [87] W. Liu, K. Chang, J. Zheng, J. Divya, J. jae Kim, and J. C. Rajapakse. Gene regulatory networks from Gene Ontology. In *International Symposium on Bioinformatics Research and Applications*, pages 87–98, 2013.
- [88] W. Liu, K. Miao, G. Li, K. Chang, J. Zheng, and J. C. Rajapakse. Extracting rate changes in transcriptional regulation from MEDLINE abstracts. *BMC Bioinformatics*, 15(Suppl 2), 2014.
- [89] K. Lo, A. E. Raftery, K. M. Dombek, J. Zhu, E. E. Schadt, R. E. Bumgarner, and K. Y. Yeung. Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. *BMC Systems Biology*, 6(101), 2012.
- [90] P. Lord, R. Stevens, A. Brass, and C.A.Goble. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19:1275–1283, 2003.
- [91] Z. Lubovac and B. Olsson. Towards reverse engineering of genetic regulatory networks. Technical Report HS-IDA-TR-03-003, University of Skövde, Sweden, 2003.
- [92] S. R. Maetschke, M. Simonsen, M. J. Davis, and M. A. Ragan. Gene ontology-driven inference of protein–protein interactions using inducers. *Bioinformatics*, 28(1):69–75, 2012.
- [93] R. Maithreye, R. R. Sarkar, V. K. Parnaik, and S. Sinha. Delay-induced transient increase and heterogeneity in gene expression in negatively auto-regulated gene circuits. *PLoS ONE*, 3(8), 2008.
- [94] T. Manoli, N. Gretz, H.-J. Gröne, M. Kenzelmann, R. Eils, and B. Brors. Group testing for pathway analysis improves comparability of different microarray datasets. *Bioinformatics*, 22(20):2500–2506, 2006.

- [95] D. Marbach, J. C. Costello, R. Küffner, N. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, the DREAM5 Consortium, M. Kellis, J. J. Collins, and G. Stolovitzky. Wisdom of crowds for robust gene network inference. *Nat Methods*, 9(8):796–804, 2012.
- [96] L. Matthews, G. Gopinath, M. Gillespie, M. Caudy, D. Croft, B. de Bono, P. Garapati, J. Hemish, H. Hermjakob, and B. Jassal. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res*, 37(Database issue):D619–622, 2009.
- [97] C. V. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel. STRING: a database of predicted functional associations between proteins. *Nucleic Acids Res*, 31(1):258–261, 2003.
- [98] M. Miwa, R. Satre, Y. Miyao, and J. Tsujii. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 788–796, 2010.
- [99] A. Moreno, D. Isern, and A. C. L. Fuentes. Ontology-based information extraction of regulatory networks from scientific articles with case studies for escherichia coli. *Expert Systems with Applications*, 40:3266–3281, 2013.
- [100] A. A. Motsinger, S. L. Lee, G. Mellick, and M. D. Ritchie. GPNN: Power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. *BMC Bioinformatics*, 7(39), 2006.
- [101] P. A. Mundra and J. C. Rajapakse. SVM-RFE with MRMR filter for gene selection. *IEEE Transactions on Nanobiosciences*, 9, 2010.
- [102] P. A. Mundra, J. Zheng, N. Mahesan, R. E. Welsch, and J. C. Rajapakse. Inferring time-delayed gene regulatory networks using cross-correlation and sparse regression. In *International Symposium on Bioinformatics Research and Applications*, pages 64–75, 2013.

- [103] N. Nariai, S. Kim, and S. Imoto. Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. In *Pacific Symposium on Biocomputing*, volume 9, pages 336–347, 2004.
- [104] W. S. Noble. Stability of a hierarchical clustering. *Nat Biotechnol.*, 27(12):1135–1137, 2009.
- [105] N. Noman and H. Iba. Inference of genetic networks using s-system: Information criteria for model selection. In *GECCO*, Seattle, Washington, USA, 2006.
- [106] S. Novichkova, S. Egorov, and N. Daraselia. MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, 19:1699–1706, 2003.
- [107] D. A. Orlando, C. Y. Lin, A. Bernard, J. Y. Wang, J. E. S. Socolar, E. S. Iversen, A. J. Hartemink, and S. B. Haase. Global control of cell-cycle transcription by coupled cdk and network oscillators. *Nature*, 453:944–947, 2008.
- [108] L. Page, S. Brin, and R. Motwani. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [109] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stumpflen, and H. Mewes. The MIPS mammalian protein-protein interaction database. *Bioinformatics*, 21(6):832–834, 2005.
- [110] J. Pandey, M. Koyutürk, S. Subramaniam, and A. Grama. Functional coherence in domain interaction networks. *Bioinformatics*, 24:i28–i34, 2008.
- [111] T. H. Pers, N. T. Hansen, K. Lage, P. Koefoed, P. Dworzynski, M. LeeMiller, T. J. Flint, E. Mellerup, H. Dam, O. A. Andreassen, S. Djurovic, I. Melle, A. D. Børghlum, T. Werge, S. Purcell, M. A. Ferreira, I. Kouskoumvekaki, C. T. Workman, T. Hansen, O. Mors, and S. Brunak. Meta-analysis of heterogeneous data sources for genome-scale identification of risk genes in complex phenotypes. *Genetic Epidemiology*, 35:318–332, 2011.
- [112] T. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal. Human protein reference database–2009 update. *Nucleic Acids Res*, 37(Database issue):D767–772, 2009.

- [113] S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50), 2007.
- [114] J. Rajapakse and I. Chaturvedi. Gene regulatory networks with variable-order dynamic Bayesian networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, 2010.
- [115] S. A. Ramsey, S. L. Klemm, D. E. Zak, K. A. Kennedy, V. Thorsson, B. Li, M. Gilchrist, E. S. Gold, C. D. Johnson, V. Litvak, G. Navarro, J. C. Roach, C. M. Rosenberger, A. G. Rust, N. Yudkovsky, A. Aderem, and I. Shmulevich. Uncovering a macrophage transcriptional program by integrating evidence from motif scanning and expression dynamics. *PLoS Comput Biol*, 4(3), 2008.
- [116] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conferences on Artificial Intelligence*, pages 448–453, 1995.
- [117] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. In *JAIR*, volume 11, pages 95–130, 1999.
- [118] S. Y. Rhee, V. Wood, and K. Dolinski. Use and misuse of the gene ontology annotations. *Nature Reviews Genetics*, 9:509–515, 2008.
- [119] A. J. Richards, B. Muller, M. Shotwell, L. A. Cowart, B. Rohrer, and X. Lu. Assessing the functional coherence of gene sets with metrics based on the gene ontology graph. *Bioinformatics*, 26:i79–i87, 2010.
- [120] J. Robinson and A. J. Hartemink. Learning non-stationary dynamic Bayesian networks. *Journal of Machine Learning Research*, 11:3647–3680, 2010.
- [121] N. Rosenfeld and U. Alon. Response delays and the structure of transcription networks. *Journal of Molecular Biology*, 329(4):645–654, 2003.

- [122] H. Salgado, S. Gama-Castro, M. Peralta-Gil, and E. Diaz-Peredo. RegulonDB (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34(Database issue):394–397, 2006.
- [123] L. Salwinski, C. Miller, A. Smith, F. Pettit, J. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Res*, 32(Database issue):D449–451, 2004.
- [124] J. Saric, L. Jensen, R. Ouzounova, I. Rojas, and P. Bork. Extraction of regulatory gene/protein networks from medline. *Bioinformatics*, 22(6):645–650, 2006.
- [125] A. Schlicker, F. S. Domingues, J. Rahnenfuhrer, and T. Lengauer. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, 2006.
- [126] C. Sima, J. Hua, and S. Jung. Inference of gene regulatory networks using time-series data: A survey. *Current Genomics*, 10(6):416–429, 2009.
- [127] S. P. Smith and R. Dubes. Stability of a hierarchical clustering. *Pattern Recognition*, 12:177–187, 1980.
- [128] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. In *Proceedings of the National Academy of Sciences*, volume 100, pages 12123–8, 2003.
- [129] C. Stark, B. Breitkreutz, T. Regul, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*, 34(Database issue):D535–539, 2006.
- [130] R. Steuer, P. Humburg, and J. Selbig. Validation and functional annotation of expression-based clusters based on gene ontology. *BMC Bioinformatics*, 2006.
- [131] S. Suthram, T. Shlomi, and E. Ruppin. A direct comparison of protein interaction confidence assignment schemes. *BMC Bioinformatics*, 7(360), 2006.

- [132] T. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, Suppl 2:i227–36, 2003.
- [133] M. Tan, M. Alshalalfa, R. Alhajj, and F. Polat. Influence of prior knowledge in constraint-based learning of gene regulatory networks. *IEEE/ACM Trans Comput Biol Bioinform.*, 8(1):130–142, 2011.
- [134] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.
- [135] M. C. Teixeira, P. Monteiro, and P. Jain. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *saccharomyces cerevisiae*. *Nucl. Acids Res.*, 34:446–451, 2006.
- [136] Z. Teng, M. Guo, X. Liu, Q. Dai, C. Wang, and P. Xuan. Measuring gene functional similarity based on group-wise comparison of go terms. *Bioinformatics*, 29(11):1424–1432, 2013.
- [137] O. G. Troyanskaya, K. Dolinski, and A. B. Owen. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *saccharomyces cerevisiae*). In *Proceedings of the National Academy of Sciences*, volume 100, pages 8348–8353, 2003.
- [138] K. Tu, H. Yu, and Y.-X. Li. Combining gene expression profiles and protein-protein interaction data to infer gene functions. *Journal of Biotechnology*, 124:475–485, 2006.
- [139] J. J. Tyson, K. Chen, and B. Novak. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology*, 2:908–916, 2001.
- [140] P. Vertino, E. Spillare, C. Harris, and S. Baylin. Altered chromosomal methylation patterns accompany oncogene-induced transformation of human bronchial epithelial cells. *Cancer Research*, 53(7), 1993.

- [141] A.-L. Veuthey, A. Bridge, J. Gobeill, P. Ruch, J. R. McEntyre, L. Bougueleret, and I. Xenarios. Application of text-mining for updating protein post-translational modification annotation in UniProtKB. *BMC Bioinformatics*, 14(104), 2013.
- [142] M. Vignes, J. Vandel, D. Allouche, N. Ramadan-Alban, C. Cierco-Ayrolles, T. Schiex, B. Mangin, and S. de Givry. Gene regulatory network reconstruction using Bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PLoS ONE*, 6(12), 2011.
- [143] N. X. Vinh, M. Chetty, R. Coppel, and P. P. Wangikar. GlobalMIT: learning globally optimal dynamic Bayesian network with the mutual information test criterion. *Bioinformatics*, 27(19):2765–2766, 2011.
- [144] N. X. Vinh, M. Chetty, R. Coppel, and P. P. Wangikar. Gene regulatory network modeling via global optimization of high-order dynamic Bayesian network. *BMC Bioinformatics*, 13, 2012.
- [145] H. Wang, L. Qian, and E. Dougherty. Inference of gene regulatory networks using s-system: a unified approach. *IET Systems Biology*, 4(2):145–156, 2010.
- [146] H. Wang, H. Zheng, F. Browne, D. H. Glass, and F. Azuaje. Integration of Gene Ontology-based similarities for supporting analysis of protein–protein interaction networks. *Pattern Recognition Letters*, 31:2073–2082, 2010.
- [147] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen. A new method to measure the semantic similarity of GO termss. *Bioinformatics*, 23:1274–1281, 2007.
- [148] A. V. Werhli and D. Husmeier. Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. *Journal of Bioinformatics and Computational Biology*, 6(3):543–572, 2008.
- [149] H. Wu, Z. Su, F. Mao, V. Olman, and Y. Xu. Prediction of functional modules based on comparative genome analysis and Gene Ontology application. *Nucleic Acids Res*, 33(9):2822–2837, 2005.

- [150] X. Wu and Y. Ye. Exploring gene causal interactions using an enhanced constraint-based method. In *Pattern Recognition*, volume 39, pages 2439–2449, 2006.
- [151] X. Wu, L. Zhu, J. Guo, D. Zhang, and K. Lin. Prediction of yeast protein–protein interaction network: insights from the Gene Ontology and annotations. *Nucleic Acids Res*, 34(7):2137–2150, 2006.
- [152] Z. Xing and D. Wu. Modeling multiple time units delayed gene regulatory network using dynamic Bayesian network. In *ICDM Workshops*, pages 190–195, 2006.
- [153] T.-H. Yang, C.-C. Wang, Y.-C. Wang, and W.-S. Wu. YTRP: a repository for yeast transcriptional regulatory pathways. *Database*, 2014, 2014.
- [154] C.-H. Yeang, H. C. Mak, S. McCuine, C. Workman, T. Jaakkola, and T. Ideker. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology*, 6, 2005.
- [155] K. Y. Yip, R. P. Alexander, K.-K. Yan, and M. Gerstein. Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PLoS ONE*, 5(1), 2010.
- [156] A. Yuryev, Z. Mulyukov, and E. Kotelnikova. Automatic pathway building in biological association networks. *BMC Bioinformatics*, 7, 2006.
- [157] S. Zambach and T. Lassen. A lexical framework for semantic annotation of positive and negative regulation relations in biomedical pathways. In *Semantic Mining in Biomedicine*, Hinxton, United Kingdom, Oct. 2010.
- [158] B. Zheng and X. Lu. Novel metrics for evaluating the functional coherence of protein groups via protein semantic network. *Genome Biology*, 8(7), 2007.
- [159] X. H. Zhou, M. J. Kao, and W. H. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. In *Proceedings of the National Academy of Sciences*, volume 99, pages 12783–12788, 2002.

- [160] P. Zoppoli, S. Morganella, and M. Ceccarelli. TimeDelayed-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(154), 2010.
- [161] M. Zou and S. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21, 2005.