Gene Network Inference using Machine Learning and Graph Algorithms on Big Biomedical Data

Migao Wu

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2016

Reading Committee:

Ke Yee Yeung, Chair

Ling-Hong Hung

Ankur Teredesai

Program Authorized to Offer Degree:

Institute of Technology - Tacoma

University of Washington

Abstract

Gene Network Inference using Machine Learning and Graph Algorithms on Big Biomedical Data

Migao Wu

Chair of the Supervisory Committee:

Associate Professor Ke Yee Yeung

Institute of Technology, University of Washington, Tacoma

Gene networks capture the interactions between different biological entities. These gene networks have many applications in modern day biology. In particular, gene networks can help to shed light on the underlying mechanisms of diseases. Advances in biotechnology have led to the generation of different types of genome-wide data, profiling the activity levels across the entire genome. In this thesis, we generated informative and accurate gene networks by integrating multiple types of big biomedical data.

Many algorithms have been proposed in the literature to infer gene networks from genome-wide data. However, it is non-trivial to distinguish direct edges between two nodes from indirect edges represented by a path connecting two nodes using these genome-wide data. In this thesis, I constructed compact and accurate gene networks by using an improved Bayesian Modeling Averaging based gene network inference algorithm which includes a post-processing step of removing indirect redundant edges. I applied this improved method to synthetic data in which the ground truth was already known and to real data in which external data sources were used to help assess and analyze the resulting gene networks. The

assessment results were presented in two different forms, graphs and tables. In general, the results showed that the new gene network inference algorithm produced more accurate networks and the implementation is more efficient.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

DNA: deoxyribonucleic acid; the molecule that encodes genetic information.

GENE: a segment of DNA which normally specifies a functional unit.

GENE EXPRESSION: the process by which a gene's coded information is converted to the structures present and operating in the cell.

LINCS: library of integrated network-based cellular signatures.

BD2K: the Big Data to Knowledge

NIH: National Institutes of Health

CELL LINE: a population of cells descended from a single cell and containing the same genetic makeup.

TRANSCRIPTOME: is the full range of messenger ribonucleic acid (RNA), or mRNA, molecules expressed by an organism.

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to his advisor, Associate Prof. Ka Yee Yeung, for her patience and guidance. He would also like to acknowledge many other faculty members, colleagues and graduate students who helped to shape his research: Dr. Ling-Hong Hung, Prof. Adrian Raftery, Prof. Ankur Teredesai, Mr. William Chad Young, Mr. Yunhong Yin, Mr. Qi Wei, Mr. Azu Lee and Mr. Kaiyuan Shi. Finally, this Master's thesis work would not be possible without the love and encouragement from his parents.

DEDICATION

to my Master program study

Chapter 1

INTRODUCTION

1.1 Introduction

Big data is ubiquitous in modern day biology. The European Bioinformatics Institute (EBI) is one of the world's largest biology data repositories which currently stores 20 petabytes data of molecules, proteins and genes [1]. Most importantly, the rate of increment of data stored each year is rising. The complexity when dealing with biological big data is very high [2]. Advances in biotechnology allow the generation of multiple types of genome-wide data. These genome-wide data profiling measurements across biological entities across the entire genome provide a global view of molecular systems. Examples of these genome-wide technology include microarrays [3, 4, 5], sequencing [6] and mass spectrometry [7]. Integrating multiple sources of genome-wide data to extract biological meanings from these abundant sources of data is a major challenge in bioinformatics [8].

Systems biology studies the interactions between different components of biological systems and how these interactions affect the behaviors within that biological system. These interactions (or gene networks) are typically represented as graphs, in which the nodes represent genes and the edges (directed or undirected) represent relationships between genes.

Gene networks are important in the development, differentiation and responding to environmental cues [9]. Also, they facilitate the understanding of the biological functioning of cells. Identifying the interactions between genes can aid biologists to understand how the cell functions both in reaction to external stimuli and steady state [10]. Modern day biological research is driven by the desire of coming up with new treatments to cure diseases.

Studying disease patterns is a key step to tentatively solve this problem. Gene networks represent complex relationships between biological entities which help us to identify putative driver and passenger genes in various diseases [11].

Gene networks have many applications in biomedical sciences. In particular, gene networks capture molecular interactions. We can statistically compare gene networks from different physiological and disease conditions which allows us to learn more about the interaction changes across different physiological or disease conditions [12, 13, 14, 15]. Usually, the large-scale generation and integration of genomic, proteomic, signalling and metabolomic data are increasingly allowing the construction of complex networks that provide a new framework for understanding the molecular basis of physiological or pathophysiological states. Network-based drug discovery aims to harness this knowledge to investigate and understand the impact of interventions, such as candidate drugs, on the molecular networks that define these states. Subsequently, these advances could lead to improved therapies [16, 17].

Visualizing and analyzing gene networks have become important tasks to accomplish. Personalized medicine is an important application of gene networks [17]. Specifically, common human diseases originate from complex interplay between constellations of changes in Deoxyribonucleic acid (DNA) (both rare and common variations) and a broad range of factors such as diet, age, gender and exposure to environmental toxins. These complex arrays of interacting factors are thought to affect entire network states that in turn increase or decrease the risk of disease or affect disease severity. The disease states can be considered emergent properties of molecular networks [18]. If we advance our knowledge in molecular networks, then we can in turn reduce the risk of disease more significantly [19].

With the application of systems biology approaches to big biological data derived from various diseased states, a new research area called systems medicine has been developed. Systems medicine unites genomics and genetics to identify disease genes. P4 medicine, which stands for predictive, preventive, personalized, and participatory, is the product of the convergence of patient-activated social networks, big data and their analytics, and systems

medicine [20]. In the context of bioinformatics, molecular networks play an important role. Molecular networks reflect DNA and environmental perturbations and, as a result, drive variations in physiological states associated with diseases [18]. Molecular networks capture the relationships between biological entities (such as genes and proteins) under different experimental conditions (such as drug perturbations). Therefore, researchers could use molecular networks in P4 medicine to define the downstream effect of drugs as the genes related to the known drug targets in gene networks.

Many software have been developed to help answer systems biology research questions. In particular, Cytoscape is a publicly available bioinformatics software tool for the integration, visualization, and exploration of biological networks. Cytoscape provides functionality for data import and export, integration of molecular states with molecular interactions, network and integrated data visualization, and data filtering and query tools [21, 22].

In addition, machine learning and data science techniques have been used in the analyses and construction of gene networks using big biological data. Machine learning methods that have been applied to these applications, include supervised classification, unsupervised clustering, linear regression, logistic regression and probabilistic graphical models and networks for knowledge discovery, as well as deterministic and stochastic heuristics for optimization [23, 24].

In this thesis, I experimented and assessed gene network construction algorithms called fast Bayesian Model Averaging (fastBMA) that combines machine learning and graph theory techniques to construct accurate and compact gene networks. In particular, fastBMA extended the regression-based gene network inference methods developed by Drs. Ka Yee Yeung, Dr. Adrian Raftery, Dr. Ling-Hong Hung and Mr. William Chad Young [25, 26, 27, 28] by providing an efficient and optimized algorithm as well as removing redundant edges using transitive reduction. My empirical experiments use both time series and static (no time point) gene expression data.

1.2 Related Work

In this subsection, we review methods for the inference of gene networks in the literature.

1.2.1 Correlation-based

Correlation-based methods are the most intuitive. In a correlation or co-expressed network, nodes represent genes and nodes are connected if the corresponding genes are significantly correlated across appropriately chosen tissue samples [29]. Co-expression networks aim to find regulatory relationships between genes using correlation. They are typically generated using correlation statistics as pairwise similarity measures. An advantage of correlation-based methods is that they are extremely useful in order to determine whether two genes have a strong global similarity over all conditions from the data set.

1.2.2 Bayesian networks

Bayesian networks are a class of graphical probabilistic models. A Bayesian network consists of an annotated directed acyclic graph (DAG), where the nodes are random variables representing genes' expressions and the edges indicate the dependencies between the nodes. Friedman *et al.* used Bayesian networks to establish regulatory relationships between genes in yeast using time-series gene expression data [30]. The same group of authors also applied Bayesian networks to perturbation gene expression data to identify regulatory relationships and in addition, to predict their nature of activation or inhibition [31]. Bayesian networks offer an probabilistic and intuitive framework in which to model and reason about qualitative properties of gene networks. The problem with Bayesian networks is if the graph model is not known then the space of all graph models has to be explored. However, this space is super exponential even for directed acyclic graphs and exploring it completely is impossible even with the fastest heuristics [32]. Subsequently, Bayesian networks are highly

computationally intensive, and hence, they can be applied only when the network size is small.

A key constraint with Bayesian networks is that a DAG is assumed, hence cycles are not allowed. We need to accommodate cycles due to potential feedback loops in networks inferred from time series data. Dynamic Bayesian networks (DBN) have been applied to time series gene expression data [33]. Given the time series data, DBN works by estimating a probabilistic graphical model. DBN works well in terms of the precision of the generated gene regulatory networks. However, it works well only when the network size is small. In another way of saying, it is limited to large network size due to its computational cost. The difference between Bayesian networks and DBN is that DBN also captures temporal relationships between variables X_t which is the vector for variables X at time point t [34]. Therefore, in DBN with time points, DAG can't be used since there are cycles in the graphs. However, Bayesian networks can have DAGs.

1.2.3 Ordinary Differential Equations (ODE)

Ordinary differential equations (ODE) could also be combined with statistical methods [35]. As an example, Network identification by multiple regression (NIR) uses wild-type gene expression data and does not require prior information about each gene's function or network structure, but only the information of which genes are directly perturbed in each experiment. The most significant advantage of NIR is that it performs with high accuracy in sparse gene networks, though the resulting networks usually lack directionality [36].

1.2.4 Graph Theoretical Models

Graph theoretical models (GTMs) are used mainly to describe the topology, or architecture, of a gene network. Inferring gene networks under GTMs amounts to identify the edges and their parameters from given expression data. Using both time-series measurements and perturbation measurements of gene expression are feasible choices of data types. GTMs

are very useful for knowledge representations but not simulation. Existing gene network inference methods use parsimonious assumptions about the nature of the networks to reduce the solution space and yield a single solution [32].

1.2.5 Regression-based method for inferring gene networks

In regression-based approaches, parent nodes (regulators) are inferred for each target gene using a regression framework. In the case of time series data, the expression level at the previous time point to predict the expression levels at the current time point. Without prior knowledge, every gene is a potential regulator of ever other gene. Since there are usually lots of potential regulators, prior probabilities are computed using other data sources to constrain the search space to the most likely regulators. Also, regression-based methods can account for multiple data sources through the use of prior information in a Bayesian statistical framework.

The regression framework can be formulated as a statistical variable selection or model selection problem. Vector auto-regressive models have been proposed for inferring causal links between genes. Least Absolute Shrinkage and Selection Operator (LASSO) [37], elastic net [38], and Bayesian Model Averaging (BMA) [39, 40] are some of the algorithms in this regression-based gene inferring category. BMA will be explained in details in Chapter 2.

1.3 Overview of chapters

Chapter 2 gives detailed explanations of Bayesian Model Averaging (BMA) gene network inference methods (including iterative BMA (iBMA), Scan BMA (ScanBMA), and fast BMA (fastBMA)). We will also introduce the concepts of transitive reduction (TR) and how to integrate transitive reduction to work with BMA. Also, Chapter 2 demonstrates the application of fastBMA on simulated Dialogue for Reverse Engineering Assessments and Methods (DREAM) time series data and yeast real time series data. Chapter 3 discusses the

Library of Integrated Network-based Cellular Signatures (LINCS) L1000 data, and explains how to adapt Bayes Factor to fastBMA. Chapter 3 also demonstrates the application of fastBMA on different types of LINCS perturbation data without time points. Finally, Chapter 4 is the conclusion of this thesis project.

Chapter 2

GENE NETWORK INFERENCE USING TIME SERIES DATA

This chapter focuses on the inference of gene networks using time series gene expression data. Also, this chapter covers concepts that will be used throughout this thesis, including detailed descriptions of the Bayesian Model Averaging (BMA) gene network inference methods (iterative BMA (iBMA), Scan BMA (ScanBMA), and fast BMA (fastBMA)), and a graph theoretical concept that removes direct spurious relations. Both simulated and real gene expression data sets were used to assess our methods. We will also introduce assessment criteria and statistics.

2.1 Background

Time series data usually consists of successive measurements made over multiple time points. These temporal data allow us to observe the pattern of changes from one time point to the next time point for each gene. Time series expression experiments have been used to study a wide range of biological systems. One of the unique features of time series experiments is the ability to infer the relationship "If A changes, then B will change". Yeung *et al.* aimed to generate testable hypotheses of gene-to-gene influences and subsequently design bench experiments to confirm these network predictions [26]. Time-series data and genetics data were used to infer the directionality of edges in regulatory networks. Time-series data contain information about the chronological order of regulatory events. Another example of using time-series data to infer gene networks is Bansal *et al.*, who inferred the local network of gene-gene interactions by perturbing a gene of interest and subsequently measuring the gene expression profiles at multiple time points. Bansal *et al.* showed that it is possible to recover the gene regulatory gene networks from a time-series data of gene expression

following a perturbation to the cell [36].

Many statistical methods have been developed for time series data analysis, and these methods could be adapted to infer gene networks from time series data. For example, Wichert *et al.* proposed a method for signal detection and gene selection in gene expression time series data [41]. Another example is Short Time-series Expression Miner (STEM) [42] which was designed for the analysis of short time series microarray gene expression data consisting of relatively few time points. STEM implements statistical methods to cluster, compare, and visualize such data. STEM also supports efficient and statistically rigorous biological interpretations of short time series data through its integration with the Gene Ontology annotations [42].

An auto regressive model is when a value from a time series is regressed on previous values from that same time series. For example, considering a problem in which we have a y-variable measured as a time series. Y is a measure of gene expression, with measurements observed each hour. We use t as a subscript to emphasize that we have measured gene expression over time. So, y_t means y measured in time period t. An auto regressive model of this example could be y_t on y_{t-1}:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$$

. In this regression model, the response variable in the previous time period has become the predictor and the errors have the usual normal assumptions about errors in a simple linear regression model. With this model, we could predict (y_t) using time series data [43].

2.2 Regression-based methods to infer gene networks from time series data

Given time series gene expression data across multiple time points, the regression-based approach used the gene expression level at the previous time point $(t-1)$ to predict the expression level at the current time point t in the same experiment. Without prior knowledge, every gene is a potential regulator of every other gene, Yeung and colleagues used prior probabilities computed from other data sources to constrain the search to the most

likely regulators. This regression approach is illustrated in Figure 2.1 [26], in which there are many potential regulators R in the previous time point $(t-1)$ that can regulate the expression level of gene g at the current time point t. We aim to identify potential regulators R that regulate gene g. Mathematically, the expression of each gene is predicted by a linear combination of the expression of candidate regulators at the previous time point [25, 26, 28]:

$$X_{i,t} = \beta_{0,i} + \sum_{h \epsilon H} \beta_{h,i} X_{h,t-1} + \epsilon_{i,t}$$

where $X_{i,t}$ is the expression of gene i at time t, H is the group of regulators for gene i in a candidate model, β's are the regression coefficients, and $\epsilon_{i,t}$ is the error term for gene $i = 1...n$ and time $t = 2, ...T$.

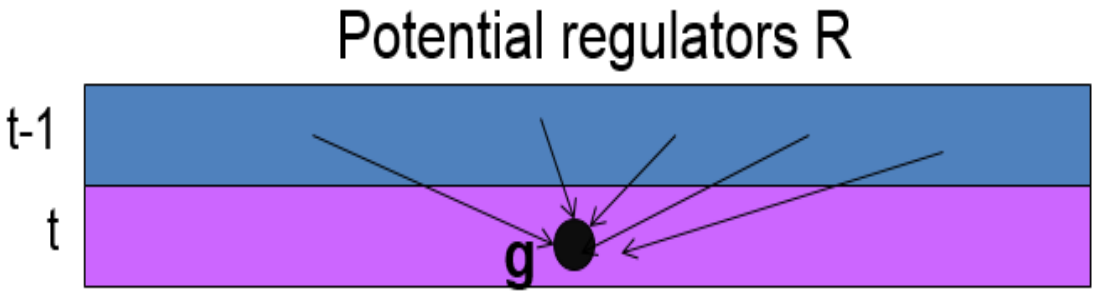


Figure 2.1: Regression-based approach on time series data

2.2.1 Bayesian Model Averaging

Bayesian Model Averaging (BMA) is a variable selection and network inference algorithm that takes model uncertainty into account by averaging over the posterior distribution of a quantity of interest based on multiple models, weighted by their posterior model probabilities [40, 44].

BMA used the leaps and bounds algorithm [45] to efficiently identify a small group of promising models out of all possible models. Specifically, it returns the $n$best models for each number of variables (regulators) and Occam's window was used to discard models with

much lower posterior model probabilities than the best one [46]. The Bayesian Information Criterion (BIC) [47] was used to approximate each model's integrated likelihood, from which its posterior model probability could be determined. BIC corrects for over-fitting with a penalty term for the number of parameters in the model.

This version of BMA cannot be applied to high-dimensional data that contain more variables than samples. Additionally, the leaps and bounds algorithm scaled poorly and was limited in practice to fewer than 50 variables. In order to fix this high dimensionality issue, Yeung *et al.* developed the iterative Bayesian Model Averaging (iBMA) algorithm [26, 48] which iteratively called BMA. In iBMA, a pre-processing step was used to rank all variables using a univariate measure. Then, the original BMA was iteratively applied to the top w variables (where w is the window size and is typically set to 30). Predictor variables with low posterior probabilities are discarded, and replaced by new variables from the ranked list. This process of repeatedly applying BMA and swapping variables was continued until the specified p top ranked variables are processed [26, 48].

2.2.2 Scan Bayesian Model Averaging

Scan Bayesian Model Averaging (ScanBMA) is an improved algorithm for gene network inference using time series expression data [28]. There are three major differences between ScanBMA and iterative Bayesian Model Averaging (iBMA). The first difference is that ScanBMA removed the pre-processing step in which variables are ranked using a univariate measure. ScanBMA is a greedy algorithm in which a single variable is considered to be added or removed to improve the best models found so far. Occam's window was used to determine whether the new model should be kept. The process repeated until no new models were added or removed from the best set of models. The second difference between ScanBMA and iBMA is that ScanBMA allowed the use of Zellner's g-prior [49], which replaced the Bayesian Information Criterion (BIC) for scoring the models. Zellner's g-prior was more flexible than BIC and could be either specified or estimated using the Expectation-Maximization (EM) algorithm. The third difference is that there is no upper limit on the

maximum size of the models inferred by ScanBMA since the fixed window size is removed. In contrast to iBMA, in which the number of maximum variables was limited by the BMA window size w.

2.3 Fast Bayesian Model Averaging

Fast Bayesian Model Averaging (fastBMA) constructed accurate and compact gene networks by using a combination of statistical and graph theory techniques. It is a distributed, parallel and scalable Bayesian type inference method that uses Zellner's g-prior [49] to guide the search in model space [50]. The first phase of fastBMA used the BMA regression-based gene network inference framework [25, 26, 28, 51]. The second post-processing phase used the idea of transitive reduction (TR) to remove indirect influence edges. fastBMA reduced the time complexity and was implemented in C++ [50]. We showed that our fastBMA method out-performed previous methods using both simulated and real time series gene expression data. In particular, fastBMA improved upon its fast predecessor Scan Bayesian Model Averaging (ScanBMA) [28] by increasing the speed by more than 500-fold and supported multiple threads. A 100 gene network was obtained in $0.1s$ and a complete 10,000 gene network could be obtained in hours [50]. When using a single thread, fastBMA is 30x faster than ScanBMA [28] and 10x-1000x faster than the Least Absolute Shrinkage and Selection Operator (LASSO) [37] with increased accuracy.

2.3.1 Optimized algorithm and implementation

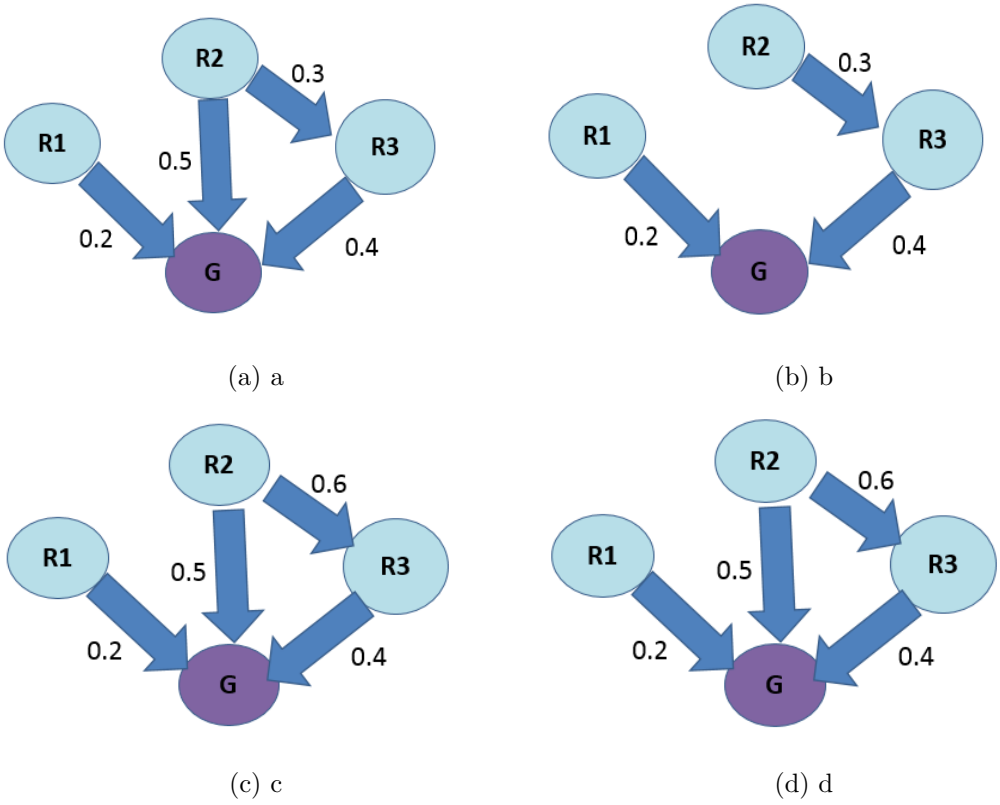
The first step formulated gene network inference as a variable selection problem in which parent nodes were selected for each target gene. fastBMA implemented a more scalable and optimized version of the ScanBMA algorithm. fastBMA took advantage of the fact that new models were always based upon existing models where the regression coefficients have already been calculated. fastBMA updated the models rather than calculating the entire regression ***de novo***. The gain of speed came from a more efficient linear regression

algorithm, faster optimized Open Basic Linear Algebra Subprogram (OpenBLAS) which is a linear algebra library routine [52], and replacing all the R routines with C++ code [50]. The faster speed allowed fastBMA to sample the model space in a more comprehensive fashion, hence increasing the accuracy of inferred models.

2.3.2 Transitive reduction

Transitive reduction (TR) removes such spurious relations if they can be explained by an indirect path [53]. A gene network usually contains edges that represent either direct or indirect interactions. TR can also be defined as the process of removing indirect influences. The resulting networks are more compact and intuitive. Therefore, it is desirable to remove edges between nodes where the regulation is indirect. fastBMA used TR to post-process the inferred gene networks from step 1 [50].

TR compared the measured influence strengths (uncertainty) between the direct and indirect interactions. The direct interaction is removed if its uncertainty is greater than the largest uncertainty of the indirect path [53]. The result of TR is that a direct interaction is removed only if there exists an indirect interaction path between the same nodes which is more certain than the direct one. In other words, the largest uncertainty of the indirect path is smaller than the uncertainty of the direct interaction. Figure 2.2 shows a small graph example of TR.



(a) a (b) b

(c) c (d) d

Figure 2.2: In panel (a), the uncertainty value of the direct edge from R2 to G is 0.5. There is also an indirect path from R2 to G which is R2 to R3 to G with the largest uncertainty equals to 0.4. Since 0.5 is greater than 0.4 which means the direct edge is more uncertain than the indirect path, the direct edge from R2 to G is removed which is shown in panel (b). In panel (c), the uncertainty of the direct edge from R2 to G is still 0.5. However, the indirect path from R2 to G which is R2 to R3 to G with the largest uncertainty equals to 0.6. Since 0.6 is greater than 0.5 which means the indirect path is more uncertain than the direct edge, the direct edge is remained which is shown in panel (d).

In BMA, iBMA, ScanBMA and fastBMA, the linear regression step evaluates correlations between the observation and the variable and is agnostic to whether the correlation is due to a direct interaction or an indirect interaction. However, the Bayesian framework allows for the specification of prior probabilities that indicates the likelihood of a direct

interaction between the parent node and the child node. This biases the formation of edges that represent direct interactions. A direct edge may still be inferred between two genes even when the actual regulatory path between these two genes is an indirect path [50]. Researchers could calculate the joint probability of any path by multiplying the probabilities together assuming all the probabilities along the path are independent.

Hung *et al.* extended the TR idea and added this as a post-processing step in fastBMA. Instead of using measured influence strength (uncertainty) to represent each edge, posterior probability resulted from BMA was used the to represent each edge in gene networks. In addition, instead of comparing the uncertainties in indirect paths and direct paths to decide whether a direct edge should be removed or preserved, the logarithmic transformation was applied to these posterior probabilities so that one could simply add the probabilities of all the edges in a path to get the probability of the entire path since multiplication is generally more expensive than addition in terms of running time. Subsequently, the exponentiation was applied to the sum of all the log transformed probabilities to compute the probability of the entire path. This exponentiation step could be skipped at the end if one just wants to know which path is more certain. Determining whether a better indirect path exists between two connected nodes (*i.e.* lower negative log sum) was formulated as the shortest path problem which has known efficient solutions. For example, Dijkstra's algorithm [54] that solves the shortest path problem in $O(n \log n + E \log n)$ time which E is the number of edges and n is the number of nodes in the gene network.

2.3.3 Input parameters in fastBMA

We tuned the following parameters when testing the fastBMA method. Number of variables means how many variables (genes) are selected from a model. *w/prior* means that external priors files are been used with fastBMA. *w/oprior* means that we did not use any priors files when generating networks using fastBMA. w/TR means that transitive reduction has been applied to the gene networks generated by fastBMA. w/oTR means that transitive reduction has not been applied to the gene networks generated by fastBMA. In other words,

we did not try to prune redundant edges in the networks.

2.4 Data

I applied the fastBMA algorithm to both simulated and real time series gene expression data. Simulated time series gene expression data have the advantage that the ground truth is known. The real time series gene expression data consist of a higher number of variables (genes), contain noise from the experimental measurements, and hence, are closer to the intended use cases for fastBMA. However, we were uncertain with the ground truth of regulatory relationships due to incomplete knowledge.

2.4.1 Simulated Data from the Dialogue for Reverse Engineering Assessments and Methods (DREAM) Challenge

For simulated data sets, we used the in-silico 10-gene and 100-gene time-series data over 21 time points and the corresponding reference networks from the DREAM4 challenge which many different teams participated to infer gene networks [55, 56, 57, 58, 59].

2.4.2 Real Yeast Data

For real time series data sets, we used a subset of the yeast time-series gene expression data consisting of 3556 genes over 6 time points [26] and the literature-curated regulatory relationships from the Yeast Search for Transcriptional Regulators And Consensus Tracking (YEASTRACT) database [60] as our assessment criteria. In this thesis, we use the terms "assessment criteria" and "gold standard" interchangeably. This yeast time series expression data measure the response to a drug perturbation over 6 time points at 10-minute intervals in 95 yeast segregants and 2 parental strains. The full data sets are publicly available from the ArrayExpress database at http://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-412/.

2.5 Assessment

2.5.1 Assessment measures

We used a number of measures to evaluate the quality of the inferred gene networks from fastBMA. We defined a true positive (TP) as an edge in the inferred gene network that was also present in the gold standard. A false positive (FP) was an edge in the inferred gene network that was missing in the gold standard. A false negative (FN) was an missing edge in the inferred gene network that was present in the gold standard, and a true negative (TN) was an missing edge that was also missing in the gold standard. True positive rate (TPR) was calculated by dividing the number of TP by the sum of TP and FN, *i.e.*, (TP/(TP+FN)). Precision was calculated by dividing the number of TP by the total number of edges in the inferred gene network, *i.e.*, (TP/(TP+FP)). Recall was calculated by dividing the number of TP by the sum of TP and FN, *i.e.*, (TP/(TP+FN)). Both precision and recall were useful measures of the positive predictive value and sensitivity of the methodology. However, TP, FP, FN and TN are all computed by thresholding the resulting posterior probabilities of edges in the gene network, and hence, precision and recall are dependent on the threshold as well. Plots of precision versus recall over different values for the posterior probability threshold give a more complete picture of the accuracy of the network inference. Similarly, the receiver operating characteristic (ROC) plots of *i.e.* (TP/(TP+FN)) versus *i.e.* (FP/(FP+TN)) for different thresholds are also useful. We summarized these plots into a single number by computing the area under the precision recall curve (AUPRC) and area under the receiver operating curve (AUROC) across different posterior probability thresholds [50].

2.5.2 Contingency table

A contingency table captures the association of two discrete variables. An example contingency table is shown in Figure 2.3. Note that the sum of the first row in the contingency

table is the number of edges at the given posterior probability threshold in the generated gene network. The sum of the first column in the contingency table is the number of edges in the gold standard.

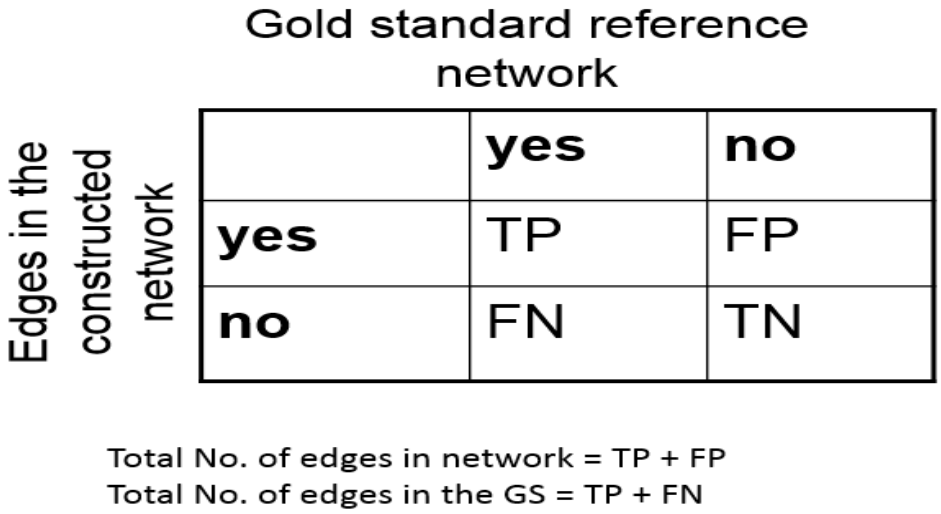


Figure 2.3: Contingency table used in assessment

2.6 Results

2.6.1 DREAM4 10-gene data

Figure 2.4 shows the result of applying fastBMA and TR to one of the DREAM4 10-gene data set.

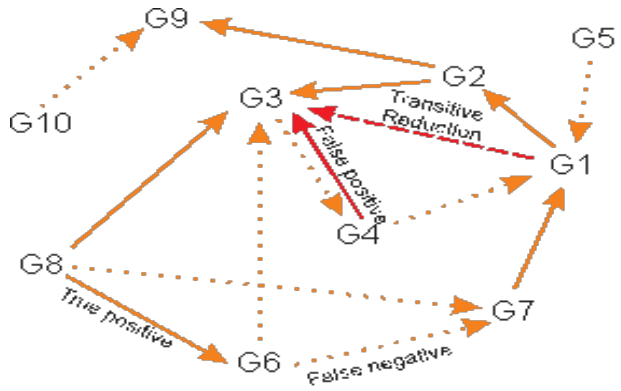


Figure 2.4: DREAM4-10 gene-set No.4 gold standard with fastBMA and transitive reduction

In Figure 2.4, the gene network inferred was shown in solid lines. The gold lines indicated the gold standard network. Red solid lines indicated the false positive predictions and dotted gold lines indicated false negative predictions. The dashed red line from G1 to G3 was a false positive edge that was eliminated by the transitive reduction process due to the better indirect path from G1 to G2 to G3.

2.6.2 DREAM4 100-gene data

Figure 2.5 summarized the assessment results for ScanBMA and fastBMA when applied to the DREAM4 100-gene simulated time series data. Since there are five simulated data sets, the TP and FP columns shown were the sum across all 5 sets. The number of FP edges inferred by fastBMA was much smaller than those of ScanBMA, subsequently resulted in higher precision. The precision of fastBMA either with or without TR was much higher than ScanBMA. The AUPRC for fastBMA was also higher than those in ScanBMA.

Method	Precision	Recall	AUROC	AUPRC	TP	FP
ScanBMA	0.153	0.942	0.657	0.101	193	1062
FastBMA (w/o TR)	0.708	0.776	0.658	0.188	159	66
FastBMA (w/ TR)	0.713	0.766	0.646	0.177	157	63

Figure 2.5: Average performance of ScanBMA and fastBMA at 50% posterior probability threshold on the DREAM4 100-gene networks. w/TR means that transitive reduction has been applied to the gene networks generated by fastBMA. w/oTR means that transitive reduction has not been applied to the gene networks generated by fastBMA.

2.6.3 Yeast time series gene expression data

Figure 2.6 summarized the assessment results of ScanBMA and fastBMA when applied to the 3556-gene yeast time series gene expression data. fastBMA with number of variables = 20 had a precision of 0.4429, higher than ScanBMA with the same number of variables. The AUPRC for fastBMA was also higher than ScanBMA. Additionally, the number of true positive edges inferred by fastBMA was also higher than that of ScanBMA.

Method	Precision	Recall	AUROC	AUPRC	TP	FP
ScanBMA[20]	0.391	0.0132	0.601	0.0747	227	353
(w/prior)						
ScanBMA[3556]	0.274	0.0074	0.629	0.074	127	336
(w/prior)						
FastBMA (w/o TR)[20]	0.4429	0.0206	0.533	0.0784	353	444
(w/prior)						
FastBMA (w/ TR)[20]	0.4428	0.0205	0.532	0.0772	352	443
(w/prior)						
FastBMA (w/o TR)[3556]	0.157	0.0022	0.503	0.0453	37	199
(w/o prior)						
FastBMA (w/ TR)[3556]	0.152	0.002	0.501	0.0445	34	189
(w/o prior)						
FastBMA (w/o TR)[3556]	0.268	0.006	0.512	0.0534	103	281
(w/ prior)						
FastBMA (w/ TR)[3556]	0.269	0.0059	0.509	0.051	101	275
(w/ prior)						

Figure 2.6: Assessment of ScanBMA and fastBMA in terms of precision, recall, AUROC and AUPRC at 50% posterior probability threshold on the yeast data. Numbers in [ ] are number of variables which means how many variables are selected from a model. *w/prior* means that external priors files are been used with fastBMA. *w/oprior* means that external priors files are not been used with fastBMA. w/TR means that transitive reduction has been applied to the gene networks generated by fastBMA. w/oTR means that transitive reduction has not been applied to the gene networks generated by fastBMA.

2.6.4 Improvement in running time

Figure 2.7 summarized the running time results for applying ScanBMA and fastBMA to the DREAM4 100-gene simulated time series data. We showed that fastBMA produced a significant improvement in running time over ScanBMA.

Method	Running time (sec)	Running time per gene (sec)
ScanBMA	1.914	0.01914
FastBMA (w/o TR)	0.0228	0.000228
FastBMA (w/ TR)	0.0289	0.000289

Figure 2.7: Running time of ScanBMA and fastBMA on the DREAM4 100-gene data. w/TR means that transitive reduction has been applied to the gene networks generated by fastBMA. w/oTR means that transitive reduction has not been applied to the gene networks generated by fastBMA.

Similarly, Figure 2.8 summarized the running time results for applying ScanBMA and fastBMA to the 3556-gene yeast time series gene expression data. We showed that when the number of variables = 20, fastBMA ran much faster than ScanBMA. Also, when the number of variables increased, fastBMA was still more efficient than ScanBMA.

Method	Running time (sec)	Running time per gene (sec)
ScanBMA[20]	1.562	0.0781
(w/ prior)		
ScanBMA[3556]	84.354	0.0237
(w/ prior)		
FastBMA (w/o TR)[20]	0.568	0.0284
(w/ prior)		
FastBMA (w/ TR)[20]	0.5773	0.0289
(w/ prior)		
FastBMA (w/o TR)[3556]	56.9762	0.016
(w/o prior)		
FastBMA (w/ TR)[3556]	56.9806	0.016
(w/o prior)		
FastBMA (w/o TR)[3556]	47.5562	0.0134
(w/ prior)		
FastBMA (w/ TR)[3556]	47.8344	0.0135
(w/ prior)		

Figure 2.8: Running time of ScanBMA and fastBMA on the 3556-gene yeast data. Numbers in [ ] are number of variables which means how many variables are selected from a model. $w/prior$ means that external priors files are been used with fastBMA. $w/oprior$ means that external priors files are not been used with fastBMA. w/TR means that transitive reduction has been applied to the gene networks generated by fastBMA. w/oTR means that transitive reduction has not been applied to the gene networks generated by fastBMA.

2.7 Summary

I experimented, benchmarked and tested fastBMA, an improved and efficient implementation of ScanBMA method for inferring gene networks using both simulated and real time-series gene expression data. fastBMA contains an additional transitive reduction post-processing step after the regression step to remove spurious paths. We showed that fastBMA

inferred gene networks with higher precision than ScanBMA. The gene networks generated from fastBMA were also comparable in size to the gold standard.

Chapter 3

GENE NETWORK INFERENCE INTEGRATING DIFFERENT TYPES OF PERTURBATION DATA

In this chapter, I will report the results of applying gene network inference methods to the gene expression data generated by the Library of Integrated Network-based Cellular Signatures (LINCS) project (http://www.lincsproject.org/), funded by the National Institutes of Health (NIH) Big Data to Knowledge (BD2K) initiative [61].

3.1 L1000 gene expression data

The NIH LINCS project aims to create a network-based understanding of biology by measuring gene expression changes subject to perturbations (http://www.lincsproject.org/). For example, data mining on LINCS data which includes clustering, visualization [62] and multiway factorization, drug to drug interaction network. There are two types of perturbations in the LINCS data, including both genetic perturbations and drug perturbations. Genetic perturbations include gene deletions, gene over-expression, insertions of other genes, and frame shift mutations [63]. Drugs bind to their target proteins which interact with downstream effectors and ultimately perturb the transcriptome of a cell. These drug perturbation data reveal information about their source, i.e., drugs' targets [64]. In this thesis, I will focus on one type of genetic perturbation data called knockdown data, in which the expression level of a given gene is reduced.

The LINCS L1000 gene expression data measured the expression level of 978 landmark genes across different perturbations [65]. The L1000 assay uses the Luminex technology, and is a bead-based, high-throughput assay in which the expression levels of two genes are

measured on the same bead [66]. The 978 landmark genes are chosen by the LincsCloud project at the Broad Institute (http://www.lincscloud.org/l1000/). The intuition is that the expression levels of many genes are highly correlated and that one could measure the expression levels of selected representative genes and use these measurements to interpolate the expression levels of other genes [67].

The LincsCloud project generated the L1000 data, and made the L1000 data publicly available at different levels such that the data at higher levels are more processed than the data at lower levels. Specifically, level 1 data was the raw unprocessed data. Level 2 data was the gene expression values per 1000 genes after deconvolution from Luminex beads. Since the expression levels of two genes are measured on the same bead, it is essential to deconvolute the data, *i.e.* to assign the two measurements to the two genes. Level 3 data was the quantile normalized data. The intuition of quantile normalization is to make sure that the data distributions from different experiments have similar statistical properties. Level 4 was the z-scores data. They were profiles of differentially expressed genes computed by comparing the perturbation experiments to the control experiments.

Dr. Hung developed an alternative data processing pipeline called L1K++ for LINCS L1000 data. The L1k++ pipeline is a fast pipeline that increases the accuracy of L1000 gene expression data. It is written in C++ and starts from the raw level 1 data. The main idea of L1K++ is to combine all data from different replicates in order to increase accuracy. Level 1 data from L1000 data was taken to make a more compact form which is level 1.5 data in L1K++. Dr. Hung combined data from all control wells in a given cell line, combined data from replicates treatment wells, used quantile normalization scheme to normalize the data across wells. After combining the data, the two signals from each bead color are separated using a modified Gaussian Mixturen Model (GMM) [68] and this generates the level 2.5 ensemble data [69]. Ensemble means that different replicates are combined together.

In our experiments of applying the fast Bayesian Model Averaging (fastBMA) to the L1000 gene expression data, we used level 3 from LincsCloud and level 2.5 ensemble data from L1k++ tool. The level 2.5 ensemble data is deconvoluted gene expression values

which is similar to level 2 data from LincsCloud. It differs from level 2 in that there is no normalization with the controls (bead types 1-10) and a quantile normalization is done before de-convolution between the replicates. Also, level 2.5 ensemble differs from level 3 data in LincsCloud which use plate level quantile normalization. Note that in the level 2.5 ensemble data, the replicates have already been combined. These LINCS L1000 data were downloaded from a browser-based tool written in php by Dr. Hung [70].

3.2 Related work on L1000 gene expression data

A systematic compound signature discovery pipeline covering from raw L1000 data processing to drug screening and mechanism generation called constrained sparse non-negative matrix factorization (csNMF) was developed [71]. csNMF improved upon the original L1000 pipeline by discovering compound signatures of breast cancer that were consistent with the LINCS data and were clinically relevant. csNMF was optimized using the multiplicative algorithm [72]. It bridges the gap between the LINCS signature library and clinical and biomedical research needs.

As another example of work using the L1000 data, Duan *et al.* developed an interactive HTML5 web-based software application called LINCS Canvas Browser (LCB) to facilitate querying, browsing and interrogating of the currently available L1000 data from LincsCloud [62].

System-wide profiling of genes and proteins produce lists of differentially expressed genes/proteins. These lists could be used as input for computing enrichment with existing lists created from prior knowledge. While there are many enrichment analysis tools out there, Chen *et al.* presented Enrichr which is an integrative web-based and mobile software application that includes new gene-set libraries, an alternative approach to rank enriched terms, and interactive visualization approaches to show enrichment results. It is easy to use to produce various types of visualization summaries of functions of gene lists and it is open source and freely available online at http://amp.pharm.mssm.edu/Enrichr/ [73].

Large corpora of kinase small molecule inhibitor data are available from the LINCS project and the literature. The question is how applicable these heterogeneous data sets are in the prediction of kinase activities. Schürer *et al.* accessed almost 500,000 molecules from the Kinase Knowledge Base (KKB) and generated over 180 distinct data sets covering all major groups of the human kinome. Then, Schürer *et al.* generated hundreds of classification and regression models. Next, Schürer *et al.* applied the best classifiers to compounds most recently profiled in the NIH LINCS program and found satisfying agreement of profiling results with predicted activities. The results show that although heterogeneous in nature, the NIH LINCS data sets are valuable to develop accurate predictors for Kinome-wide virtual screening applications [74].

Reconstructing signaling and regulatory response networks is one of the main goals of systems biology. There are some successful methods for doing this task, however, these methods have so far been applied to reconstruct a single response network at a time. In order to improve this, Jain *et al.* developed the Multi-Task Signaling and Dynamic Regulatory Events Miner (MT-SDREM) which is a multi-task learning method which jointly models networks for several related conditions. Jain *et al.* applied MT-SDREM to reconstruct dynamic human response networks for three flu strains. The MT-SDREM method was able to identify known and novel factors and genes, improving upon previous methods that only model each condition independently. The networks generated by MT-SDREM were also better at identifying proteins which indicates that join learning can still lead to condition-specific and accurate networks [75].

ChIP-seq experiments provide a plethora of data regarding transcription regulation. Kou *et al.* collected and expanded a database where results are converted from ChIP-seq experiments into gene-set libraries. In addition, Kou *et al.* compiled data from the Encyclopedia of DNA Elements (ENCODE) project [76]. Moreover, Kou *et al.* processed data from the NIH Epigenomics Roadmap project. All the data were available as gene-set libraries which are useful for gene-set enrichment analyses. Also, Kou *et al.* constructed regulatory networks to identify groups of regulators from these gene-set libraries. Furthermore, Kou *et*

al. created a web-based application software where users can conduct enrichment analyses or download the combined data set data in various formats. The open source ChIP-seq Enrichment Analysis 2 (ChEA2) web-based application software and data sets are available online at http://amp.pharm.mssm.edu/ChEA2 [77].

3.3 Using genetic perturbation in gene network inference

Pinna *et al.* demonstrated how simulated knockdown data provided key information in gene network inference [78]. Pinna *et al.* proposed the winning submission in the Dialogue for Reverse Engineering Assessments and Methods (DREAM)4 challenge [55, 56, 79]. In particular, they only used the knockout data to infer gene networks, and ignored the other data sources provided by the challenge [78].

They presented a two-step approach for reconstructing regulatory gene networks: the first step was to identify the observed effects induced by directed perturbations were collected in a signed and directed perturbation graph (PG). The second step was to use a graph algorithm to identify and eliminate those edges in PG that can be explained by paths that are likely to reflect indirect effects. Therefore, it is necessary to have the capability of generating perturbation graphs. A perturbation graph was generated from the perturbation data [80].

One method called Down-ranking of feed-forward loops (DR-FFL) used a z-score-based strategy to compute the PG and did not consider edge signs in the post-processing graph algorithm. In the pre-processing step, a confidence weight was assigned to each possible edge $i \rightarrow j$ of the network by computing the absolute value of the standard z-score Z_{ij}. The latter quantified the difference between the expression $G^{ko}_{i,j}$ of gene j under knockout/perturbation of gene i and its mean u_j, normalized by the standard deviation σ_j:

$$Z_{ij} = \frac{G^{ko}_{i,j} - u_j}{\sigma_j}$$

Mean u_j and standard deviation σ_j were computed on all variable expression measurements of gene j, including the wild type G^{wt}_j. Then, the PG was obtained by selecting all those

edges whose $|Z_{ij}|$ were larger than a given threshold β. After generating the PG, they trimmed the perturbation graph using a graph algorithm which removed edges that connect nodes from different strongly connected components. Specifically, DR-FFL removed an edge $i \rightarrow j$ from the PG if i and j were from different components and if there was an alternative path connecting i and j without using edge $i \rightarrow j$ [78, 80].

3.4 Data

In this thesis, we focused on the A375 cell line which is a human skin melanoma cell line with over 100,000 experiments in the LINCS data. Among all these experiments, about 11,335 (11.34%) are drug perturbation experiments, about 15,409 (15.41%) are genetic perturbation experiments and about 13,121 (13.12%) are knockdowns. I experimented with both the level 3 data from the LincsCloud project of Broad Institute and the level 2.5 ensemble data from the L1K++ tool [69].

3.5 Methods

3.5.1 Posterior probability odds method developed by Young et al.

A key difference between the work presented in this chapter and the previous chapter is that the L1000 knockdown data do not consist of any time points. Therefore, the methods described in Chapter 2 of this thesis are not directly applicable. Young *et al.* developed a fast, simple method for inferring regulatory relationships from knockdown experiments using the L1000 gene expression data [10]. In particular, they used Bayes factors to infer differentially expressed genes when comparing the knockdown experiments to the control experiments [10]. Bayes factor is the posterior odds of the null hypothesis when the prior probability on the null is one-half. It is a practical tool of applied statistics, it offers a way of evaluating evidence in favor of a null hypothesis, it provides a way of incorporating external information into the evaluation of evidence of a hypothesis, and it doesn't require alternative models to be nested [81]. The core step in the newly developed posterior odds

method by Young *et al.* was to calculate plate-level z-values for each gene in a knockdown experiment. In order to compute the z-scores, Young *et al.* first calculated the plate-level means and standard deviations. After transferring data this way, Young *et al.* used a simple linear regression model to model the change in a target gene t as dependent on the change in the knockdown gene h. Then, Young *et al.* estimated this model with a Bayesian approach using Zellner's g-prior [49] for the model parameters. Next, the regression model with the g-prior allowed to calculate the Bayes factor for the chosen model versus the null model of no dependency of target gene t on knockdown gene h. Young *et al.* calculated the correlations between the knockdown gene and each other gene to get the Bayes factor. Finally, the posterior probability that h regulated t could be calculated given the Bayes factor [10].

3.5.2 Our contributions

Our goal here was to integrate different types of L1000 gene expression data in gene network inference. In particular, we integrated the posterior probabilities computed using the L1000 knockdown data into fastBMA. Including external knowledge through the prior edge probability could provide a significant boost in accuracy and precision [28]. These prior edge probabilities guide the search of the candidate regulator (parent nodes) in the model space. The prior edge probabilities were generated from the A375 cell line knockdown data using the posterior odds method and implementation developed by Young *et al.* [10]. Young *et al.* applied the posterior odds method to the level 3 L1000 data from LincsCloud [10], and we applied their method and implementation to level 2.5 of the L1K++ processed data. Figure 3.1 showed an overview of the integration pipeline.

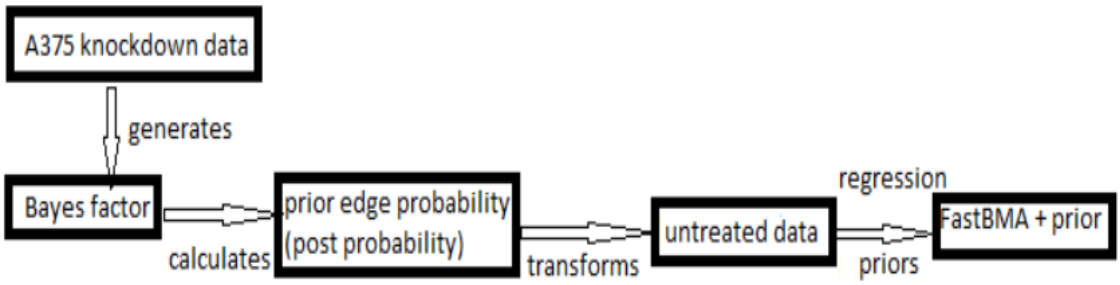


Figure 3.1: Overview of the data integration pipeline. We integrated the L1000 knockdown data in the A375 cell line with fastBMA.

Young *et al.* took the knockdown experiments which target a specific gene to suppress its expression level. This specific gene is the regulator in these knockdown experiments, and the remaining genes are potential target genes, by which to infer networks. Young *et al.* designed a posterior probability approach using the knockdown data to infer edges. The first step was to standardize the knockdown data using the untreated experiments on the same plate to get z-values. The second step was to use a linear regression model to regress each potential target gene on the knocked down gene. This could be converted into a posterior probability if there exists an edge from the knocked-down gene to the target gene. This fast approach also allows the use of prior probabilities. The output of this approach is a ranked edge list [10].

Building on the approach proposed by Young *et al.* [10], we applied this approach to the A375 cell line experiments. First, after we downloaded the level 3 L1000 data for cell line 375 untreated experiments, we calculated the plate-level means and plate-level variances for each gene in the A375 cell line untreated data by using both the level 3 gene expression data and the corresponding meta data. Also, we calculated the standard deviations from the plate-level means and variances. Second, we calculated the z-scores, Bayes factor and posterior probabilities for each knockdown experiment in cell line A375 using the methods and implementation provided by Young *et al.* [10].

Figure 3.2 showed part of the table results. However, in order to apply the information of "postprob" column as priors into fastBMA algorithm, we had to convert the output from Young *et al.* (see Figure 3.2) into a matrix where the columns and rows were both 978 landmark genes. Figure 3.3 showed the posterior probability matrix of knockdown data of the A375 cell line after this conversion. We applied fastBMA was able to run on different LINCS perturbation data sets using this knockdown data matrix as prior.

	row.names	kdgene	target	cor	n	logodds	bf	bf.wp	postprob	postprob.wp
1	1	PSME1	ATF1	0.1784824989	6	-0.5619111774	0.5701184	0.0002852018	0.3631054	0.0002851205
2	2	PSME1	RHEB	0.5948312309	6	0.1042664564	1.1098962	0.0005552257	0.5260430	0.0005549176
3	3	PSME1	FOXO3	-0.2963921186	6	-0.4580835049	0.6324947	0.0003164055	0.3874406	0.0003163055
4	4	PSME1	RHOA	0.2642031929	6	-0.4920178600	0.6113914	0.0003058486	0.3794183	0.0003057551
5	5	PSME1	IL1B	-0.4705241058	6	-0.1915148268	0.8257074	0.0004130602	0.4522671	0.0004128897
6	6	PSME1	ASAH1	-0.1009025846	6	-0.6009730448	0.5482779	0.0002742761	0.3541211	0.0002742009
7	7	PSME1	RALA	0.6446059337	6	0.2549909616	1.2904500	0.0006455478	0.5634046	0.0006451313
8	8	PSME1	ARHGEF12	0.7330058301	6	0.5821779407	1.7899326	0.0008954140	0.6415684	0.0008946129
9	9	PSME1	SOX2	-0.1049058434	6	-0.5994993184	0.5490865	0.0002746806	0.3544583	0.0002746052
10	10	PSME1	SERPINE1	0.6398792705	6	0.2397480713	1.2709289	0.0006357824	0.5596516	0.0006353784
11	11	PSME1	HLA-DMA	0.4456556060	6	-0.2390523850	0.7873736	0.0003938838	0.4405199	0.0003937287
12	12	PSME1	EGF	-0.4658694794	6	-0.2006789148	0.8181751	0.0004092922	0.4499980	0.0004091247
13	13	PSME1	APP	-0.1047056548	6	-0.5995743969	0.5490453	0.0002746600	0.3544411	0.0002745845
14	14	PSME1	NOS3	0.3925466382	6	-0.3294026470	0.7193533	0.0003598566	0.4183860	0.0003597271
15	15	PSME1	CSNK1A1	-0.2150319384	6	-0.5356497571	0.5852889	0.0002927908	0.3692001	0.0002927051
16	16	PSME1	NFATC4	-0.6460237526	6	0.2596034065	1.2964158	0.0006485322	0.5645388	0.0006481119
17	17	PSME1	TBP	0.2691807264	6	-0.4870539410	0.6144339	0.0003073706	0.3805878	0.0003072762
18	18	PSME1	BRCA1	0.3602942824	6	-0.3773409640	0.6856822	0.0003430126	0.4067684	0.0003428950

Figure 3.2: Partial results of Bayes factor and posterior probability of knockdown data of the A375 cell line. 'kdgene' stands for knocked down gene name, 'target' is the target gene name, 'cor' stands for correlation between a knocked down gene and another gene, 'logodds' stands for log of the Bayes factor, 'bf' stands for Bayes factor, 'bf.wp' stands for Bayes factor includes prior (wp = with prior), 'postprob' stands for posterior probability, 'postprob.wp' stands for posterior probability includes prior (wp = with prior).

	row.names	PSME1	ATF1	RHEB	FOXO3	RHOA	IL1B	ASAH1	RALA	ARHGEF12	SOX2
1	PSME1	0.0000000	0.3631054	0.5260430	0.3874406	0.3794183	0.4522671	0.3541211	0.5634046	0.6415684	0.3544582
2	ATF1	0.3552145	0.0000000	0.6706113	0.7710775	0.7684400	0.5362033	0.3673135	0.3504214	0.6505514	0.8564237
3	RHEB	0.0005000	0.0005000	0.0000000	0.0005000	0.0005000	0.0005000	0.0005000	0.0005000	0.0005000	0.0005000
4	FOXO3	0.7452627	0.6679491	0.5358072	0.0000000	0.6712059	0.6996203	0.4579738	0.4701799	0.3247432	0.5435749
5	RHOA	0.5242365	0.5107980	0.8953242	0.4060721	0.0000000	0.9099797	0.8220782	0.6792076	0.3104262	0.6900110
6	IL1B	0.3887802	0.4508270	0.4055154	0.3351157	0.5973977	0.0000000	0.3599746	0.3333432	0.3732718	0.5489256
7	ASAH1	0.3689637	0.8006001	0.6274711	0.5821019	0.3525945	0.3352460	0.0000000	0.4854531	0.6424056	0.3742163
8	RALA	0.5555061	0.6336738	0.3360389	0.3478160	0.3792759	0.7287072	0.3341113	0.0000000	0.3387776	0.4070266
9	ARHGEF12	0.3539401	0.3619711	0.3915302	0.3947385	0.5077988	0.5451159	0.5516177	0.3503110	0.0000000	0.6931124
10	SOX2	0.7798951	0.4490889	0.5764537	0.5416587	0.4133682	0.3262498	0.3311663	0.5481997	0.4740140	0.0000000
11	SERPINE1	0.7458426	0.3678754	0.4240673	0.5548149	0.4172127	0.6595325	0.4302032	0.5275254	0.5459148	0.3847069
12	HLA-DMA	0.5077839	0.5661437	0.5700973	0.4760951	0.6445560	0.3508807	0.3784274	0.4249947	0.4226567	0.3729324
13	EGF	0.4080601	0.5139527	0.3140550	0.5805232	0.3144900	0.3316563	0.6171204	0.3146304	0.4329120	0.3944060
14	APP	0.3571923	0.4029171	0.3575274	0.6302917	0.4250855	0.3740414	0.3530604	0.3506329	0.3630217	0.3640851
15	NOS3	0.4569545	0.4720944	0.4473849	0.3439788	0.3659316	0.3511484	0.5486973	0.4015676	0.4135930	0.7733403
16	CSNK1A1	0.3408969	0.5856240	0.3212577	0.3343862	0.4156167	0.3264092	0.4327323	0.3212522	0.5379147	0.3282037
17	NFATC4	0.3552145	0.6706113	0.7710775	0.7684400	0.5362033	0.3673135	0.3504214	0.6505514	0.8564237	0.7169650
18	TBP	0.6137360	0.7191529	0.5016827	0.3794490	0.4640215	0.4259312	0.6862124	0.6730414	0.4404220	0.5285140

Figure 3.3: Partial matrix result of posterior probability of knockdown data of the A375 cell line. The first row and second column have all the gene names. Values in the corresponding cells are the posterior probabilities from figure 3.2

3.6 Assessment

In order to assess the generated gene networks, we used the same assessment criteria adopted by Young *et al.* [10], *i.e.* the regulatory relationships documented in the TRANSFAC [82] and JASPAR [83] (T&J) databases. We downloaded the processed versions of these regulatory relationships from ENRICHR at http://amp.pharm.mssm.edu/Enrichr/ [73]. TRANSFAC is a database on transcription factors, their genomic binding sites and DNA-binding profiles [82]. JASPAR is an open-access database of high-quality, annotated, matrix-based transcription factor binding site profiles for multicellular eukaryotes. JASPAR is available at http://jaspar.genereg.net/ [83]. This T&J assessment criteria included 37 transcription

factors that overlapped with the LINCS landmark genes. For these 37 transcription factors, the T&J criteria consist of approximately 4,200 regulation-target pairs among the landmark genes [10]. We used the measures described in Chapter 2 (precision, recall, area under the receiver operating curve (AUROC), area under the precision recall curve (AUPRC), true positive (TP), and false positive (FP)) to evaluate the quality of the inferred gene networks from fastBMA using LINCS data sets.

3.7 Results

Note that in the figures below, Broad means the Broad Institute. Level 3 data came from the LincsCloud project, however, the statement of Broad level 3 data is correct as well. Thus, Broad level 3 data is the same as LincsCloud level 3 data.

3.7.1 Integration of knockdown and untreated data

Figure 3.4 summarized the assessment results using knockdown data as priors and the untreated data in the regression step. We experimented with both the level 3 data from Broad and the level 2.5 ensemble data from L1K++ using the A375 cell line. The TP inferred by L1K++ level 2.5 ensemble data processing was higher than the TP inferred from Broad level 3 data processing. The FP inferred by L1K++ level 2.5 ensemble data processing was smaller than the FP inferred from Broad level 3 data processing. These two statistical phenomenon resulted in higher precision and recall of L1K++ level 2.5 ensemble data processing. In general, we conclude that when input data was untreated cell line data and prior was knockdown data, gene networks inferred by L1k++ level 2.5 ensemble were more accurate than networks inferred from Broad level 3 data processing.

Data Processing	Input Data	Priors	Parameters	Precision	Recall	AUROC	AUPRC	TP	FP
Broad lvl 3	A375 untrt	A375 KD	TR, no self loop, nvar 20	0.0138	0.0042	0.501	0.1041	18	1282
L1K++ lvl 2.5 ensemble	A375 untrt	A375 KD	TR, no self loop, nvar 20	0.0164	0.0049	0.5011	0.104	21	1261

Figure 3.4: Performance of fastBMA at 95% posterior probability threshold using knockdown data as priors and the untreated data in the regression step. We experimented with both the level 3 data from Broad and the level 2.5 ensemble data from L1K++ using the A375 cell line.

3.7.2 Integration of knockdown and drug perturbation data

Figure 3.5 summarized the assessment results using knockdown data as priors and the drug perturbation data in the regression step. We experimented with both the level 3 data from Broad and the level 2.5 ensemble data from L1K++ using the A375 cell line. The TP inferred from L1K++ level 2.5 ensemble data processing was higher than the TP inferred by Broad level 3 data processing. Although the AUROC and AUPRC of L1K++ level 2.5 ensemble data processing were smaller than the AUROC and AUPRC in Broad level 3 data processing, the precision and recall of L1K++ level 2.5 ensemble data processing were higher than the precision and recall in Broad level 3 data processing. In general, we conclude that when the drug perturbation data was used as the input data and the prior probabilities were derived from knockdown data, gene networks inferred by L1k++ level 2.5 ensemble were more accurate than networks inferred from the Broad level 3 data processing.

Data Processing	Input Data	Priors	Parameters	Precision	Recall	AUROC	AUPRC	TP	FP
Broad lvl 3	A375 drug perturbations	A375 KD	TR, no self loop, nvar 20	0.0119	0.0035	0.5005	0.1037	15	1247
L1K++ lvl 2.5 ensemble	A375 drug perturbations	A375 KD	TR, no self loop, nvar 20	0.0126	0.0037	0.4855	0.0934	16	1257

Figure 3.5: Performance of fastBMA at 95% posterior probability threshold using knockdown data as priors and the drug perturbation data in the regression step. We experimented with both the level 3 data from Broad and the level 2.5 ensemble data from L1K++ using the A375 cell line.

3.7.3 Performance of fastBMA with transitive reduction using knockdown data only

Figure 3.6 summarized the assessment results when fastBMA was applied to the knockdown data only. Specifically, no regression step was involved, and the gene networks were inferred using the posterior probability odds method by Young *et al.* [10]. The numbers of TP were the same in both cases with TR and without TR. However, TR helped remove 10 FP edges as shown which resulted in higher precision. And the AUROC and AUPRC when applying fastBMA with TR were higher. In general, we could conclude that for Broad level 3 data, using knockdown data only, TR helped to generate more accurate networks than without TR.

Data Processing	Input Data	Priors	Parameters	Precision	Recall	AUROC	AUPRC	TP	FP
Broad lvl 3	N/A	A375 KD	TR	0.1404	0.0098	0.5016	0.1011	41	251
Broad lvl 3	N/A	A375 KD	noTR	0.1358	0.0098	0.5014	0.101	41	261

Figure 3.6: Performance of fastBMA at 50% posterior probability threshold with transitive reduction using knockdown data only.

3.7.4 Integration of knockdown from different data processings with drug perturbation data

Figure 3.7 summarized the assessment results using knockdown data from two different data processings (Broad level 3 and L1K++ level 2.5 ensemble) as priors when applying fastBMA using the drug perturbation data. The first row in Figure 3.7 showed the results of A375 knockdown data from Broad level 3 and the second row showed the results of A375 knockdown data from L1K++ level 2.5 ensemble. The precision and recall both were higher when we used knockdown data from L1K++ level 2.5 ensemble as prior. Also, the TP was increased by a single when we used knockdown data from L1K++ level 2.5 ensemble as prior. Therefore, we conclude that the knockdown data extracted from L1K++ level 2.5 ensemble was more informative than the knockdown data extracted from Broad level 3. Thus, outputting more accurate gene networks.

Input Data	Priors	Parameters	Precision	Recall	AUROC	AUPRC	TP	FP
A375 drug perturbations	A375 KD from Broad lvl 3	TR, no self loop, nvar 20	0.0119	0.0035	0.4859	0.0935	15	1247
A375 drug perturbations	A375 KD from L1K++ lvl 2.5e	TR, no self loop, nvar 20	0.0126	0.0037	0.4855	0.0934	16	1257

Figure 3.7: Performance of knockdown data from different data processing as priors at 95% posterior probability threshold on inferring gene networks from the A375 cell line data using all drug perturbations as input.

3.8 Conclusion

I experiment and tested the fastBMA method for inferring gene networks using different types of perturbation data from LINCS L1000 gene expression data. We learned that the knockdown data source was the most informative data source. This observation is consistent with the observations from Pinna *et al.*. We also observed that the L1K++ pipeline helped generate gene networks with higher precision and recall.

Chapter 4

CONCLUSIONS AND FUTURE WORK

We benchmarked the performance of fastBMA in the inference of gene networks using simulated data, real time-series data, and the Library of Integrated Network-based Cellular Signatures (LINCS) perturbation and knockdown data. The fastBMA approach is fast and has the ability to incorporate prior information when available. It contains transitive reduction after regression step to remove spurious direct paths and replace them with indirect paths. In general, fastBMA infers networks with higher precision than the previously developed methods. The networks generated from fastBMA are also similar in size to the target networks.

My major contribution in this thesis is in systematically benchmarking different methods using different data sources. Also, I was also responsible in collecting and summarizing results from the benchmarking experiments. I have learned that benchmarking is not easy and it often involves multiple rounds in order to achieve the goal. For example, I had to change the relevant parameters many times to test fastBMA method in order to achieve satisfying results eventually. Using different data sets was another challenge in this process. Different data sets came in various formats. I first had to change their formats so that they could be fitted into the testing method. The interpretation of benchmarking data could turn out to be a difficult and time consuming task. Also, I learned that there are many assessment criteria in gene network inference, including computation time, precision, recall and area under the curve.

BIBLIOGRAPHY

[1] Kashyap H, Ahmed HA, Hoque N, Roy S, and Bhattacharyya DK. Big Data Analytics in Bioinformatics: A Machine Learning Perspective. *Computing Research Repository*, abs/1506.05101, 2015.

[2] Singer E. Biology's Big Problem: There's Too Much Data to Handle. *Quanta Magazine Science*, 2013.

[3] Schena M, Shalon D, Heller R, Chai A, Brown PO, and Davis RW. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proceedings of the National Academy of Sciences of the USA*, 93(20):10614–10619, Oct 1996.

[4] Schena M, Shalon D, Davis RW, and Brown PO. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, Oct 1995.

[5] Bumgarner R. DNA microarrays: Types, Applications and their future. *Current Protocols in Molecular Biology*, 0(22):Unit–22.1., Jan 2013.

[6] Meyerson M, Gabriel S, and Getz G. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics*, 11(10):685–696, Oct 2010.

[7] Nilsson T, Mann M, Aebersold R, Yates III JR, Bairoch A, and Bergeron JJM. Mass spectrometry in high-throughput proteomics: ready for the big time. *Nature Methods*, 7:681–685, 2010.

[8] Joyce AR and Palsson B. The model organism as a system: integrating 'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7:198–210, March 2006.

[9] Nature Publishing Group. Gene regulatory networks. www.nature.com/subjects/gene-regulatory-networks, 2016.

[10] Young WC, Yeung KY, and Raftery AE. Model-Based Clustering with Data Correction for Removing Artifacts in Gene Expression Data. Technical Report 641, Department of Statistics University of Washington, Feb 2016.

[11] Merid SK, Goranskaya D, and Alexeyenko A. Distinguishing between driver and passenger mutations in individual cancer genomes by network enrichment analysis. *BMC Bioinformatics*, 15:308, 2014.

[12] Dehmer M and Emmert-Streib F. Comparing large graphs efficiently by margins of feature vectors. *Applied Mathematics and Computation*, 188(2):1699–1710, May 2007.

[13] Dehmer M and Mehler A. A new method of measuring similarity for a special class of directed graphs. *Tatra Mountains Mathematical Publications*, 36(125):1–22, 2007.

[14] Ideker T and Krogan NJ. Differential network biology. *Molecular Systems Biology*, 8:565, Jan 2012.

[15] Islam MF, Hoque MM, Banik RS, Roy S, Sumi SS, Hassan FM, Tomal MT, Ullah A, and Rahman KM. Comparative analysis of differential network modularity in tissue specific normal and cancer protein interaction networks. *Journal of Clinical Bioinformatics*, 3(1):19, Oct 2013.

[16] Schadt EE, Friend SH, and Shaywitz DA. A network view of disease and compound screening. *Nature Reviews Drug Discovery*, 8(4):286–295, Apr 2009.

[17] Emmert-Streib F, Dehmer M, and Haibe-Kains B. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in Cell and Developmental Biology*, 2:38, Aug 2014.

[18] Schadt EE. Molecular networks as sensors and drivers of common human diseases. *Nature*, 461:218–223, Sep 2009.

[19] Pereira E, Camacho-Vanegas O, Anand S, Sebra R, Catalina CS, Garnar-Wortzel L, Nair N, Moshier E, Wooten M, Uzilov A, Chen R, Prasad-Hayes M, Zakashansky K, Beddoe AM, Schadt E, Dottino P, and Martignetti JA. Personalized Circulating Tumor DNA Biomarkers Dynamically Predict Treatment Response and Survival In Gynecologic Cancers. *PLoS One*, 10(12):e0145754, Dec 2015.

[20] Hood L. Systems biology and p4 medicine: past, present, and future. *Rambam Maimonides Medical Journal*, 4(2):e0012, Apr 2013.

[21] Chuang HY, Hofree M, and Ideker T. A decade of systems biology. *Annual Review of Cell and Developmental Biology*, 26:721–744, 2010.

[22] Cytoscape Consortium. Cytoscape. www.cytoscape.org, 2001–2015.

[23] Yoo C, Ramirez L, and Liuzzi J. Big Data Analysis Using Modern Statistical and Machine Learning Methods in Medicine. *International Neurourology Journal*, 18(2):50–57, Jun 2014.

[24] Larraaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armaanzas R, Santaf G, Prez A, and Robles V. Machine learning in bioinformatics. *Brief Bioinformatics*, 7(1):86–112, Mar 2006.

[25] Lo K, Raftery AE, Dombek KM, Zhu J, Schadt EE, Bumgarner RE, and Yeung KY.

Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. *BMC Systems Biology*, 6(1):101, Aug 2012.

[26] Yeung KY, Dombek KM, Lo K, Mittler JE, Zhu J, Schadt EE, Bumgarner RE, and Raftery AE. Construction of regulatory networks using expression time-series data of a genotyped population. *Proceedings of the National Academy of Sciences*, 108(48):19436–19441, Nov 2011.

[27] Fronczuk M, Rafery AE, and Yeung KY. CyNetworkBMA: a Cytoscape app for inferring gene regulatory networks. *Source Code for Biology and Medicine*, 10:11, Nov 2015.

[28] Young WC, Raftery AE, and Yeung KY. Fast Bayesian inference for gene regulatory networks using ScanBMA. *BMC Systems Biology*, 8:47, Apr 2014.

[29] Zhang B and Horvath S. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1), Aug 2005.

[30] Friedman N, Linial M, Nachman I, and Pe'er D. Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, 7(3–4):601–620, Jul 2004.

[31] Pe'er D, Regev A, Elidan G, and Friedman N. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215–S224, 2001.

[32] Filkov V. *Handbook of Computational Molecular Biology*, chapter 27 Identifying Gene Regulatory Networks from Gene Expression Data, pages 27.1–27.24. Chapman&Hall/CRC Press, 2005.

[33] Murphy K and Mian S. Modelling Gene Expression Data using Dynamic Bayesian Networks. Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[34] Zhu J, Chen Y, Leonardson AS, Wang K, Lamb JR, Emilsson V, and Schadt EE. Characterizing Dynamic Changes in the Human Blood Transcriptional Network. *PLoS Computational Biology*, Feb 2010.

[35] D'haeseleer P, Wen X, Fuhrman S, and Somogyi R. Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing (PSB)*, 4:41–52, 1999.

[36] Bansal M, Della Gatta G, and Di Bernardo D. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22(7):815–822, Apr 2006.

[37] Tibshirani R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B (Methodol)*, 58(1):267–288, 1996.

[38] Zou H and Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B (Methodol)*, 67(2):301–320, 2005.

[39] Raftery AE, Madigan D, and Hoeting JA. Bayesian Model Averaging for Linear Regression Models. *Journal of the American Statistical Association*, 92(437):179–191, Mar 1997.

[40] Hoeting JA, Madigan D, Raftery AE, and Volinsky CT. Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–417, 1999.

[41] Wichert S, Fokianos K, and Strimmer K. Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, 20(1):5–20, 2004.

[42] Ernst J and Bar-Joseph Z. STEM: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, 7:191, Apr 2006.

[43] The PSU. Autoregressive Models. https://onlinecourses.science.psu.edu/stat501/node/358, 2016.

[44] Raftery AE. Bayesian Model Selection in Social Research. *Sociological Methodology*, 25:111–163, 1995.

[45] Furnival GM and Wilson RW. Regressions by Leaps and Bounds. *Technometrics*, 16(4):499–511, Nov 1974.

[46] Madigan D and Raftery AE. Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. *Journal of the American Statistical Association*, 89(428):1535–1546, Dec 1994.

[47] Schwarz G. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978.

[48] Yeung KY, Bumgarner RE, and Raftery AE. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21(10):2394–2402, May 2005.

[49] Zellner A. On assessing Prior Distributions and Bayesian Regression Analysis with g-Prior Distributions. *Bayesian Inference Decis Tech: Essays Honor of Bruno De Finetti*, 6:233–243, 1986.

[50] Hung LH, Shi K, Wu M, Young WC, Raftery AE, and Yeung KY. Fastbma: Scalable gene network inference and transitive reduction. Manuscript under preparation.

[51] Yeung KY, Fraley C, Young WC, Bumgarner R, and Raftery AE. Bayesian Model Averaging methods and R package for gene network construction. pages 9–14, New York City, Aug 2014. Big Data Analytic Technology For Bioinformatics and Health Informatics (KDDBHI), workshop at the 20th ACM SIGKDD Conference on Knowledge

Discovery and Data Mining (KDD).

[52] Zhang X. OpenBLAS. www.openblas.net.

[53] Bonaki D, Odenbrett MR, Wijs A, Ligtenberg W, and Hilbers P. Efficient reconstruction of biological networks via transitive reduction on general purpose graphics processors. *BMC Bioinformatics*, 13:281, Oct 2012.

[54] Morris J. Dijkstra's Algorithm. www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html, 1998.

[55] Marbach D, Schaffter T, Mattiussi C, and Floreano D. Generating Realistic in silico Gene Networks for Performance Assessment of Reverse Engineering Methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

[56] Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, and Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences of the USA*, 107(14):6286–6291, 2010.

[57] Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alexopoulos LG, Xue X, Clarke ND, Altan-Bonnet G, and Stolovitzky G. Towards a Rigorous Assessment of Systems Biology Models: The DREAM3 Challenges. *PLoS One*, 5(3):9202, 2010.

[58] Prill RJ, Saez-Rodriguez J, Alexopoulos LG, Sorger PK, and Stolovitzky G. Crowdsourcing Network Inference: The DREAM Predictive Signaling Network Challenge. *Science Signaling*, 4(189):mr7, Sep 2011.

[59] Stolovitzky G, Monroe D, and Califano A. Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Annals of the New York Academy of Sciences*, 1115:1–22, Dec 2007.

[60] Teixeira MC, Monteiro P, Jain P, Tenreiro S, Fernandes AR, Mira NP, Alenquer M, Freitas AT, Oliveira AL, and S-Correia I. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Research*, 34:446–451, Jan 2006.

[61] National Institutes of Health. BD2K. https://datascience.nih.gov/bd2k, 2015.

[62] Duan Q, Flynn C, Niepel M, Hafner M, Muhlich JL, Fernandez NF, Rouillard AD, Tan CM, Chen EY, Golub TR, Sorger PK, Subramanian A, and Ma'ayan A. LINCS Canvas Browser: interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures. *Nucleic Acids Research*, 42(1):449–460, Jul 2014.

[63] StackExchange. Biology. http://biology.stackexchange.com/questions/17384/gene-perturbation-what-is-it-used-for-explain-to-computer-scientists, 2014.

[64] Isik Z, Baldow C, Cannistraci CV, and Schroeder M. Drug target prioritization by

perturbed gene expression and network information. *Scientific Reports*, 5(17417), Nov 2015.

[65] Broad Institute. Gene Expression Data (L1000). www.lincscloud.org/l1000, 2016.

[66] Peck D, Crawford ED, Ross KN, Stegmaier K, Golub TR, and Lamb J. A method for high-throughput gene expression signature analysis. *Genome Biology*, 7:R61, Jul 2006.

[67] Lincscloud. Benchmarking the Landmark Genes. http://support.lincscloud.org/hc/en-us/articles/202092616-The-Landmark-Genes.

[68] Scikit-Learn Developers. Gmm. http://scikit-learn.org/stable/modules/mixture.html, 2010–2014.

[69] Hung LH. L1K++: A Fast Pipeline that Increases the Accuracy of L1000 Gene Expression Data. https://www.youtube.com/watch?v=jcpEagg1iaQ, 2015.

[70] Hung LH. L1000 Tool. http://ubuntu-dream.cloudapp.net/L1000/index.php?page=1&ipp=2, 2015.

[71] Liu C, Su J, Yang F, Wei K, Ma J, and Zhou X. Compound signature detection on LINCS L1000 big data. *Molecular BioSystems*, 11(3):714–722, Mar 2015.

[72] Lee DD and Seung HS. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[73] Chen EY, Tan CM, Kou Y, Duan Q, Wang Z, Meirelles GV, Clark NR, and Ma'ayan A. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, 14:128, Apr 2013.

[74] Schürer SC and Muskal SM. Kinome-wide activity modeling from diverse public high-quality data sets. *Journal of Chemical Information and Modeling*, 53(1):27–38, Jan 2013.

[75] Jain S, Gitter A, and Bar-Joseph Z. Multitask Learning of Signaling and Regulatory Networks with Application to Studying Human Response to Flu. *PLoS Computational Biology*, 10(12):e1003943, Dec 2014.

[76] The ENCODE Consortium Project. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489:57–74, Sep 2012.

[77] Kou Y, Chen EY, Clark NR, Duan Q, Tan CM, and Ma'ayan A. ChEA2: Gene-Set Libraries from ChIP-X Experiments to Decode the Transcription Regulome. *Lecture Notes in Computer Science*, 8127:416–430, 2013.

[78] Pinna A, Soranzo N, and Fuente A de la. From Knockouts to Networks: Establishing Direct Cause-Effect Relationships through Graph Analysis. *PLoS One*, 5(10):e12912,

Oct 2010.

[79] Dream Challenges. DREAM Challenges. http://dreamchallenges.org/project/closed/dream4-in-silico-network-challenge/, 2016.

[80] Pinna A, Heise S, Flassig RJ, Fuente A de la, and Klamt S. Reconstruction of large-scale regulatory networks based on perturbation graphs and transitive reduction: improved methods and their evaluation. *BMC Systems Biology*, 7:73, Aug 2013.

[81] Kass RE and Raftery AE. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, Jun 1995.

[82] Wingender E, Chen X, Hehl R, Karas H, Liebich I, Matys V, Meinhardt T, Prüss M, Reuter I, and Schacherer F. TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Research*, 28(1):316–319, Jan 2000.

[83] Sandelin A, Alkema W, Engström P, Wasserman WW, and Lenhard B. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32:D91–D94, Jan 2004.