Knowledge-fused Identification of Condition-specific Rewiring of Dependencies in Biological Networks

Ye Tian

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Yue Wang, Chair
Jason Xuan
T. Charles Clancy
Luiz A. DaSilva
Chang-Tien Lu

September 29, 2014
Arlington, Virginia

Keywords: Gaussian Graphical model, differential dependency network, Lasso, ℓ_1-regularization, knowledge incorporation, excess risk, degree of knowledge incorporation

Knowledge-fused Identification of Condition-specific Rewiring of Dependencies in Biological Networks

Ye Tian

(ABSTRACT)

Gene network modeling is one of the major goals of systems biology research. Gene network modeling targets the middle layer of active biological systems that orchestrate the activities of genes and proteins. Gene network modeling can provide critical information to bridge the gap between causes and effects which is essential to explain the mechanisms underlying disease. Among the network construction tasks, the rewiring of relevant network structure plays critical roles in determining the behavior of diseases. To systematically characterize the selectively activated regulatory components and mechanisms, the modeling tools must be able to effectively distinguish significant rewiring from random background fluctuations. While differential dependency networks cannot be constructed by existing knowledge alone, effective incorporation of prior knowledge into data-driven approaches can improve the robustness and biological relevance of network inference. Existing studies on protein-protein interactions and biological pathways provide constantly accumulated rich domain knowledge. Though novel incorporation of biological prior knowledge into network learning algorithms can effectively leverage domain knowledge, biological prior knowledge is neither condition-specific nor error-free, only serving as an aggregated source of partially-validated evidence under diverse experimental conditions. Hence, direct incorporation of imperfect and non-specific prior knowledge in specific problems is prone to errors and theoretically problematic.

To address this challenge, we propose a novel mathematical formulation that enables incorporation of prior knowledge into structural learning of biological networks as Gaussian graphical models, utilizing the strengths of both measurement data and prior knowledge. We

propose a novel strategy to estimate and control the impact of unavoidable false positives in the prior knowledge that fully exploits the evidence from data while obtains "second opinion" by efficient consultations with prior knowledge. By proposing a significance assessment scheme to detect statistically significant rewiring of the learned differential dependency network, our method can assign edge-specific p-values and specify edge types to indicate one of six biological scenarios. The data-knowledge jointly inferred gene networks are relatively simple to interpret, yet still convey considerable biological information. Experiments on extensive simulation data and comparison with peer methods demonstrate the effectiveness of knowledge-fused differential dependency network in revealing the statistically significant rewiring in biological networks, leveraging data-driven evidence and existing biological knowledge, while remaining robust to the false positive edges in the prior knowledge.

We also made significant efforts in disseminating the developed method tools to the research community. We developed an accompanying R package and Cytoscape plugin to provide both batch processing ability and user-friendly graphic interfaces. With the comprehensive software tools, we apply our method to several practically important biological problems to study how yeast response to stress, to find the origin of ovarian cancer, and to evaluate the drug treatment effectiveness and other broader biological questions. In the yeast stress response study our findings corroborated existing literatures. A network distance measurement is defined based on KDDN and provided novel hypothesis on the origin of high-grade serous ovarian cancer. KDDN is also used in a novel integrated study of network biology and imaging in evaluating drug treatment of brain tumor. Applications to many other problems also received promising biological results.

Acknowledgments

First I want to take this best opportunity to thank my advisor, Dr. Yue Wang, for his guidance and support during my Ph.D study. With the constant and visionary help from Dr. Yue Wang, I was able to explore the beauty of scientific research as well as practical applications with resources and freedom. The wisdom of Dr. Yue Wang has enlightened my life and will continue to do so far beyond of completing this dissertation.

I would like to thank Dr. Jason Xuan for his longtime support and help. My deep gratitude also goes to my other Ph.D committee members Dr. Luiz DaSilva, Dr. Charles Clancy and Dr. Chang-Tien Lu for their insightful suggestions on my dissertation work.

I must thank all the past and current CBIL lab members for making these five years unforgettable. I would like to specifically thank Dr. Bai Zhang for his enormous help and all the interesting discussions in research. I also want to thank Dr. Guoqiang Yu for his help from many ways. I am grateful for Drs. Robert Clarke, Lenna Hilakivi-Clarke, Eric Hoffman, Ie-Ming Shih, Zhen Zhang and David Herrington for their support in collaborative research.

My deepest gratitude goes to my parents, Boyu Tian and Shuzhen Zhou, for their love and support, as always. Finally and most importantly, I want to thank my wife, Wei Ding, for her love and support, for her accompany along the course of everything I face, as always. And my daughter, Norah Tian, who came when I was iterating between Chapters 2 and 3, for the unmatched courage she brought to me, as always will be.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the past decade, we have witnessed great advances in high-throughput genomic and proteomic profiling technologies, such as DNA microarrays, next-generation sequencing (NGS) and mass spectrometry based proteomics and metabolomics. These advances, capable of generating massive amounts of genomic, transcriptomic, proteomic and metabolomic data, provide new opportunities to understand human diseases, identify potential biomarkers, and develop new treatments. This data intensive paradigm has fundamentally transformed biomedical science and holds great promise for the betterment of human health (Bell *et al.*, 2009).

These high-throughput biotechnologies have also heightened the challenges of how to distill biological knowledge and novel insights from the sea of data. New computational approaches and statistical models are needed to effectively model and better interpret these high-dimensional, multi-platform data. Reductionism has led to tremendous success in molecular biology: we can now zoom in to study each individual genes or proteins, be-

ing able to study their composition and aberrations at the resolution of single nucleotide or amino acid and inquire their structural and chemical properties. However, genes and proteins in cells do not work in isolation: they communicate and coordinate with each other to carry out various biological functions. Discovering the mechanisms that orchestrate the activities of genes and proteins in cells remains one of the key goals of systems biology studies. Understanding biological systems at the systems level become an important and promising alternative.

Systems approaches are particularly valuable to study complex diseases like cancers. Unlike many Mendelian diseases, where we can often pinpoint genetic culprits in single or a few genes, causes of complex diseases are multifaceted, involving a multitude of molecular aberrations and environmental factors (Clarke *et al.*, 2008). This complexity is further amplified by the interconnected nature of the biomolecules in the cells, which propagates these aberrations or erroneous signals throughout the system, thus posing a great challenge to elucidate the true causes and underlying mechanisms.

Biological networks provide a conceptual and intuitive framework to investigate, model, characterize and understand complex interactions of different components in a biological system. By employing a holistic approach, network biology studies the "interactome", a set of direct or indirect molecular interactions, of the biological system. A biological network hence represents the molecular "wiring" diagram of a cells information processing system.

Biological networks are useful representations to visualize and understand the functions and interactions of biomolecules. It is challenging to discern patterns and distill knowledge from massive amount of data in a high-dimensional space. An informative network model and a graphical representation reveal the relationships among different cellular components and help to detect subtle patterns by "connecting the dots".

Biological networks reveal high-level relationships, enrichment patterns and system-wide

properties, which are lacking in univariate analyses. Network theory (and graph theory), as a subfield of computer science and mathematics, has established rich theory and found many applications in the world-wide web, social networks, particle physics, *etc.*, some of which are readily applicable to biological networks and disease networks (Barabasi *et al.*, 2011).

Some biological network models are capable of simulating a biological system's dynamical behaviors and properties. The complexity of a biological system is reflected in part by a large number of interacting variables, the dynamics of which are governed by numerous linear or nonlinear relationships, chemical kinetics, feedback loops and stochasticity. Like many physical and engineering systems, biological scientists now have mathematical or computer models at their disposal to perform numerical simulations to understand their temporal behaviors and properties.

Biological networks are a natural and versatile framework to incorporate different sources of data and prior knowledge. The advances in molecular profiling technologies enable data collection at different levels of the biological system and empower scientists with new tools to probe the system. However, due to differences in these technology platforms, data sets are often heterogeneous in nature. Further, signicant efforts have also been made to manually curate and document molecular interactions in cells, such as protein-protein interactions and biological pathways, providing rich domain knowledge. The computational challenge we are facing is how to "optimally" fuse these heterogeneous data and prior knowledge to answer meaningful biological questions, and biological networks present a flexible framework for multi-level data representation and integration.

Identifying the mechanisms that orchestrate the activities of genes and proteins in cells is one of the key goals in systems biology studies. Biological networks are context-specific and dynamic in nature. They continuously evolve to adapt to the changing environment (Luscombe *et al.*, 2004; Consortium, 2011). Under different conditions, different regulatory components and mechanisms are activated and accordingly the topology of the underlying

biological network changes to response to outside stimuli (Beyer *et al.*, 2007; Shen *et al.*, 2011; Chen *et al.*, 2008). The condition-specific biological networks and their topological changes provide great insights into the underlying biology of how the organisms adapt to different conditions and open up an new era of understanding and treatment of diseases (Bandyopadhyay *et al.*, 2010; Barabasi *et al.*, 2011; Hood *et al.*, 2004). With the advances in high-throughput genomic technologies, large amounts of genomic data demand effective computational methods to extract biological knowledge and insights (Clarke *et al.*, 2008). Therefore, data-driven learning algorithms for construction of condition-specific biological networks is of great interest.

A great deal of effort has been made to accomplish this goal. Many approaches have been proposed previously to model biological networks using gene expression data (Li *et al.*, 2008), such as probabilistic Boolean networks (Shmulevich *et al.*, 2002), state-space models (Rangel *et al.*, 2004), and probabilistic graphical models (Friedman, 2004; Friedman *et al.*, 2000; Zhang and Wang, 2010). Some recent works began to address the condition-specific network construction problem. In (Zhang *et al.*, 2009), the authors proposed to identify differential dependency networks between two conditions by testing the significance of changes in local dependencies. Roy *et al.* (2011) extended the condition-specific network learning to multiple conditions. Differential biological network analysis has presented successes in new findings and superiorities over conventional differential analysis such as differential expression (Hudson *et al.*, 2009; Reverter *et al.*, 2010). These methods utilized data in different strategies but haven't made use of the rich domain knowledge.

In parallel, in the past decade, a lot of efforts have also been made to manually curate molecular interactions in cells, such as protein-protein interactions and biological pathways (Kanehisa and Goto, 2000), which can now be conveniently retrieved from relevant biological databases. These biological databases summarize existing knowledge and experimental evidence from multiple sources under diverse conditions, and attempt to delineate a more

detailed picture of the interactome in the cells. Such biological prior information provides rich domain knowledge to the biological network inference problem (Ochs, 2010). Compared with computational inference purely based on data, proper incorporation of biological prior knowledge into network learning algorithms can effectively leverage domain knowledge and make the inference more biologically meaningful. However, as biological prior knowledge is usually aggregated from multiple sources and under diverse experimental settings, direct incorporation of prior knowledge in specific problems is prone to errors or may even lead to biased results.

To efficiently utilize prior knowledge in the network inference while remaining robust to the false positive edges in the knowledge, we here propose a biological prior knowledge guided learning framework to infer condition-specific biological network structures and their topological changes under two different conditions. In this report, we also use graphical models to represent condition-specific biological networks. Two characteristics make graphical models very suitable for representing biological networks: (1) the probabilistic nature of graphical models automatically takes into account the noise in the data and intrinsic uncertainties in the models, and (2) the graphical representations of the models naturally visualize the relationships of genes, which can facilitate new insights and motivate new biological hypotheses. A similar formulation is used in (Ahmed and Xing, 2009), where the time evolvement of network structures is examined with a fused penalty term to encode relationship between adjacent time points, which is substantially different from our formulation where block-wise separable penalties guarantees a fast closed-form solution to the problem. In another work authors combined using copy number data together with gene expression data to infer associations (Yuan *et al.*, 2011). An important character makes our work differ from these efforts is a flexible framework and unique strategy that work together to achieve efficient and robust incorporation of biological knowledge while fully exploring the evidence from data.

1.2 Proposed Approach

We formulate the problem of biological network structure learning and topological change detection with prior knowledge as a convex optimization problem with ℓ_1-penalties, and derive an efficient algorithm to solve it. Further, we propose a sampling scheme to estimate the expected excess risk induced by incorporating random knowledge, which has a maximum entropy distribution over the edges given the number of edges specified in the knowledge. And by controlling this excess risk under the worst-case scenario, our method can efficiently utilize prior knowledge in the network inference while remaining robust to the false positive edges in the knowledge. We propose a significance test scheme to further assign edge-specific p-values to the detected significant rewiring of the learned differential dependency network and detail the manner of edge changes. We test the proposed method on synthetic data to demonstrate the effectiveness of this method, and the simulation results further corroborate our theoretical analysis.

1.3 Outline of the Dissertation

The remainder of this research proposal proceeds as follows. In Chapter 2, we introduce a knowledge-fused differential dependency network (KDDN) method as a general machine learning framework to study the biological dependency network learning problem and identification of structural changes. The framework is suitable for the study of statistically significant topological changes in the transcriptional networks between two biological conditions with prior biological knowledge guidance. In Chapter 3 we extend the application of the methods proposed in Chapter 2 to multiple biological discovery scenarios to demonstrate the strengths of network analysis. We conclude this research proposal with summary of contributions and future work in Chapter 4.

Chapter 2

Knowledge-fused Differential Dependency Network Analysis to Study the Condition-specific Rewiring in Biological Networks

2.1 Introduction

Biological networks are context-specific and dynamic in nature. Under different conditions, different regulatory components and mechanisms are activated and the topology of the underlying biological network changes accordingly. For example, in response to diverse conditions in the yeast, transcription factors alter their interactions and rewire the signaling networks (Luscombe *et al.*, 2004).

It is important to focus on the topological changes in biological networks between disease and normal conditions, or across different stages of cell development. For example, a deviation

from normal regulatory network topology may reveal the mechanism of pathogenesis, and the genes that undergo the most network topological changes may serve as biomarkers for the disease state or as targets for drug discovery or therapeutic intervention. Differential network analysis can also help to identify key genetic players or disease markers. Differential network biology has become an active research area in recent years (Ideker and Krogan, 2012).

In this chapter, we focus on the mathematical models, in particular Gaussian graphical models, that can model the biological networks. We propose a model to formulate the problem of differential network identification across conditions with the capabilities of utilizing existing domain knowledge. Background of the method, novel formulations, algorithms, software implementation and extensive simulation validations are presented.

2.1.1 Gaussian Graphical Models

Graphical models used to model and solve real-world problems have been extensively studied and succeeded across broad areas. It efficiently and clearly represents the conditional independence structure among random variables by factorizing their joint distributions. Bayesian network and Markov random field are two major branches of graphical models, recognized by the edges are directed or undirected (Bishop, 2007).

Graphical models are also very suitable for dealing with biological problems, especially when modeling biological networks (Airoldi, 2007). The probabilistic nature of graphical models automatically takes into account the noise in the data and intrinsic uncertainties in the models, and the graphical representations of the models naturally visualize the relationships of genes, which can facilitate new insights and motivate new biological hypotheses.

Gaussian graphical model is an undirected graphical model in which the variables follows multivariate Gaussian distribution. Given a graph $G = (V, E)$, where $V = 1, \cdots, p$ is

the set of vertex or variables and E is the set of undirected edges. By location and scale transformation, we assume the multivariate Gaussian model is

$$\boldsymbol{X} = (X_1, \cdots, X_p) \sim \mathcal{N}_p(0, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} e^{-\frac{1}{2}\boldsymbol{X}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{X}}, \text{ where } \Sigma_{ii} = 1. \tag{2.1}$$

We denote the inverse of covariance matrix or precision (concentration) matrix by $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$. Note in (2.1), variable pairs corresponding to $\Theta_{ij} = 0, i \neq j$ do not have interaction terms in the joint distribution, implying that they are independent conditioned on the rest of the variables. In graph representations, edges only exist between variables that have corresponding non-zero elements in the precision matrix $\boldsymbol{\Theta}$.

The major task of Gaussian graphical model learning is to infer from data the non-zero pattern of precision matrix $\boldsymbol{\Theta}$. The non-zeros in $\boldsymbol{\Theta}$ determine the graph structure that codes dependencies among variables.

$$X_i \text{ is conditionally independent of } X_j \text{ given others} \iff \Theta_{ij} = 0. \tag{2.2}$$

2.1.2 ℓ_1-regularized Graphical Model Learning Methods

Accurate recovery of the precision matrix is a daunting task. A graph with p nodes has $\mathcal{O}(p^2)$ unique parameters to estimate in the precision matrix. The question is first addressed by Dempster (1972) known as covariance selection by setting subset of Θ_{ij} to zero. The zero Θ_{ij} selection paradigm inherits the principle of parsimony and naturally receives impassioned exploration after the invention of Lasso (Tibshirani, 1996), an ℓ_1-regularized linear regression model, which gives sparse solutions.

Several authors proposed to solve the covariance selection problem as an exact maximum likelihood estimation with ℓ_1 penalties applied to the precision matrix to set components of precision matrix to zero (Banerjee *et al.*, 2006, 2008; Yuan and Lin, 2007a). Friedman

et al. (2008) introduced Graphical Lasso as a fast algorithm to solve the Lasso problem when applied to regularize the precision matrix.

The ℓ_1-regularized maximum likelihood estimation take the form

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta} \succ 0} \log\det \mathbf{\Theta} - \text{trace}(\mathbf{S}\mathbf{\Theta}) - \lambda \|\mathbf{\Theta}\|_1 \tag{2.3}$$

to induce the sparsity in the precision matrix, where $\mathbf{S}$ is the sample covariance matrix.

When we adopt a nodewise view of the Gaussian graphical model, the linear relationships among variables are explicitly defined by Gaussian assumption in (2.1). We regress one variable i against all others:

$$X_i = \sum_{j \neq i} \beta_j^i X_j + \epsilon_i,\ i = 1, 2, \cdots, p, \tag{2.4}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_{\epsilon_i}^2)$, is independent of $X_j; j \neq i$, the linear coefficient $\beta_j^i = -\Theta_{ij}/\Theta_{ii}$. The equivalence of conditional independency in (2.2) can be extended as:

$$X_i \text{ is conditionally independent of } X_j \text{ given others} \Leftrightarrow \Theta_{ij} = 0 \Leftrightarrow \beta_j^i = 0. \tag{2.5}$$

This can be seen by deriving the conditional probability of x_p. Without loss of generalization, we denote p variables follow zero mean multivariate Gaussian distribution:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \sim N(\mathbf{0}, \mathbf{\Sigma}) = C_1 \exp(-\frac{1}{2}\mathbf{X}^T \mathbf{\Sigma}^{-1} \mathbf{X}). \tag{2.6}$$

We always move the node being inferred to the position of x_p in $\mathbf{X}$. Let $\mathbf{X}_{-p} = [x_1, x_2, \cdots x_{p-1}]^T$, and

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{1p} \\ \mathbf{\Sigma}_{p1} & \Sigma_{pp} \end{bmatrix}^{-1}, \tag{2.7}$$

in which $\mathbf{\Sigma}_{11}$ is a $(p-1)\times(p-1)$ matrix.

Using block matrix inverse, we get

$$\begin{aligned}
\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X} =&\mathbf{X}_{-p}^T(\mathbf{\Sigma}_{11}-\mathbf{\Sigma}_{1p}\Sigma_{pp}^{-1}\mathbf{\Sigma}_{p1})^{-1}\mathbf{X}_{-p}-\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p}(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}x_p\\
&-x_p\Sigma_{pp}^{-1}\mathbf{\Sigma}_{p1}(\mathbf{\Sigma}_{11}-\mathbf{\Sigma}_{1p}\Sigma_{pp}^{-1}\mathbf{\Sigma}_{p1})^{-1}\mathbf{X}_{-p}+x_p(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}x_p\\
=&\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p}+\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p}(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}\mathbf{X}_{-p}\\
&-2\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p}(\Sigma_{pp}-\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}x_p+x_p(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}x_p\\
=&\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p}+(x_p-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p})^T(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}(x_p-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p})
\end{aligned} \tag{2.8}$$

The conditional probability of x_p

$$\begin{aligned}
P(x_p|\mathbf{X}_{-p}) =&\frac{P(x_p,\mathbf{X}_{-p})}{P(\mathbf{X}_{-p})}\\
=&\frac{C_1\exp(-\frac{1}{2}\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X})}{C_2\exp(-\frac{1}{2}\mathbf{X}_{-p}^T\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p})}\\
=&C_3\exp(-\frac{1}{2}(x_p-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p})^T(\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})^{-1}(x_p-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p}))\\
\sim&N(x_p-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p},\Sigma_{pp}-\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{1p})
\end{aligned} \tag{2.9}$$

is also Gaussian distributed. So the maximum likelihood estimation of x_p is

$$x_{p(\mathrm{MLE})}=\mathbf{\Sigma}_{p1}\mathbf{\Sigma}_{11}^{-1}\mathbf{X}_{-p}, \tag{2.10}$$

which is boiled down to the linear combination of the remaining nodes.

Following this notion, Meinshausen and Bühlmann (2006) decompose the problem into a series of linear regressions by node with ℓ_1 penalties in the form of

$$\hat{\boldsymbol{\beta}}^i=\arg\min_{\boldsymbol{\beta}^i}(X_i-\sum_{j\neq i}X_j\beta_j^i)^2+\lambda|\boldsymbol{\beta}^i|_1 \tag{2.11}$$

for node i. The composite results of all regressions describe the sparsity pattern of the precision matrix. Having sufficient samples, this approach recovers the nonzero patterns

of precision matrix consistently and can be considered as an approximation of maximum likelihood result (Banerjee *et al.*, 2008).

Zhou *et al.* (2011) combined the use of thresholds to increase zeros after nodewise Lasso regression. The nonzero patterns of recovered precision matrix may not be symmetric due to different regression coefficients of β_i^j and β_j^i, nonetheless Meinshausen and Bühlmann (2006) shows either of AND or OR of β_i^j and β_j^i will both converge to true model given sufficient samples. Peng *et al.* (2009) lumped individual regressions together and enforced the symmetry of β_i^j and β_j^i. Mazumder and Hastie (2012) proposed to threshold covariance matrix to dissect the whole graph into connected components, the separate learning of which will jointly recover the precision matrix in a more efficient way.

2.1.3 Network Topological Change Identification

While the structural learning of Gaussian graphical models has been extensively studied, the change or evolvement of the graph which is also meaningful theoretically and practically has not yet been carefully studied. Jointly learn the graph structure as well as the changes can provide more insights into the underlying mechanisms of the observed system and is especially significant in biological networks modeling.

Differential Dependency Network (Zhang *et al.*, 2009, 2011) is a pioneer method proposed to identify the biological network changes under two conditions. The method uses Lasso to learn local structures under two conditions separately and then test the significance of the changed structures in a following step.

Kolar *et al.* (2009, 2010), Song *et al.* (2009) and Ahmed and Xing (2009) jointly investigated the time varying network structure based on ℓ_1-regularized regression model. ℓ_1-regularized regression has also been used in multiple condition data to find the common network structure rather than changed parts (Hara and Washio, 2011).

2.1.4 Biological Prior Knowledge Incorporation

As the nodewise regression method to infer the structure of Gaussian graphical models gained popularity, some research found that this approach could become inconsistent in certain scenarios (Zou, 2006; Yuan and Lin, 2007b; Zhao and Yu, 2006; Meinshausen and Bühlmann, 2006), *i.e.*, not able to recover the true model asymptotically. The reason is in the formulation (2.11) the ℓ_1 penalties are applied equally to all regressors which may against their underlying importance.

To take into account of the different importance of regressors, Zou (2006) proposed adaptive Lasso to re-weight the ℓ_1 penalty coefficients by the ordinary least square (OLS) regression coefficients to favor the larger ones more. This adaptive modification is proved to assure the consistency of the model.

Keeping in mind that ℓ_1-regularization coefficients could reflect the importance of the regressors, the proper modification of the penalties could potentially reflect the prior information of the precision matrix. However, as pointed out in (Heckerman *et al.*, 2001), such dependency model, the representation of which can be thought of as a collection of regressions among variables in a domain, is difficult to construct using a knowledge-based approach. The current practice of introducing prior knowledge into network learning relies on the Bayesian approached design for specific problems (Werhli and Husmeier, 2007; Jensen *et al.*, 2007; Gao and Wang, 2011). In theses methods the prior knowledge is more a natural framework determined characteristic rather than active pursuit. The knowledge incorporation scheme proposed here is different and unique, which is a general accessory built upon the self-sufficient system to improve the overall performance.

2.2 Problem Formulation

2.2.1 Problem Statement

We represent the condition-specific biological networks as graphs. Here we focus on condition-specific biological network structure and their corresponding structural changes under two conditions, which is a common experiment setting in biomedical research, such as controlled experiments and comparisons between two populations. Suppose there are p nodes (genes) in the network of interest, and we denote the vertex set as V. Let $G^{(1)} = (V, E^{(1)})$ and $G^{(2)} = (V, E^{(2)})$ be the two undirected graphs under the two conditions. $G^{(1)}$ and $G^{(2)}$ have the same vertex set V, and condition-specific edge sets $E^{(1)}$ and $E^{(2)}$. $E^{(1)}$ and $E^{(2)}$ are expected to have considerable overlap, with only a small amount of edges being different. Such edge changes are of particular interest, since such rewiring may reveal pivotal information on how the organisms response to different conditions. We use graph $G = (V, E)$ to represent the composite network of graphs $G^{(1)}$ and $G^{(2)}$ under the two conditions. Edges in E are common edges and unique edges under each conditions. We use a combined asymmetric adjacency matrix to represent G, with the upper triangular matrix indicating the connections in $G^{(1)}$ and lower triangular matrix indicating the connections in $G^{(2)}$.

Biological prior knowledge is collected from biological databases such as KEGG pathway database and Human Protein Reference Database. We denote the biological prior knowledge as a knowledge graph $G_{\mathbf{W}} = (V, E_{\mathbf{W}})$, where the vertex set V is the same set of nodes (genes) and edge set $E_{\mathbf{W}}$ over V is retrieved from biological databases as supported by the existing knowledge or other experimental evidence.

We use matrix $\mathbf{W} \in \Re^{p \times p}$ to represent the prior knowledge, which is the adjacency matrix of $G_{\mathbf{W}}$. The elements of $\mathbf{W}$ are either 1 or 0, with $W_{ij} = 1$ indicating the existence of an edge between the i^{th} gene and j^{th} gene (or their gene products) in the databases, where

$i, j = 1, 2, \cdots, p, i \neq j$. If the prior knowledge is not condition-specific, which is common in most biological databases, the prior knowledge adjacency matrix $\mathbf{W}$ will be symmetric.

The main task in this paper is to infer from data and prior knowledge $G_{\mathbf{W}}$ the condition-specific edge sets E, corresponding to $E^{(1)}$ and $E^{(2)}$, as illustrated in Figure 2.1.

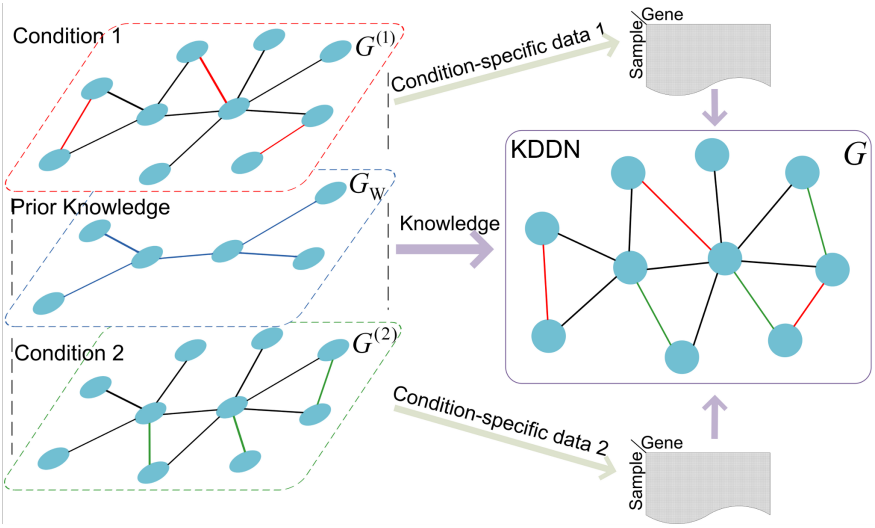


Figure 2.1: KDDN method illustration.

2.2.2 Convex Optimization Formulation

To take advantage of the prior knowledge in the structural learning and avoid the potential bias introduced by knowledge, we formulate the problem into a convex optimization problem with sparsity constraints, and set the proper weights to achieve both the effectiveness of utilizing the domain knowledge and the robustness to the false positives in the knowledge.

We consider the p nodes in V as p random variables, and denote them as $X_1, X_2, \cdots, X_p$. Suppose there are N_1 samples under condition 1 and N_2 samples under condition 2. Without loss of generality, we assume $N_1 = N_2 = N$. Under the first condition, for variable X_i, we have observations $\boldsymbol{x}_i^{(1)} = [x_{1i}^{(1)}, x_{2i}^{(1)}, \cdots, x_{Ni}^{(1)}]^T$, $i = 1, 2, \cdots, p$, while under the sec-

ond condition, we have $\boldsymbol{x}_i^{(2)} = [x_{1i}^{(2)}, x_{2i}^{(2)}, \cdots, x_{Ni}^{(2)}]^T$, $i = 1, 2, \cdots, p$. Further, let $\mathbf{X}^{(1)} = [\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_p^{(1)}]$ be the data matrix under condition 1 and $\mathbf{X}^{(2)} = [\boldsymbol{x}_1^{(2)}, \boldsymbol{x}_2^{(2)}, \cdots, \boldsymbol{x}_p^{(2)}]$ be the data matrix under condition 2.

Denote

$$\boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{x}_i^{(1)} \\ \boldsymbol{x}_i^{(2)} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}^{(2)} \end{bmatrix}, \tag{2.12}$$

and

$$\boldsymbol{\beta}_i = \begin{bmatrix} \boldsymbol{\beta}_i^{(1)} \\ \boldsymbol{\beta}_i^{(2)} \end{bmatrix} = [\beta_{1i}^{(1)}, \beta_{2i}^{(1)}, \cdots, \beta_{pi}^{(1)}, \beta_{1i}^{(2)}, \beta_{2i}^{(2)}, \cdots, \beta_{pi}^{(2)}]^T. \tag{2.13}$$

Following our previous work in (Zhang and Wang, 2010), we formulate the problem of learning structural changes between two conditions as a convex optimization problem. Network structures under two conditions as well as their changes are simultaneously obtained by solving the optimization problem for each node (variable) X_i, $i = 1, 2, \cdots, p$, $j = 1, 2, \cdots, p$, with the objective function

$$\begin{aligned} f(\boldsymbol{\beta}_i) = \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 + \lambda_1 \sum_{j=1}^{p} (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) \\ + \lambda_2 \|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1. \end{aligned} \tag{2.14}$$

The solution is obtained by minimizing (2.14),

$$\begin{aligned} \hat{\boldsymbol{\beta}}_i &= \arg\min_{\boldsymbol{\beta}_i} f(\boldsymbol{\beta}_i) \\ &= \arg \min_{\boldsymbol{\beta}_i^{(1)}, \boldsymbol{\beta}_i^{(2)}} \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 \\ &+ \lambda_1 \sum_{j=1}^{p} (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2 \|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1 \\ &\text{s.t. } \beta_{ii}^{(1)} = 0, \beta_{ii}^{(2)} = 0. \end{aligned} \tag{2.15}$$

In (2.15), the structures of the graphical model under two conditions are learned jointly. The first weighted ℓ_1-regularization term leads to the identification of sparse graph structure. The second ℓ_1-regularization term, $\lambda_2\|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1$, encourages sparse changes in the model structure and parameters between two conditions, and thereby suppresses the structural and parametric inconsistencies due to noise in the data and limited samples.

As discussed in Section 2.3.1, the problem (2.15) can be solved efficiently by the block coordinate descent algorithm proposed. We repeat this procedure to each node X_i, $i = 1, 2, \cdots, p$. The non-zero elements of $\boldsymbol{\beta}_i^{(1)}$ indicate the neighbors of the i^{th} node under the first condition and the non-zero elements of $\boldsymbol{\beta}_i^{(2)}$ indicate the neighbors of the i^{th} node under the second condition. In biological network modeling, we are particularly interested in where and how the network exhibits different topologies between two conditions. The different connections between two conditions are highlighted to indicate such changes.

2.2.3 Prior Knowledge Incorporation

The prior knowledge is explicitly incorporated into the formulation by W_{ji} and θ in the weighted ℓ_1-regularization term, $\lambda_1 \sum_{j=1}^{p}(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|)$. The non-zero elements in $\mathbf{W}$ introduce knowledge to the objective function (2.14). θ is a ℓ_1 penalty relaxation parameter taking value in $[0, 1]$, which works with $\mathbf{W}$ to reduce the penalties on the edges with supporting evidence in the prior knowledge while having no effects on edges with no knowledge.

θ determines to what degree the knowledge will affect the inference. Let us consider two extreme cases: when $\theta = 0$, the algorithm ignores all knowledge information and infers the network structures solely based on data, which is equivalent to the algorithm in (Zhang and Wang, 2010); on the other hand, when $\theta = 1$, the edge between X_j and X_i will always be included if such an edge exists in the prior knowledge, which implies the prior knowledge will

have a determinant effect on the network inference. Therefore the prior knowledge incorporation needs to find a proper balance between the experimental data and prior knowledge. A proper θ will reduce the penalty applied to $\beta_{ji}^{(c)}$ corresponding to the connection between X_j and X_i with existing prior knowledge. As a result, the connection between X_j and X_i will more likely be detected and therefore reduce the number of missing edges.

From a Bayesian perspective, as pointed out in (Tibshirani, 1996), the ℓ_1 penalty term $\|\boldsymbol{\beta}_i^{(c)}\|_1$ is equivalent to independent Laplace priors for the $\beta_{ji}^{(c)}$, $j = 1, 2, ..., p$, $c = 1, 2$, which follow

$$pdf(\beta_{ji}^{(c)}) = \frac{1}{2b} \exp\big(- \frac{-|\beta_{ji}^{(c)}|}{b}\big) \tag{2.16}$$

where $b = \frac{1}{\lambda_1(1-W_{ji}\theta)}$. Note that the Laplace distribution with a larger b has a larger variance $(2b^2)$ and thus distributes less mass around zero and more mass in the two tails. When supporting evidence for the edge between node i and node j is present in the prior knowledge, a non-zero θ adjusts the prior distribution for this edge $(\beta_{ji}^{(c)})$ with a larger b, and thereby makes this edge more likely be detected.

We want to effectively incorporate the useful information in the prior knowledge, while at the same time, we shall be able to limit the adverse effects caused by the spurious edges in the prior knowledge. Here we choose a strategy to control such adverse effects incurred in the worst-case scenario under which the given prior knowledge is totally random. In this case, the entropy of the knowledge distribution over the edges is maximized and the information introduced to the inference is minimal. The maximum entropy distribution of the random prior knowledge is discussed and proved in the supplementary information. Incorporated with such random knowledge, the inference results will deviate from the purely data driven result. Then, θ is carefully chosen so that the expected deviation is controlled within acceptable range in the worst-case scenario, which guarantees the robustness when the prior knowledge is highly inconsistent with the underlying ground-truth. When the prior knowledge is more consistent with ground-truth than the random prior knowledge, the

inference result will be better than the worst-case, and the more consistent the closer the inference result is to ground-truth.

We use graph edit distance between two adjacency matrices as a measurement for the dissimilarity of two graphs. The graph edit distance used here is a simplified case where we count the number of different edges, essentially a Hamming distance calculation. Let $G_{\mathbf{T}} = (V, E_{\mathbf{T}})$ denote the ground-truth graph with edge set $E_{\mathbf{T}}$, $G_{\mathbf{X}} = (V, E_{\mathbf{X}})$ denote the graph learned purely from data, *i.e.* $\mathbf{W} = \mathbf{0}$, and $G_{\mathbf{X},\mathbf{W}_R,\theta}(V, E_{\mathbf{X},\mathbf{W}_R,\theta})$ denote the graph learned with prior knowledge. $\mathbf{W}_R$ indicates that the prior knowledge is "random". Let $d(G_*, G_\star)$ denote the graph edit distance between two graphs. Further, let $|E_*|$ be the number of edges in the graph G_*.

The inference error rate associated with the purely data result $G_{\mathbf{X}}$ is $\frac{d(G_{\mathbf{T}},G_{\mathbf{X}})}{|E_{\mathbf{T}}|}$, which is an unknown baseline when determining the degree of knowledge incorporation θ. Even if the prior knowledge is the worst case, a proper θ is expected to control the increase in the error rate or excess risk within an acceptable range.

Since $G_{\mathbf{T}}$ is unknown, we instead control the excess risk indirectly by evaluating the effect of random knowledge against $G_{\mathbf{X}}$, the purely data-driven inference result. To be more specific, we use a sampling-based algorithm to find the empirical distribution of $d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})$, and choose the largest $\theta \in [0, 1]$ that satisfies:

$$\begin{aligned} &\hat{\theta} = \max \theta \\ &\text{s.t.} \quad \frac{\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]}{|E_{\mathbf{X}}|} \leq \delta. \end{aligned} \tag{2.17}$$

A natural question is whether using $G_{\mathbf{X}}$ instead of $G_{\mathbf{T}}$ to control the excess risk induced by random knowledge is legitimate. To answer this question, we show in Theorem 1 that the θ obtained in (2.17) in fact controls an upper bound of $\frac{\mathbb{E}[d(G_{\mathbf{T}},G_{\mathbf{X},\mathbf{W}_{\mathbf{R}},\theta})]}{|E_{\mathbf{T}}|}$, *i.e.* the increase in the network inference error rate induced by random prior knowledge (the worst-case scenario), under the assumption that the number of false negatives (FN) in $G_{\mathbf{X}}$ is no smaller than the

number of false positives (FP). We can always set the λ to get conservative results. The direct effect of knowledge guidance is to make the false negatives easier to be identified and at the same time reduce false positives indirectly. This theorem is illustrated in Figure 2.2.

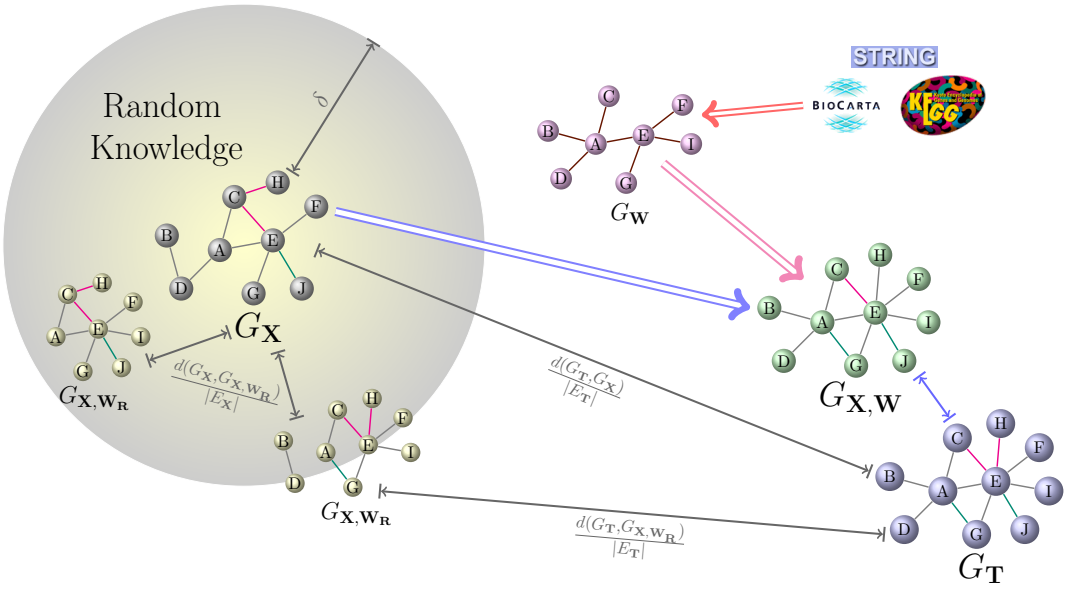


Figure 2.2: Illustration of Theorem 1. Networks are represented by connected clusters of nodes, with gray edges indicating common connections under both conditions, green edges indicating connections uniquely exist under one condition and red edges indicating connections uniquely exist under the other condition. $G_{\mathbf{T}}$ is the underlying ground-truth network. $G_{\mathbf{X}}$ is the network learned purely from data. δ will control the increase in the error rate induced by random knowledge within the shaded region. By incorporating prior knowledge with good quality, the learning result $G_{\mathbf{X},\mathbf{W}}$ can be significantly improved.

Theorem 1. *For a given $\delta \in [0, 1)$, if the prior knowledge incorporation parameter θ satisfies the inequality*

$$\frac{\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]}{|E_{\mathbf{X}}|} \leq \delta, \tag{2.18}$$

then the excess risk introduced by incorporating random prior knowledge is bounded by δ,

more specifically,

$$\frac{\mathbb{E}[d(G_{\mathbf{T}}, G_{\mathbf{X},\mathbf{W_R},\theta})]}{|E_{\mathbf{T}}|} \leq \frac{d(G_{\mathbf{T}}, G_{\mathbf{X}})}{|E_{\mathbf{T}}|} + \delta. \tag{2.19}$$

Proof. The graph edit distance between $G_{\mathbf{X}}$ and $G_{\mathbf{T}}$ is

$$d(G_{\mathbf{T}}, G_{\mathbf{X}}) = FP + FN, \tag{2.20}$$

in which FP and FN are number of false positives and false negatives.

The relationship between $|E_{\mathbf{X}}|$ and $|E_{\mathbf{T}}|$ is

$$|E_{\mathbf{X}}| = |E_{\mathbf{T}}| + FP - FN. \tag{2.21}$$

The deviation of learned network $G_{\mathbf{X},\mathbf{W}_R,\theta}$ after knowledge incorporation from purely data-driven result $G_{\mathbf{X}}$ is expected to have a newly identified edges and b disappeared edges,

$$\mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})] = a + b. \tag{2.22}$$

Denote the expected number of false positives and false negatives of $G_{\mathbf{X},\mathbf{W}_R,\theta}$ compared to the ground-truth $G_{\mathbf{T}}$ as FP' and FN'. We have

$$FP' \leq FP + a, \tag{2.23}$$

$$FN' \leq FN + b. \tag{2.24}$$

Therefore,

$$\begin{aligned}
\frac{\mathbb{E}[d(G_{\mathbf{T}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]}{|E_{\mathbf{T}}|} &= \frac{FP' + FN'}{|E_{\mathbf{T}}|} \\
&\leq \frac{FP + FN + a + b}{|E_{\mathbf{T}}|} \\
&= \frac{d(G_{\mathbf{T}}, G_{\mathbf{X}}) + \mathbb{E}[d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})]}{|E_{\mathbf{T}}|} \\
&\leq \frac{d(G_{\mathbf{T}}, G_{\mathbf{x}})}{|E_{\mathbf{T}}|} + \delta \frac{|E_{\mathbf{x}}|}{|E_{\mathbf{T}}|} \\
&= \frac{d(G_{\mathbf{T}}, G_{\mathbf{x}})}{|E_{\mathbf{T}}|} + \delta \Big(\frac{|E_{\mathbf{T}}| + FP - FN}{|E_{\mathbf{T}}|} \Big) \\
&\leq \frac{d(G_{\mathbf{T}}, G_{\mathbf{x}})}{|E_{\mathbf{T}}|} + \delta
\end{aligned} \tag{2.25}$$

Q.E.D.

□

An important property that governs (2.17) and facilitates the search of θ in the later Section 2.3.2 is the monotonicity between θ and the excess risk related to δ. We show this monotonicity empirically using an experiment on a network with 100 nodes and 100 samples. We gradually increase θ and calculate the corresponding excess risk with 1000 random realizations. The excess risk with respect to θ is shown in Figure 2.3.

To prove the random knowledge generation process has the maximum entropy, let $\mathbf{W} \in \mathbb{R}^{p \times p}$ be the (symmetric) adjacency matrix that encodes the "random" prior knowledge. There are M edges in the knowledge and therefore $2M$ elements of $\mathbf{W}$ are "1" and the remaining elements of $\mathbf{W}$ are "0".

Since $\mathbf{W}$ is symmetric, we rearrange the elements of the upper triangle of $\mathbf{W}$ in a vector form, denoted by $\mathbf{x} \in \mathbb{R}^{p(p-1)/2}$. Each element of $\mathbf{x}$ takes values in $\{0, 1\}$. Now we want to find the maximum entropy distribution of $\mathbf{x}$, $P(\mathbf{x})$, given that there are *exactly* M "1" in $\mathbf{x}$.

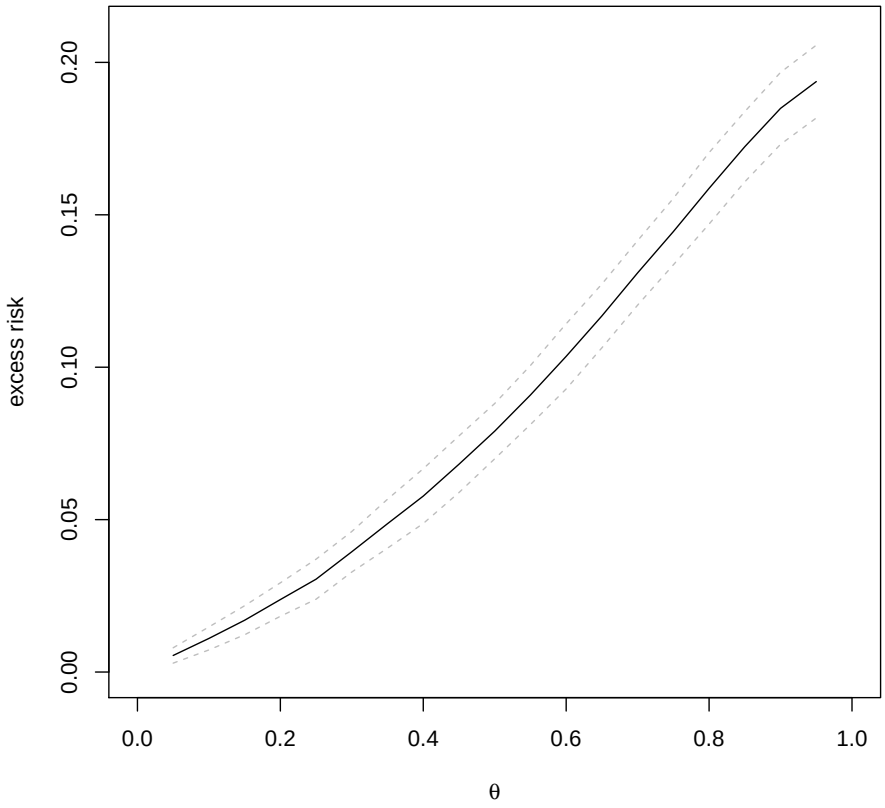


Figure 2.3: The excess risk with respect to θ. The solid line is the average excess risk, and the dashed lines mark the one standard deviation. The monotonicity is between θ and excess risk is verified.

We have

$$\begin{aligned} \underset{P(\mathbf{x})}{\text{maximize}} \quad & H(\mathbf{x}) \\ \text{s.t.} \quad & \sum \mathbf{x}[i] = M \end{aligned} \tag{2.26}$$

The number of possible values taken by $\mathbf{x}$ is $2^{p(p-1)/2}$. However, the number of feasible values that satisfy the equality constraint is $C^M_{p(p-1)/2}$. Denote the set of feasible values of $\mathbf{x}$ as $\mathbb{X}'$. Therefore, the support of the $P(\mathbf{x})$ for the above maximum entropy problem is $\mathbb{X}'$. The

entropy of $P(\mathbf{x})$ for the above problem is:

$$H(\mathbf{x}) = -\sum_{\mathbf{x}\in\mathbb{X}'} P(\mathbf{x}) \log P(\mathbf{x}) \tag{2.27}$$

Applying Thereom 2.6.4 in (Cover and Thomas, 2006), $H(\mathbf{x})$ obtains its maximum when $\mathbf{x}$ is uniformly distributed over $\mathbb{X}'$.

Therefore the maximum entropy distribution for the "random" knowledge is

$$P(\mathbf{x}) = \begin{cases} \frac{1}{C_{p(p-1)/2}^{M}}, & \mathbf{x} \in \mathbb{X}', \\ 0, & \text{otherwise.} \end{cases} \tag{2.28}$$

2.3 Algorithms

In this section we propose a block coordinate descent algorithm to solve the optimization problem in (2.15), and derive the corresponding closed-form solution for the sub-problem. Then we introduce a sampling and estimation algorithm to determine the degree of knowledge incorporation. Choice of penalty parameters λ_1 and λ_2 is also discussed.

2.3.1 Block Coordinate Descent Algorithm

Although the optimization problems with ℓ_1-regularization can be solved readily by existing convex optimization techniques, a lot of efforts have been made to solve the problems efficiently by exploiting the special structures of the problems. Recently, coordinate-wise descent algorithms have been studied in Lasso related problems, such as Lasso, garotte and elastic net (Friedman *et al.*, 2007). It is shown in (Friedman *et al.*, 2008) with experiments that a coordinate descent procedure for Lasso, graphical Lasso, is 30-4000 times faster than competing methods, making it a computationally attractive method.

We adopt this idea since in our formulation the penalty term $\lambda_1 \sum_{j=1}^{p}(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2 \|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1$ has block-wise separability (Tseng, 2001). We propose a block coordinate descent algorithm to solve the optimization problem (2.15) for each node X_i, $i = 1, 2, \cdots, p$.

The objective function (2.14) can be rewritten as

$$f(\boldsymbol{\beta}_i) = \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 + \lambda_1 \sum_{j=1}^{p}(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2 \sum_{j=1}^{p}(|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}|). \tag{2.29}$$

The non-differentiable part of $f(\boldsymbol{\beta}_i)$ can be written as the sum of p terms with non-overlapping members, $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$, $j = 1, 2, ..., p$. Each $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$, $j = 1, 2, ..., p$, is a coordinate block. This property is essential to guaranteeing the convergence of the block coordinate descent algorithm.

The essence of the block coordinate descent algorithm is "one-block-at-a-time". At iteration $r+1$, only one coordinate block, $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$, is updated, with the remaining $(\beta_{li}^{(1)}, \beta_{li}^{(2)})$, $l \neq j$, fixed at their values at iteration r. Given

$$\boldsymbol{\beta}_i^r = [\beta_{1i}^{(1),r}, \beta_{2i}^{(1),r}, \cdots, \beta_{pi}^{(1),r}, \beta_{1i}^{(2),r}, \beta_{2i}^{(2),r}, \cdots, \beta_{pi}^{(2),r}]^T, \tag{2.30}$$

at iteration $r + 1$, the estimation is updated according to the following sub-problem

$$\begin{aligned} &\boldsymbol{\beta}_i^{r+1} = \arg\min_{\boldsymbol{\beta}_i} f(\boldsymbol{\beta}_i) \\ &\text{s.t. } \beta_{li}^{(1)} = \beta_{li}^{(1),r}, \beta_{li}^{(2)} = \beta_{li}^{(2),r}, \\ &\qquad \text{for } l = 1, 2, \cdots, p, l \neq j. \end{aligned} \tag{2.31}$$

We use a cyclic rule to solve the sub-problem and update parameter estimation iteratively, *i.e.*, update parameter pair $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$ at iteration $r + 1$, and $j = ((r + 1) \mod p) + 1$.

For each iteration the closed-form solution to the sub-problem exists and can be computed efficiently.

For notational simplicity, we can always normalize the variables to mean 0 and unit length by location and scale transformations,

$$\begin{aligned} \sum_{k=1}^{N} x_{ki}^{(1)} = 0, \qquad & \sum_{k=1}^{N} (x_{ki}^{(1)})^2 = 1, \\ \sum_{k=1}^{N} x_{ki}^{(2)} = 0, \qquad & \sum_{k=1}^{N} (x_{ki}^{(2)})^2 = 1, \end{aligned} \tag{2.32}$$

where $i = 1, 2, \cdots, p$. Here we assume this normalization step has already been performed. Additionally, the orthogonality between j^{th} column of matrix $\mathbf{X}$, $\boldsymbol{x}_j$ and the $(j+p)^{\text{th}}$ column of $\mathbf{X}$, $\boldsymbol{x}_{j+p}$ simplifies the derivation of closed-form solutions to the sub-problems in each iterations of the block coordinate descent.

Since $\beta_{li}^{(1)}$ and $\beta_{li}^{(2)}$, $l = 1, 2, \cdots, p, l \neq j$, are fixed during iteration $r+1$, we rewrite the objective function of (2.31) as

$$\begin{aligned} & \tilde{f}(\boldsymbol{\beta}_i) \\ =& \frac{1}{2} \| \boldsymbol{y}_i - \sum_{l \neq i,j} \boldsymbol{x}_l \beta_{li}^{(1),r} - \sum_{l \neq i,j} \boldsymbol{x}_{p+l} \beta_{li}^{(2),r} \\ & \qquad - \boldsymbol{x}_j \beta_{ji}^{(1)} - \boldsymbol{x}_{p+j} \beta_{ji}^{(2)} \|_2^2 \\ & + \lambda_1 \sum_{l \neq i,j} (1 - W_{li}\theta)(|\beta_{li}^{(1),r}| + |\beta_{li}^{(2),r}|) \\ & + \lambda_2 \sum_{l \neq i,j} (|\beta_{li}^{(1),r} - \beta_{li}^{(2),r}|) \\ & + \lambda_1 (1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2 (|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}|) \end{aligned} \tag{2.33}$$

Let

$$\tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \sum_{l \neq i,j} \boldsymbol{x}_l \beta_{li}^{(1),r} - \sum_{l \neq i,j} \boldsymbol{x}_{p+l} \beta_{li}^{(2),r} \tag{2.34}$$

Therefore, updating $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$ is equivalent to

$$\begin{aligned}
&(\beta_{ji}^{(1),r+1}, \beta_{ji}^{(2),r+1}) \\
=&\arg\min_{\beta_{ji}^{(1)},\beta_{ji}^{(2)}} \tilde{f}(\boldsymbol{\beta}_i) \\
=&\arg\min_{\beta_{ji}^{(1)},\beta_{ji}^{(2)}} \frac{1}{2}\|\tilde{\boldsymbol{y}}_i - \boldsymbol{x}_j\beta_{ji}^{(1)} - \boldsymbol{x}_{p+j}\beta_{ji}^{(2)}\|_2^2 \\
&+ \lambda_1(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2(|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}|)
\end{aligned} \tag{2.35}$$

Denote

$$\rho_1 = \tilde{\boldsymbol{y}}_i^T \cdot \boldsymbol{x}_j, \tag{2.36}$$

$$\rho_2 = \tilde{\boldsymbol{y}}_i^T \cdot \boldsymbol{x}_{p+j}. \tag{2.37}$$

First, we examine a simple case, the solution, $(\beta_{ji}^{(1)}, \beta_{ji}^{(2)})$, satisfies

$$\begin{cases}
\beta_{ji}^{(1)} > 0, \\
\beta_{ji}^{(2)} > 0, \\
\beta_{ji}^{(1)} < \beta_{ji}^{(2)}.
\end{cases} \tag{2.38}$$

Take derivative of objective function (2.33), and we have

$$\begin{aligned}
\frac{\partial \tilde{f}}{\partial \beta_{ji}^{(1)}} = &\beta_{ji}^{(1)} - \rho_1 + \lambda_1(1 - W_{ji}\theta)\mathrm{sgn}(\beta_{ji}^{(1)}) \\
&+ \lambda_2\mathrm{sgn}(\beta_{ji}^{(1)} - \beta_{ji}^{(2)}),
\end{aligned} \tag{2.39}$$

$$\begin{aligned}
\frac{\partial \tilde{f}}{\partial \beta_{ji}^{(2)}} = &\beta_{ji}^{(2)} - \rho_2 + \lambda_1(1 - W_{ji}\theta)\mathrm{sgn}(\beta_{ji}^{(2)}) \\
&+ \lambda_2\mathrm{sgn}(\beta_{ji}^{(1)} - \beta_{ji}^{(2)}),
\end{aligned} \tag{2.40}$$

where $\mathrm{sgn}(\cdot)$ is the sign function.

When $\rho_1 > \lambda_1(1 - W_{ji}\theta) - \lambda_2$ and $\rho_2 > \rho_1 + 2\lambda_2$, we have

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 - \lambda_1(1 - W_{ji}\theta) + \lambda_2, \\ \beta_{ji}^{(2)} = \rho_2 - \lambda_1(1 - W_{ji}\theta) - \lambda_2. \end{cases} \tag{2.41}$$

Similarly, we derive all closed-form solutions to problem (2.31), depending on the values of ρ_1, ρ_2 with respect to $\lambda_1(1 - W_{ji}\theta), \lambda_2$. The plane (ρ_1, ρ_2) is divided into 13 regions, as shown in Figure 2.4.

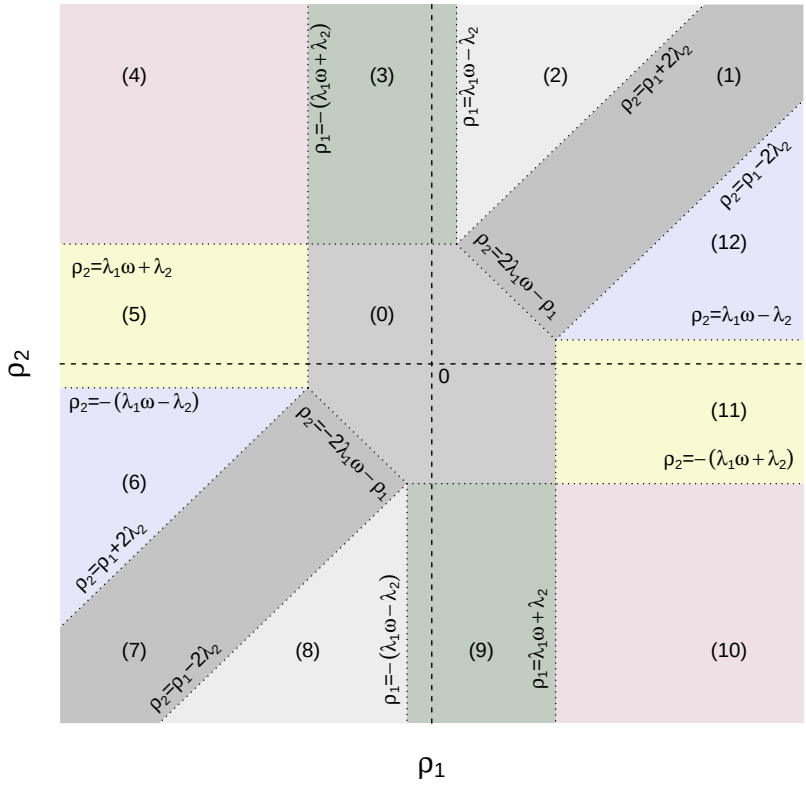


Figure 2.4: Solution regions of the sub-problem (2.31), $\omega = 1 - W_{ji}\theta$.

Depending on the location of (ρ_1, ρ_2) in the plane, the solutions to problem (2.31) are as follows.

If (ρ_1, ρ_2) is in region (0), then

$$\beta_{ji}^{(1)} = \beta_{ji}^{(2)} = 0. \tag{2.42}$$

If (ρ_1, ρ_2) is in region (1), then

$$\beta_{ji}^{(1)} = \beta_{ji}^{(2)} = \frac{1}{2}(\rho_1 + \rho_2) - \lambda_1(1 - W_{ji}\theta) \tag{2.43}$$

If (ρ_1, ρ_2) is in region (2), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 - \lambda_1(1 - W_{ji}\theta) + \lambda_2, \\ \beta_{ji}^{(2)} = \rho_2 - \lambda_1(1 - W_{ji}\theta) - \lambda_2. \end{cases} \tag{2.44}$$

If (ρ_1, ρ_2) is in region (3), then

$$\begin{cases} \beta_{ji}^{(1)} = 0, \\ \beta_{ji}^{(2)} = \rho_2 - \lambda_1(1 - W_{ji}\theta) - \lambda_2. \end{cases} \tag{2.45}$$

If (ρ_1, ρ_2) is in region (4), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 + \lambda_1(1 - W_{ji}\theta) + \lambda_2, \\ \beta_{ji}^{(2)} = \rho_2 - \lambda_1(1 - W_{ji}\theta) - \lambda_2. \end{cases} \tag{2.46}$$

If (ρ_1, ρ_2) is in region (5), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 + \lambda_1(1 - W_{ji}\theta) + \lambda_2, \\ \beta_{ji}^{(2)} = 0. \end{cases} \tag{2.47}$$

If (ρ_1, ρ_2) is in region (6), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 + \lambda_1(1 - W_{ji}\theta) + \lambda_2, \\ \beta_{ji}^{(2)} = \rho_2 + \lambda_1(1 - W_{ji}\theta) - \lambda_2. \end{cases} \tag{2.48}$$

If (ρ_1, ρ_2) in region (7), then

$$\beta_{ji}^{(1)} = \beta_{ji}^{(2)} = \frac{1}{2}(\rho_1 + \rho_2) + \lambda_1(1 - W_{ji}\theta). \tag{2.49}$$

If (ρ_1, ρ_2) is in region (8), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 + \lambda_1(1 - W_{ji}\theta) - \lambda_2, \\ \beta_{ji}^{(2)} = \rho_2 + \lambda_1(1 - W_{ji}\theta) + \lambda_2. \end{cases} \tag{2.50}$$

If (ρ_1, ρ_2) is in region (9), then

$$\begin{cases} \beta_{ji}^{(1)} = 0, \\ \beta_{ji}^{(2)} = \rho_2 + \lambda_1(1 - W_{ji}\theta) + \lambda_2. \end{cases} \tag{2.51}$$

If (ρ_1, ρ_2) is in region (10), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 - \lambda_1(1 - W_{ji}\theta) - \lambda_2 \\ \beta_{ji}^{(2)} = \rho_2 + \lambda_1(1 - W_{ji}\theta) + \lambda_2. \end{cases} \tag{2.52}$$

If (ρ_1, ρ_2) is in region (11), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 - \lambda_1(1 - W_{ji}\theta) - \lambda_2 \\ \beta_{ji}^{(2)} = 0. \end{cases} \tag{2.53}$$

If (ρ_1, ρ_2) is in region (12), then

$$\begin{cases} \beta_{ji}^{(1)} = \rho_1 - \lambda_1(1 - W_{ji}\theta) - \lambda_2 \\ \beta_{ji}^{(2)} = \rho_2 - \lambda_1(1 - W_{ji}\theta) + \lambda_2. \end{cases} \tag{2.54}$$

We summarize the block coordinate descent optimization procedure to solve problem (2.15) in Algorithm 1.

The convergence of the algorithm is guaranteed by Theorem 4.1 proposed by Tseng (2001).

The block coordinate descent algorithm explores the special structure of the problem and avoided conventional convex optimization algorithm, largely improved the computational

Algorithm 1 Block coordinate descent algorithm to solve problem (2.15)

Initialization: $\boldsymbol{\beta}_i^0 = [0, 0, ..., 0]$, $r = 0$

while $\boldsymbol{\beta}_i^r$ is not converged **do**

$j \leftarrow (r \mod p) + 1$

if $j \neq i$ **then**

Let $\beta_{li}^{(1),r+1} = \beta_{li}^{(1),r}$, $\beta_{li}^{(2),r+1} = \beta_{li}^{(2),r}$, $l \neq j$

Solve the j^{th} sub-problem using (2.34), (2.36), (2.37) and (2.42)-(2.54).

end if

$r \leftarrow r + 1$

end while

efficiency. But the network inference itself is notoriously difficult and the overall computation of KDDN is still complex.

Let's keep the assumption of p nodes and N samples for both conditions. The block coordinate descent algorithm circulate among the p nodes, and in each iteration there are a number of multiplications and additions performed linear to N. Assuming the multiplication and addition can finish in $\mathcal{O}(1)$ time, and the block coordinate descent algorithm converge within linear constant number of iterations, the computational complexity of block coordinate descent algorithm is $\mathcal{O}(pN)$. Because we need to solve for all p nodes, the overall computational complexity of KDDN is $\mathcal{O}(p^2N)$.

2.3.2 Degree of Prior Knowledge Incorporation

We want to select a proper θ that efficiently utilizes the information in the prior knowledge while remaining robust to the spurious edges in the knowledge. We estimate the expected normalized deviation under random knowledge by a sampling algorithm, and solve (2.17) subsequently.

When $\theta = 0$, any input knowledge is ignored. Intuitively, larger θ yields deeper influence of knowledge, and the result with knowledge will deviate farther. We studied the relation between θ and the deviation through empirical experiment, which showed consistency with intuition. This monotonicity helps to save the computation time.

Given the number of edges specified in the prior knowledge, M, $M = 0, \cdots, C_p^2/2$, we randomly sample M edges from all possible edges formed by pairs of vertex in V as "random" prior knowledge to form a $\mathbf{W}_R$, and solve the optimization problem (2.15) to learn the condition-specific network with prior knowledge. The deviation of the network with prior knowledge from the network inferred purely based on data is calculated according the method introduced in Section 2.2.3. By repeating this process B times (B is set to 1000 in this report), we can obtain the empirical distribution of this deviation. Based on this empirical distribution, we use its average in the estimate of the expected excess risk. We use binary search to find the θ as the solution of problem (2.17) efficiently. To further speed up the computation we parallelize the sampling process to fully utilized the nowadays ubiquitous multi-core computing facilities.

We summarize the steps above in Algorithm 2. This procedure is computationally intensive, and therefore the computational efficiency of Algorithm 1 becomes indispensable in the development of this approach.

2.3.3 Choice of Model Parameters

In problem formulation (4), the first ℓ_1-regularization term, $\lambda_1 \sum_{j=1}^{p}(1-W_{ji}\theta)(|\beta_{ji}^{(1)}|+|\beta_{ji}^{(2)}|)$, leads to the identification of sparse graph structures. Let $\mathbf{W} = \mathbf{0}$ or $\theta = 0$ to examine the performance based solely on data, which regresses to the problem in (Zhang and Wang, 2010), λ_1 is expected to yield a network neither too sparse nor too dense. One approach to determine λ_1 under Gaussian assumption and $\lambda_2 = 0$ is by controlling the risk of falsely

Algorithm 2 Sampling and estimation method to solve problem (2.17)

Inputs: HIGH=0.9, LOW=0.1, MID=(HIGH-LOW)/2+LOW, $B(=1000), M$

Initialization: θ =MID, $D=0$

Solve problem (2.15) with $\mathbf{W}=\mathbf{0}$ using Algorithm 1, get $G_{\mathbf{X}}$, $|E_{\mathbf{X}}|$

while HIGH-LOW<0.01 **do**

for $i=1$ **to** B **do**

Let $\mathbf{W}_R=\mathbf{0}$

Sample M elements in the upper triangle of $\mathbf{W}_R$, $(a_j,b_j), j=1,2,\cdots,M$

Let $\mathbf{W}_R(a_j,b_j)=1, \mathbf{W}_R(b_j,a_j)=1$

Solve problem (2.15) with $\mathbf{W}_R$ using Algorithm 1, get $G_{\mathbf{W}_R}$

Calculate $d_i=d(G_{\mathbf{X}}, G_{\mathbf{X},\mathbf{W}_R,\theta})$

Let $D=D+d_i$

end for

if $\frac{D}{B|E_{\mathbf{X}}|}>\delta$ **then**

HIGH=MID

else

LOW=MID

end if

MID=(HIGH-LOW)/2+LOW

end while

Output: θ =MID

extending connectivity to distinct components in the graph no larger than α_1 (which is typically set to 0.05),

$$\lambda_1 = \frac{2}{N}\left(1 - \Phi(\frac{\alpha_1}{2p^2})\right), \tag{2.55}$$

which is guaranteed by Theorem 3 in (Meinshausen and Bühlmann, 2006).

The second ℓ_1-regularization term, $\lambda_2\|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1$, works specifically on differential edges to suppress inconsistencies of the network structures and parameters between two conditions. At a given significance level α_2, (e.g., $\alpha_2 = 0.05$ is used in this paper), only differential edges that are statistically significant are expected to enter the differential dependency network. So the λ_2 corresponding to α_2 is found by putting the type I error rate under null distribution in the vicinity of α_2 using batches of permuted samples. Suppose the size of the network or the number of edges in the network is E under null distribution, the expected type I error rate should be α_2, *i.e.*, there are still $\alpha_2 E$ edges falsely claimed as differential edges. λ_2 is then found by gradually increasing its value from 0 until the type I error rate falls into the vicinity of α_2 under null distribution to guarantee desired detection power.

2.3.4 Significance Assessment

The overall significance of learned differential dependency network is determined by the parameter λ_2 set in section 2.3.3. But there is no distinction among p-values of differential edges. This section provides a p-value assessment using permutation test as a second step after a screening of differential edges at the same significance level to give finer resolution of network changes while ruling out wrongly claimed differential edges.

With parameters λ_1 and λ_2 chosen in section 2.3.3, the connections for node i is determined

by the regression coefficient β_{ji} between nodes i and $j, j \neq i$,

$$\hat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta}_i^{(1)}, \boldsymbol{\beta}_i^{(2)}} \frac{1}{2}\|\boldsymbol{y}_i - \mathbf{X}\boldsymbol{\beta}_i\|_2^2 + \lambda_1 \sum_{j=1}^{p}(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2\|\boldsymbol{\beta}_i^{(1)} - \boldsymbol{\beta}_i^{(2)}\|_1$$
$$\text{s.t. } \beta_{ii}^{(1)} = 0, \beta_{ii}^{(2)} = 0. \tag{2.56}$$

This problem can be solved by Algorithm 1 through iteratively updating a pair of $\beta_{ji}^{(1)}$ and $\beta_{ji}^{(2)}$ while keeping other pairs fixed.

$$\begin{aligned}(\beta_{ji}^{(1),r+1}, \beta_{ji}^{(2),r+1}) = &\arg\min_{\beta_{ji}^{(1)}, \beta_{ji}^{(2)}} \frac{1}{2}\|\tilde{\boldsymbol{y}}_i - \boldsymbol{x}_j\beta_{ji}^{(1)} - \boldsymbol{x}_{p+j}\beta_{ji}^{(2)}\|_2^2 \\ &+ \lambda_1(1 - W_{ji}\theta)(|\beta_{ji}^{(1)}| + |\beta_{ji}^{(2)}|) + \lambda_2(|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}|),\end{aligned} \tag{2.57}$$

in which $\tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \sum_{l \neq i,j} \boldsymbol{x}_l \beta_{li}^{(1),r} - \sum_{l \neq i,j} \boldsymbol{x}_{p+l}\beta_{li}^{(2),r}$.

When the algorithm converges in the last step, $\beta_{ji}^{(1)}$ and $\beta_{ji}^{(2)}$ will remain fixed. The connection will be a differential edge if $\beta_{ji}^{(1)} - \beta_{ji}^{(2)} \neq 0$.

Permutation test will be used to test the significance of $\beta_{ji}^{(1)} - \beta_{ji}^{(2)}$ not being 0. The null hypothesis is $\beta_{ji}^{(1)} - \beta_{ji}^{(2)} = 0$. In each permutation, we permute the elements in $\boldsymbol{x}_l$ and $\boldsymbol{x}_{p+l}$ together, and use the last step of block-coordinate-descent algorithm to calculate the permuted β-difference.

We use two-sided test to assign p-values. In the last step, denote $|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}| = \Delta$. By counting the number of $|\beta_{ji}^{(1)} - \beta_{ji}^{(2)}|$ larger than Δ divided by total number of permutations we get the p-value of the edge change.

By examining the coefficients of significant differential edges we are able to further delineate the type of edge changes as listed in Table 2.1.

To assess the effectiveness of permutation test, we apply permutation test to screen the claimed significant changes. Using multiple simulated datasets with 100 nodes and 200 samples we compare the precision and recall of differential edge detection before and after permutation screening.

Table 2.1: Types of significant differential edges.

$\beta_{ji}^{(1)}$	$\beta_{ji}^{(2)}$	Edge in condition 1	Edge in condition 2
> 0	$= 0$	Positive	NO
< 0	$= 0$	Negative	NO
$= 0$	> 0	NO	Positive
$= 0$	< 0	NO	Negative
> 0	< 0	Positive	Negative
< 0	> 0	Negative	Positive

We use p-value cutoff of 0.01, $i.e. \alpha = 0.01$, to compare the precision and recall before and after permutation test. The precision is the ratio between true positives and claimed positives and recall is the ratio between true positives and sum of true positives and false negatives. As the permutation test step aims to screen the false positives out from the claimed positives, the precision is expected to increase while recall can only decrease. Effective permutation screening is expected to increase precision while keeping recall close to the original result.

We summarize the increase of precision and decrease of recall in Figure 2.5. The first box is the increase of precision after permutation test step. In more than 75% simulations the increase is larger than 18%, and in more than half of the scenarios the increase is above 22%. Results show that the introduction of permutation filtered out many false positives and precision increased. The second box is the decrease of recall. In more than 75% of the realizations the decrease of recall is less than 10%. We would conclude that the permutation step gains big increase in precision while sacrificing small percentage of true positives.

The third box gives some insights into the reason of the effectiveness of permutation step. In order for the permutation screening take effect, it would be necessary that true positives rank high in terms of p-value in all claimed positives. The "p-value rank of ground truth"

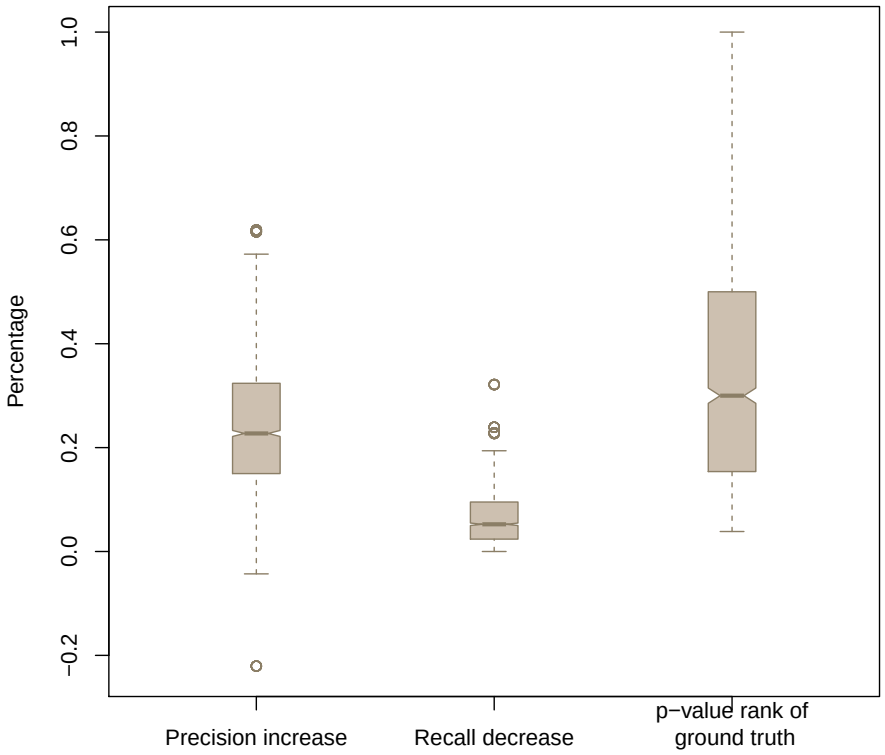


Figure 2.5: Increase of precision, decrease of recall, and the p-value ranking of ground truth in all claimed differential edges using permutation test as a second step accurate p-value assessment.

box shows the empirical ranking of ground truth in terms of p-value. The box shows that at least 75% of ground truth rank in the first half of all claimed positives. Because the permutation p-value rank the true positives higher, the use of it as screening is able to increase precision while maintaining recall. The permutation test also assigns accurate p-values to all differential edges.

2.4 Experimental Results

We test the proposed approach on multiple simulated data sets. The experiments on simulated data demonstrate the effectiveness of the proposed method and corroborate the theoretical analysis of our methods.

2.4.1 A Toy Model

In the first simulation experiment, we create a toy model, a Gaussian Markov random field consisting of 12 nodes under two conditions. The composite network of the two condition-specific networks is shown in Figure 2.6(a). The black edges indicate the connections that exist under both conditions; the red edges indicate the connections that only exist under condition 1; and the green edges indicate the connections that only exist under condition 2. Under each condition, 100 samples are simulated from a multivariate Gaussian distribution.

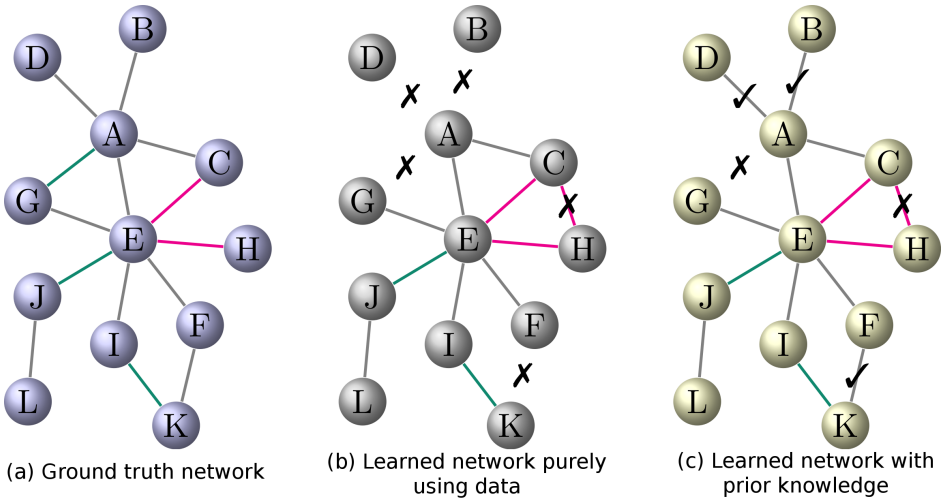


Figure 2.6: Network inference results in the toy model. (a) The ground-truth network topology under two conditions; (b) Network inference result purely based on data; (c) Network inference result with prior knowledge.

Figure 2.6(b) is the network inference result without prior knowledge. This purely data-driven approach is the same as the method proposed in (Zhang and Wang, 2010). As illustrated in Figure 2.6(b), the inference algorithm without prior knowledge missed four edges under both conditions (black edges) and falsely detected one edge under condition 2 (green edge), as marked by x marks.

For the knowledge-fused network learning approach, we use the nine black edges as prior knowledge. Therefore, the number of edges in prior knowledge M is 9, and let $\delta = 0.1$. As shown in Figure 2.6(c), the newly proposed method with prior knowledge significantly improved the network inference results: the number of falsely missed edges decreased to 1; and one edge under condition 2 was falsely detected.

2.4.2 Performance Evaluation Using Simulation Data

In the simulation studies, we used Gaussian Markov random field to generate the simulation data following four steps. Firstly, generate two adjacency matrices with sparse changes. Secondly, create the precision matrix with same structures with the adjacency matrices. Thirdly, get valid covariance matrices by inverting the precision matrices. Lastly, simulate data according to the covariance matrices.

A network structure can be represented by an adjacency matrix, where non-zeros indicate dependencies between nodes. We used Gaussian Markov random field to generate the simulation data. Under such model, nodes follow multivariate Gaussian distribution and their dependencies are reflected by the non-zero elements in precision matrix, which can be equivalently treated as the adjacency matrix of a network. So we generate an $N \times N$ precision matrix, then the network structure and simulation data can be derived from the precision matrix. In the generation of precision matrix, we first initialize an $N \times N$ empty matrix, and then every node is randomly connected to d neighbors. d is the degree of connection

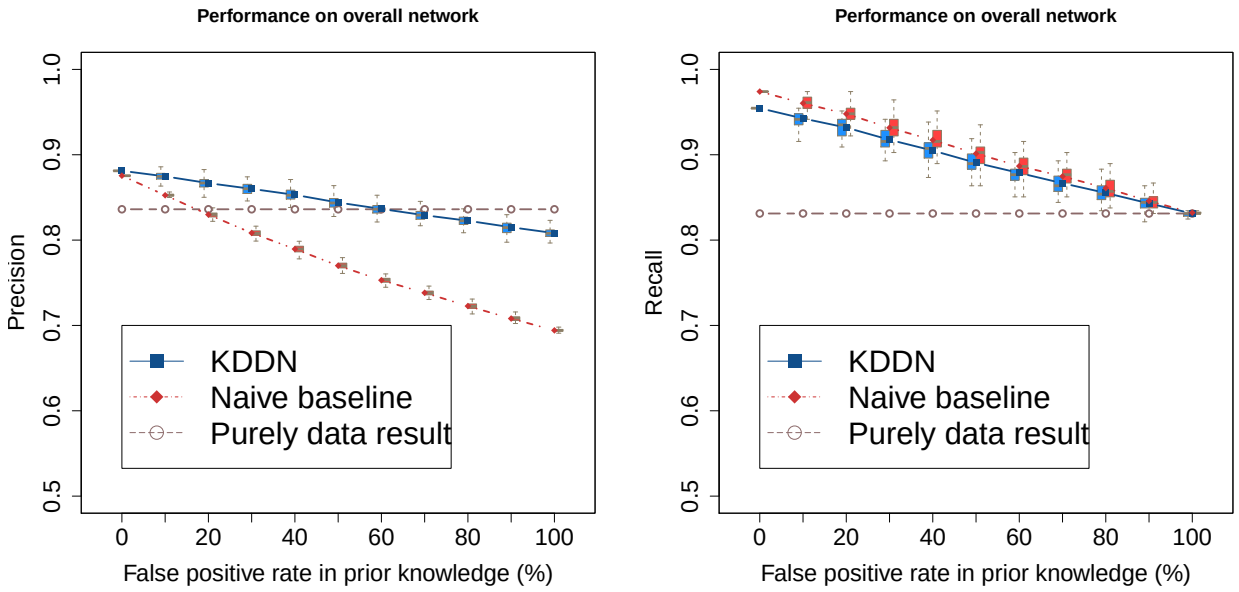
uniformly chosen between 1 and 4 to get a sparse structure. Nodes with more than d connections are performed with random connection removal and nodes with no connections are performed with random connection addition until all nodes have connections but no more than 4. In order to make the precision matrix invertible and invert to a valid positive semi-definite covariance matrix, we randomly assign and adjust the values in precision matrix in the range [0.2, 0.3], while keeping the sum of each row less than 1 (Meinshausen and Bühlmann, 2006). The simulation data is finally generated according to covariance matrix using R package mvtnorm (Genz *et al.*, 2012; Genz and Bretz, 2009).

Performance of Network Construction and Knowledge Incorporation

To assess the effectiveness of prior knowledge incorporation and robustness of KDDN when false positive edges were present in prior knowledge, we examined the network inference precision and recall of the overall network and the differential network at different levels of false positive rate in the prior knowledge. The number of edges in prior knowledge M was set to be the number of common edges in the two conditionspecific networks, and δ was set to 0.1.

It is important to note that both false positives and false negatives in the prior knowledge here are with respect to the condition-specific ground truth from which the data are generated. Thus, false positives in prior knowledge may contribute more learning errors, false negatives will not worsen network learning performance (results shown in Section 2.4.2).

Starting from prior knowledge without any false positive edges, we gradually increased the false positive rate in prior knowledge until all prior knowledge was false. At each given false positive rate in the prior knowledge, we randomly created 1,000 sets of prior knowledge, and compared the performance of KDDN in terms of precision and recall with two baselines: (1) a purely data-driven result without incorporating knowledge; and (2) a naive baseline of



(a) (b)

Figure 2.7: The effects of false positive rate in the prior knowledge on inference precision and recall of overall network.

knowledge incorporation by directly superimposing the prior knowledge network upon the purely data result. The results are shown in Figure 2.7.

The dot-connected lines are averaged precision or recall and the box plot shows the first, second and third quartiles of precision or recall at each false positive rate in prior knowledge (with the ends of the whiskers extending to the lowest datum within 1.5 interquartile range of the lower quartile, and the highest datum within 1.5 interquartile range of the upper quartile).

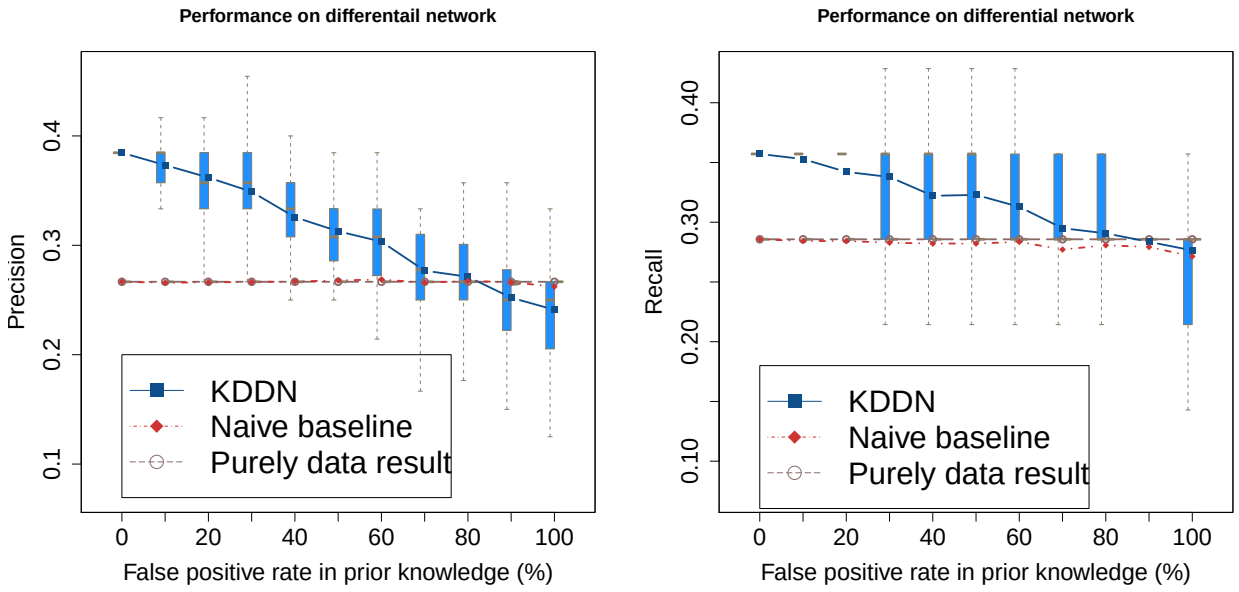
Precision reflects the robustness to the false positive edges and efficiency of utilizing the information in prior knowledge. Figure 2.7(a) shows that, as expected, the false positive rate in prior knowledge has limited effect on the precision of KDDN (blue squared lines). With more false positives in the prior knowledge, the precision decreases but is still around

the purely data baseline (brown circle lines) and much better than the naive baseline (red diamond lines). While the naive baseline suffers significantly from the false positives in prior knowledge as it indiscriminately accepts all edges in prior knowledge without considering evidence in the data. This observation corroborates the design of our method: to control the false detection incurred by the false positives in the prior knowledge. At the point where the false positive rate in the prior knowledge is 100%, the decrease of precision compared with the purely data based result is bounded within δ.

Recall reflects the ability of prior knowledge in helping recover missing edges. Figure 2.7(b) shows that when the prior knowledge is 100% false, the recall is the same as that of the purely data based result, because in this case the prior knowledge brings in no useful information for correct edge detection. When the true positive edges are included in the prior knowledge, the recall becomes higher than that of the purely data based result as more edges are correctly detected by harnessing the correct information in the prior knowledge. The naive baseline is slightly higher in recall since it calls an edge as long as knowledge contains it, while KDDN calls an edge only when both knowledge and data evidence are present. The closeness between KDDN and naive baseline demonstrates the high efficiency of our method in utilizing the true information in prior knowledge.

We then evaluated the effect of knowledge incorporation solely on the identification of differential network following the same protocol. The results are shown in Figure 2.8.

For differential network recovery, the naive baseline is almost identical to purely data results because the prior knowledge seldom includes a differential edge in a large network with sparse changes. While similar advantages of KDDN apply, our method has better precision and recall, and bounds the performance degradation when knowledge is totally wrong. Unlike the naive baseline where knowledge and data are not linked, we model the inference with knowledge and data together, so knowledge is also able to help identify differential edges.



(a) (b)

Figure 2.8: The effects of false positive rate in the prior knowledge on inference precision and recall of differential network.

Performance in Noise Cases

We demonstrate the performance of the methods with noise corrupted simulation data. We fix the sample size at 100, and generated simulation data for $p = 50, 100, 200$ added Gaussian white noise with signal to noise ratio $SNR = 0, 1, 2, 3, 4, 5$. For each case we compare the performance of the method under: purely data without noise, purely data with noise, and gradually increase false positives in knowledge with and without noise.

The results of data sets with $p = 50$ are in Figure 2.9.

From the results we still see the effectiveness of knowledge incorporation in all noise level. On one hand, the performance degrades as a result of noise corruption compared with noise free. But on the other hand we are happy to see that in all noise level knowledge incorporation largely improved the performance and the adverse effects of all wrong malicious knowledge.

The experiment results with $p = 100$ and $p = 200$ are in Figures 2.10 and 2.11, from which we can see the similar trends and conclusions.

Effects of Nonuniform Random Knowledge

In practice, FPs in prior knowledge may be more likely to bias towards or against certain nodes rather than distributed uniformly. Proteins/genes with important functions tend to be studied more intensively and therefore accumulated more knowledge. Under specific conditions, knowledge associated with those proteins/genes are more likely to include FPs. In such cases, FPs concentrate more on some nodes than others, which actually makes KDDN avoid FPs more efficiently due to sparse selection mechanism. We compared the performance with random knowledge and biased knowledge using simulation data. Instead of adding false positives uniformly, we add false positives to nodes as follows to biased towards top nodes: the first node contains 1/3 of all FPs, the second node contains 1/3 of the remaining FPs

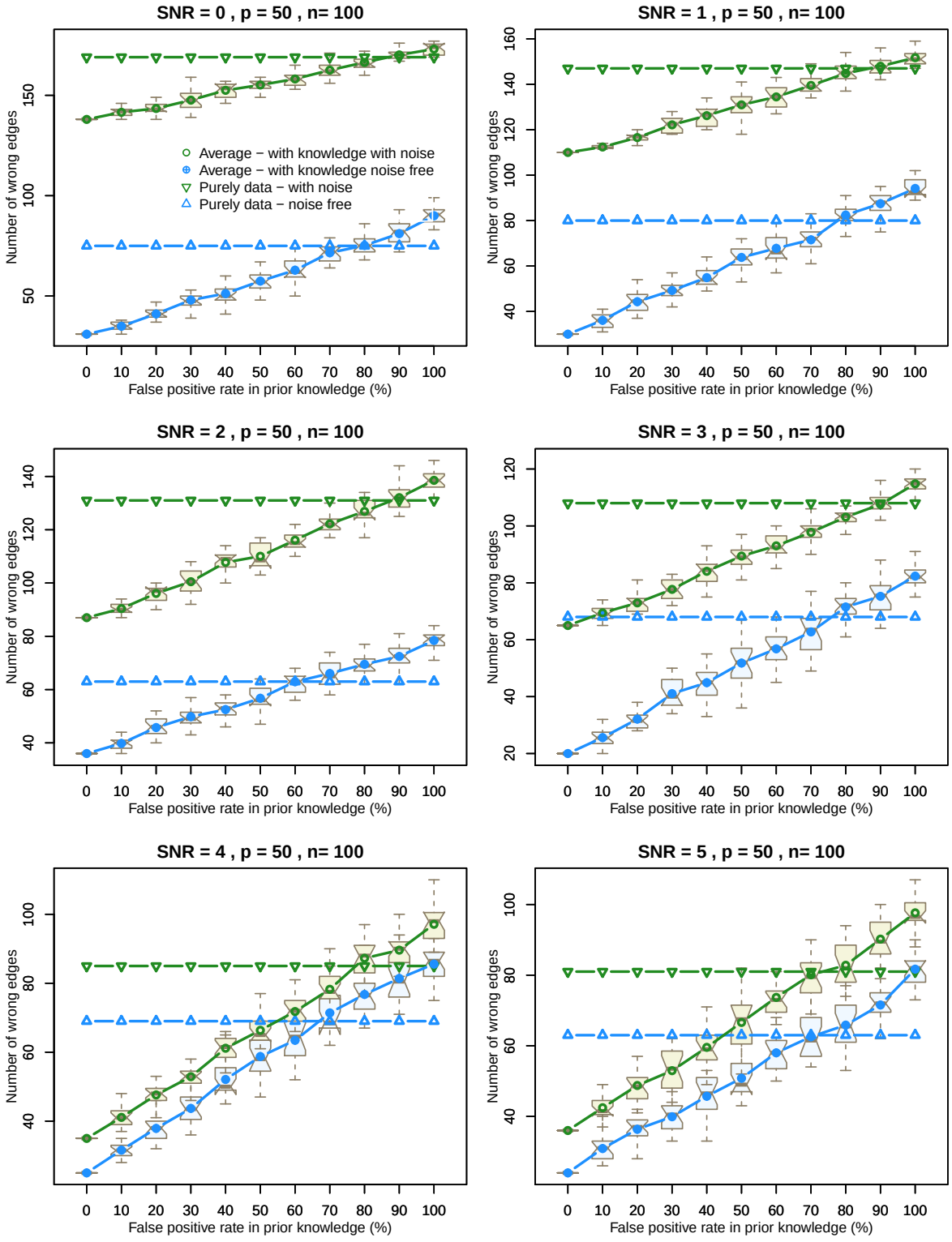


Figure 2.9: Performance in noise cases, p=50.

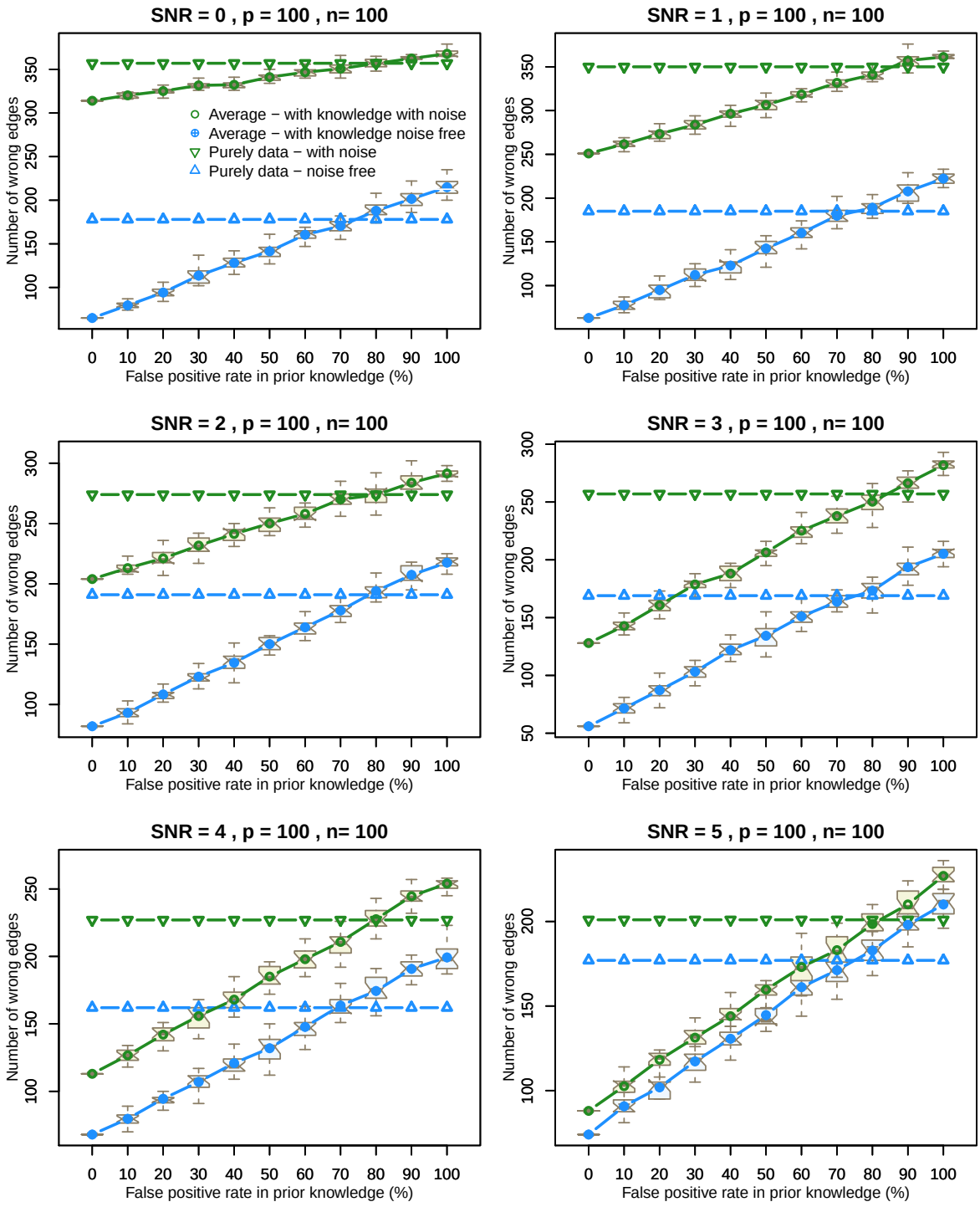


Figure 2.10: Performance in noise cases, p=100.

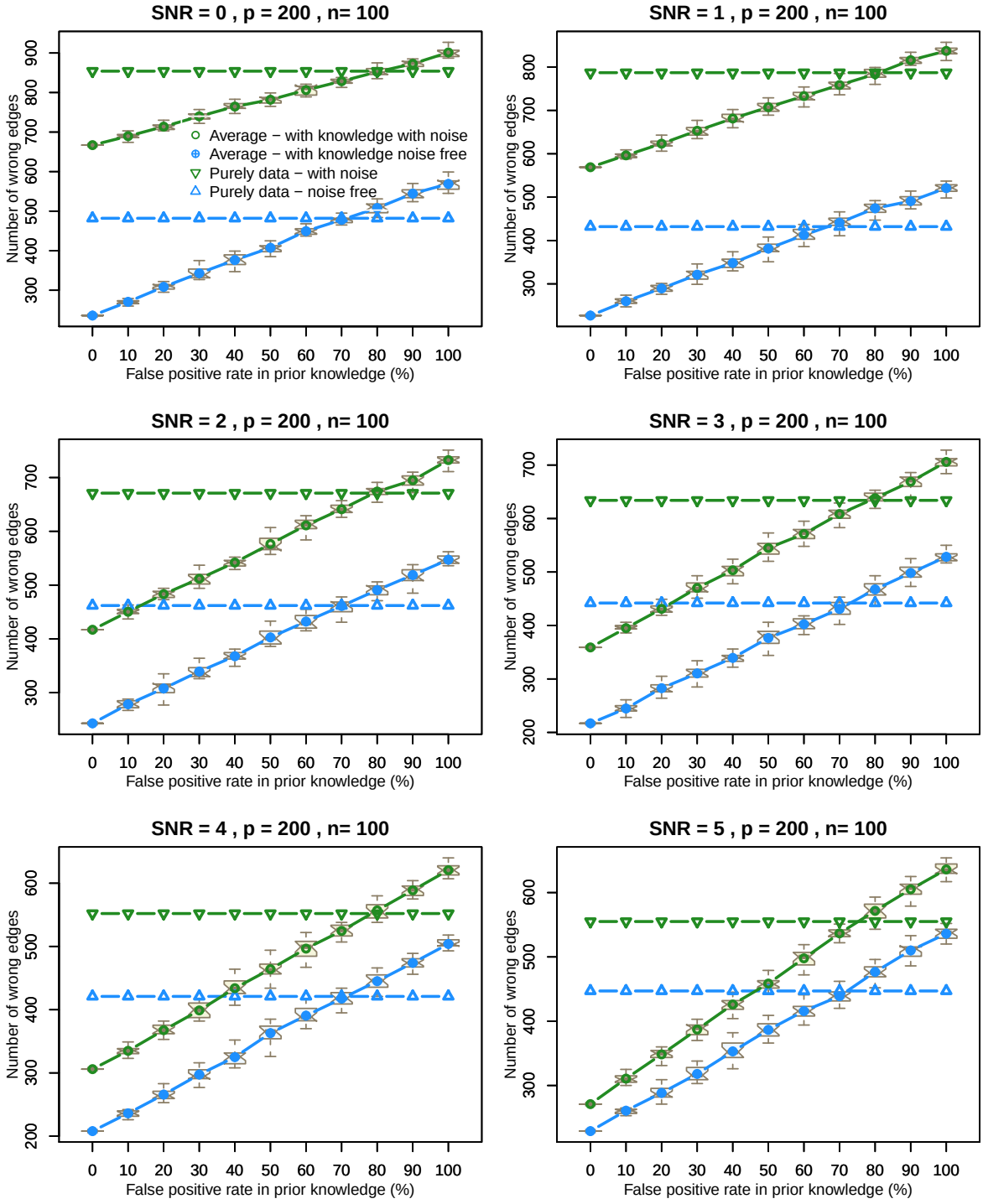
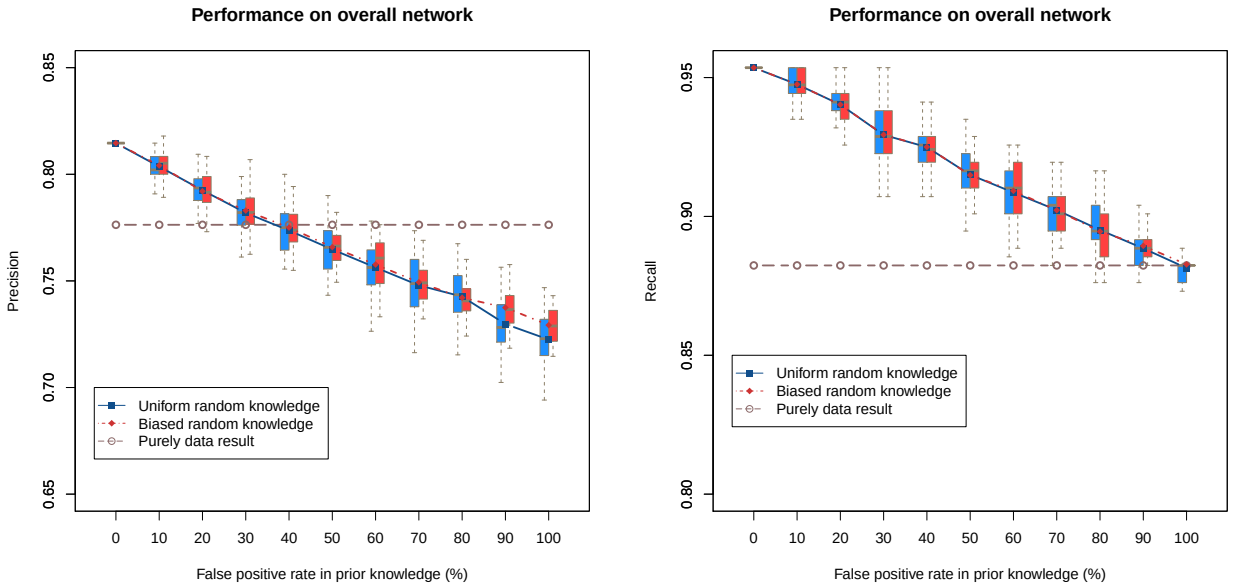


Figure 2.11: Performance in noise cases, p=200.



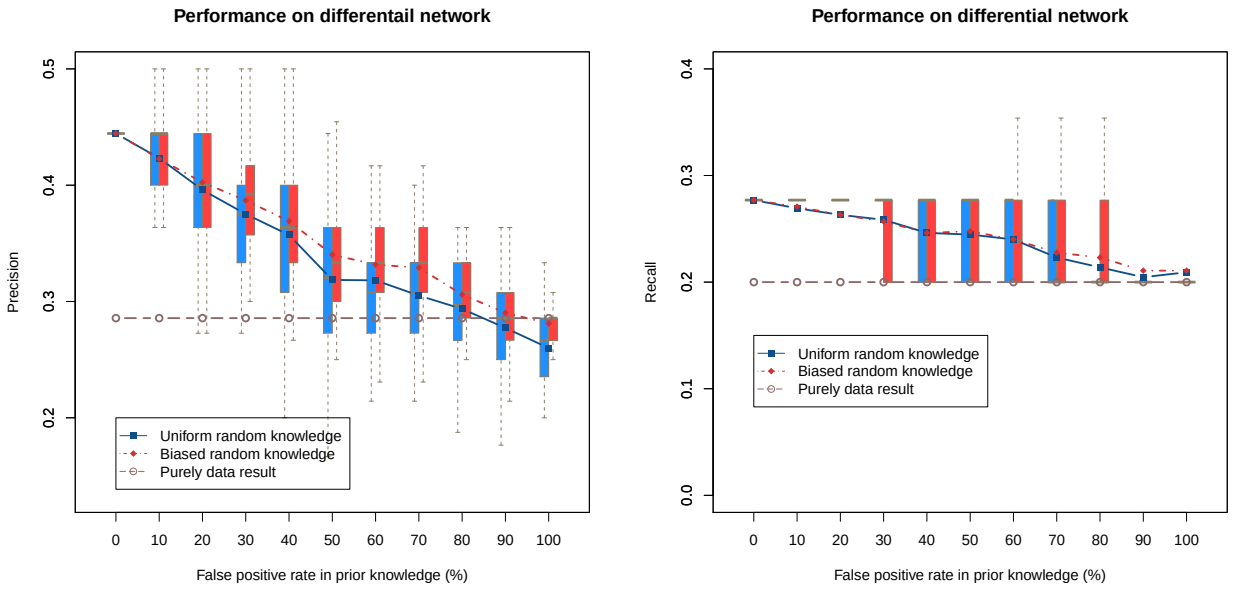
(a) (b)

Figure 2.12: The effects of nonuniform random prior knowledge on inference precision and recall of overall network.

which is 2/9, and the third node contains 1/3 of the remaining FPs, etc. Since in practice the knowledge and data are independent and the data generation process is equal for all nodes, the order of nodes does not matter in this biased knowledge assignment. The results are shown in Figures 2.12 and 2.13 for 100 node example. This change of random knowledge generation either made no difference or slightly improved the performance, confirming that the uniformly random is the worst case and KDDN bounds the performance under worst case scenario.

Effects of False Negatives in Prior Knowledge

Existing biological knowledge databases are mainly manually curated, which may suffer more from false negatives than false positives. However, the knowledge in databases are aggregated



(a) (b)

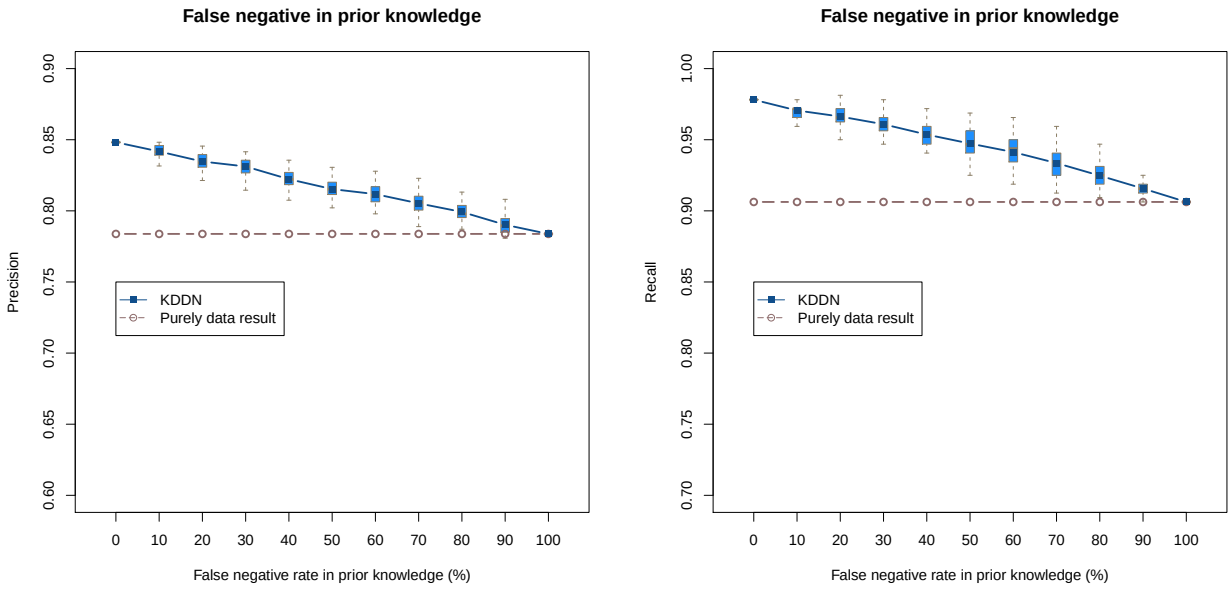
Figure 2.13: The effects of nonuniform random prior knowledge on inference precision and recall of differential network.

from general conditions. When they are used to guide inference of network under specific biological conditions, the inconsistency between knowledge and ground-truth become false positives. Unlike the assessment of databases quality, in knowledge incorporated inference, false positives are the major concerns as they directly affect the inference results, while false negatives which are the ground-truth not reflected by the knowledge do not affect the inference results.

To show this experimentally, we simulated the scenarios with different amount of false negatives in prior knowledge with fixed size of prior knowledge without the presence of false positives. When false positives present, increasing false negatives is equivalent to increasing false positives as we did in Figures 3 and 4, given fixed size of prior knowledge. Starting from all true knowledge, we gradually decrease the size of prior knowledge which is equivalent to adding in false negatives, until the size of prior knowledge is 0 and the inference purely relies on data. The results are shown in Figure 2.14. From the results of experiments with various false positive rate and false negative rate in prior knowledge, we can conclude that true positives in knowledge benefit the inference, false positives degrade the performance but our method controls it, and false negatives do not affect the inference.

Empirical Type I Error Rate for Simulated Data Sets Under the Null Hypothesis

To the best of our knowledge, our method is the first to assess the significance of differential edges. To assess if our method identifies the differential edges at the right significance level, we test the type I error rate of differential edge detection using multiple simulation datasets under the null distribution (no differential edges between the two networks). If the type I error rate is either too conservative or too liberal, the p-value fails to reflect the actual false positive rate and we cannot control how many false positives are detected by setting a p-value based threshold (Chen *et al.*, 2011). Null data sets based on multivariate Gaussian distribution are simulated and the false positive rate when $\alpha = 0.05$ are calculated with



(a) (b)

Figure 2.14: The effects of false negatives in prior knowledge on inference precision and recall.

networks of size 80, 100, 120 and 150 in number of nodes. Experiments show the average type I error rate under null converge exactly to α and the standard deviation decreases with larger network size, see Figure 2.15. This accuracy in p-value estimation gives stronger confidence in differential edge detection.

2.4.3 Performance Comparison

We compared our joint learning method KDDN with four peer methods: 1) DDN (independent learning) (Zhang *et al.*, 2009), 2) csLearner (joint learning) (Roy *et al.*, 2011), 3) Meinshausen's method (independent learning) (Meinshausen and Bühlmann, 2006), and 4) Tesla (joint learning) (Ahmed and Xing, 2009). csLearner can learn more than two networks but we restricted the condition to two. Meinshausen's method learns the network under single condition, and we combined the results learned under each condition to get conserved network and differential network. Tesla learns a time-evolving network, but can be adapted to two-condition learning as well. Only KDDN can assign edge-specific p-values to differential edges.

Parameters in KDDN are automatically inferred from data as described in Section 2.3.3. For the competing methods in the comparison, we manually tested and tuned their parameters to obtain their best performance. We set DDN to detect pairwise dependencies. The number of neighbors in csLearner is set to "4" (the ground truth value). Meinshausen's method uses the same λ_1 as inferred by KDDN as it is a special case of KDDN under one condition without prior knowledge. Tesla uses the empirically-determined optimal parameter values, since the parameter selection was not automatic but relies on user input.

To assess the impact of prior knowledge, we run KDDN under three scenarios: data-only (KDDN.dt), data plus true prior knowledge (KDDN.tk), and data plus "random" prior knowledge (KDDN.fk). Only KDDN is able to utilize prior knowledge.

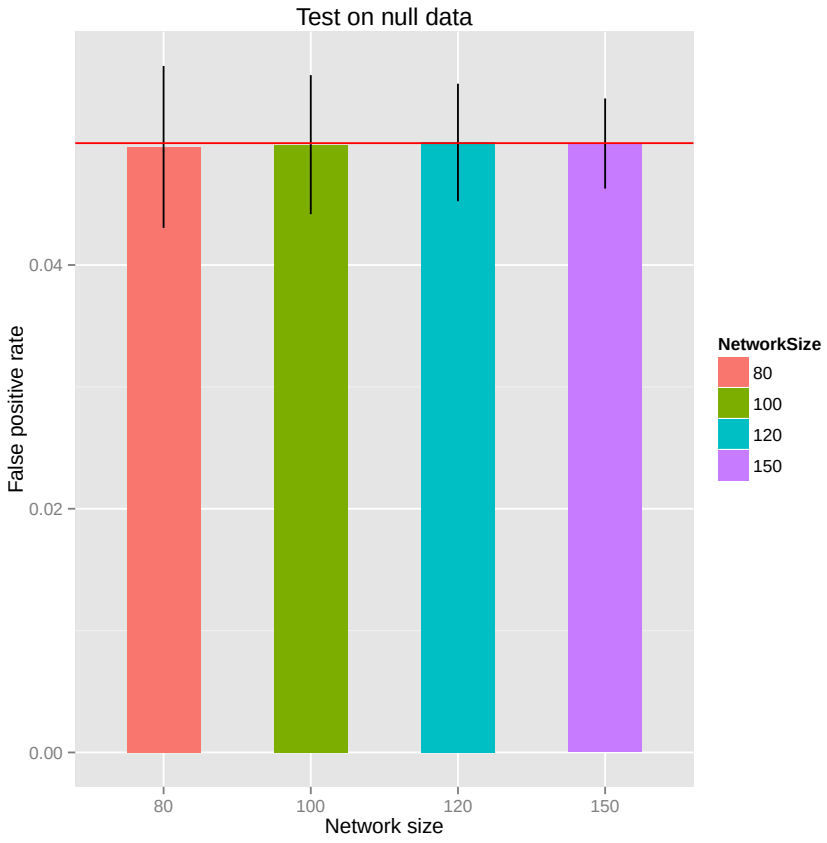


Figure 2.15: Empirical type I error rate (false positive rate) for simulated data sets under the null hypothesis under four different network sizes. The designed significance level $\alpha = 0.05$ is indicated by the red line. Across multiple runs, the average type I error rate is close to , which shows the differential edge detection is neither too conservative nor too liberal.

The ground truth networks consist of 80, 100, 120, 140 and 160 nodes, respectively, and correspondingly 120, 150, 200, 200 and 240 samples. For each network size, 100 simulation datasets are generated. We evaluate the performance of inferring both overall and differential edges of the underlying networks using the F-score (harmonic mean of precision and recall, $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$) and precision-recall averaged over all datasets under each network size.

Figure 2.16(a) compares the ability of recovering overall networks. We see KDDN.tk consistently outperforms all peer methods, and KDDN.dt and KDDN.fk performs comparatively to Tesla (best-performing peer method). Independent learning methods, DDN and Meinshausen's method, place third due to their inability to jointly use data. KDDN.tk increased the average F-score of best performing peer method (Tesla) by 25%, 20%, 7%, 13% and 23% in networks with size 80, 100, 120, 140 and 160 respectively.

Figure 2.16(b) shows the comparison of performance on recovering differential edges. Results show that KDDN consistently outperforms all peer methods under all scenarios. The fact that KDDN determines λ_2 according to the statistical significance of differential edges helps KDDN outperforms Tesla in differential edge detection. It is also clear that a single-condition method cannot find the differential edges correctly and has the worst performance. KDDN.tk increased the average F-score of best performing peer method (Tesla) by 55%, 66%, 102%, 74% and 106% in networks with size 80, 100, 120, 140 and 160 respectively.

To assess the statistical significance of whether KDDN performs better than the peer methods, one-sided t-test is used to compare the results across all network sizes. The null hypothesis is that there is no difference between the mean of F-score of KDDN and the peer methods. The alternative hypothesis is that KDDN has greater mean of F-score. The detailed results are included in Tables 2.2 and 2.3, which show that KDDN.tk performs significantly better than peers in all cases, and KDDN.dt and KDDN.fk performs better than peers in 118 out of all 120 cases. Better results at 0.05 significance level are highlighted by bold font.

Table 2.2: T-test result of KDDN with peers in overall network learning performance.

Nodes in network	KDDN	csLearner	Meinshausen	DDN	Tesla
80	KDDN.dt	**4.19e-43**	**1.52e-56**	**6.26e-33**	**8.67e-08**
	KDDN.tk	**6.64e-46**	**3.86e-69**	**9.02e-45**	**2.08e-44**
	KDDN.fk	**2.06e-42**	**2.40e-52**	**7.58e-30**	**0.0039**
100	KDDN.dt	**1.81e-37**	**4.72e-53**	**2.11e-45**	**0.0499**
	KDDN.tk	**1.53e-41**	**2.79e-62**	**2.07e-50**	**4.21e-42**
	KDDN.fk	**6.40e-37**	**1.04e-45**	**7.19e-43**	0.9778
120	KDDN.dt	**2.20e-35**	**1.11e-99**	**3.99e-46**	**3.64e-06**
	KDDN.tk	**9.66e-37**	**4.86e-94**	**1.53e-47**	**8.74e-28**
	KDDN.fk	**5.49e-34**	**3.60e-94**	**1.19e-42**	0.99
140	KDDN.dt	**7.23e-34**	**1.63e-104**	**1.15e-47**	**1.35e-08**
	KDDN.tk	**3.45e-35**	**1.08e-101**	**6.83e-49**	**4.25e-15**
	KDDN.fk	**1.91e-32**	**2.37e-99**	**1.77e-44**	**0.0057**
160	KDDN.dt	**3.34e-46**	**1.75e-90**	**2.69e-51**	**2.56e-44**
	KDDN.tk	**4.31e-47**	**8.08e-83**	**5.32e-52**	**2.17e-47**
	KDDN.fk	**3.76e-45**	**4.55e-90**	**7.65e-50**	**4.22e-37**

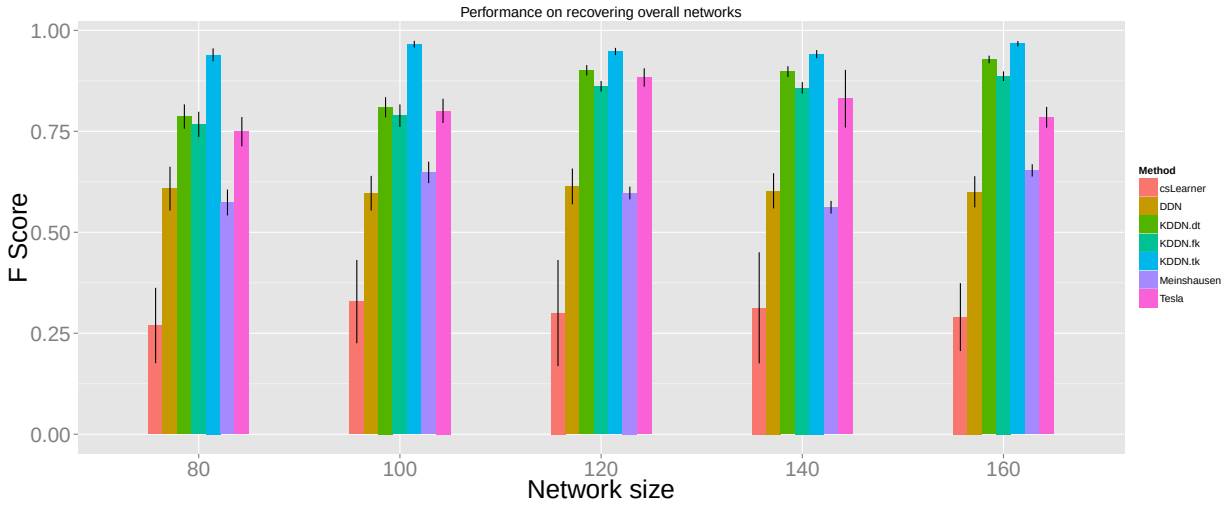
Table 2.3: T-test result of KDDN with peers in differential network learning performance.

Nodes in network	KDDN	csLearner	Meinshausen	DDN	Tesla
80	KDDN.dt	**4.77e-12**	**3.45e-13**	0.2005	**0.0047**
	KDDN.tk	**5.62e-16**	**1.34e-16**	**0.0075**	**1.72e-05**
	KDDN.fk	**3.83e-12**	**2.20e-13**	0.2669	**0.0075**
100	KDDN.dt	**8.65e-16**	**3.41e-15**	**2.76e-05**	**0.0005**
	KDDN.tk	**9.65e-18**	**8.55e-17**	**4.73e-08**	**1.90e-06**
	KDDN.fk	**2.27e-14**	**6.65e-14**	**0.0001**	**0.0015**
120	KDDN.dt	**4.89e-40**	**3.95e-34**	**3.73e-25**	**2.08e-13**
	KDDN.tk	**1.95e-41**	**1.31e-34**	**1.17e-28**	**9.33e-17**
	KDDN.fk	**2.62e-39**	**2.46e-34**	**1.39e-22**	**4.39e-11**
140	KDDN.dt	**4.53e-35**	**9.76e-33**	**1.10e-24**	**1.01e-09**
	KDDN.tk	**4.47e-36**	**4.67e-33**	**4.15e-27**	**4.41e-12**
	KDDN.fk	**1.38e-34**	**3.15e-33**	**5.88e-22**	**3.24e-07**
160	KDDN.dt	**4.43e-41**	**4.20e-34**	**1.45e-32**	**2.05e-19**
	KDDN.tk	**4.32e-46**	**4.28e-37**	**4.21e-38**	**2.64e-25**
	KDDN.fk	**4.20e-40**	**1.14e-33**	**3.09e-29**	**4.53e-15**

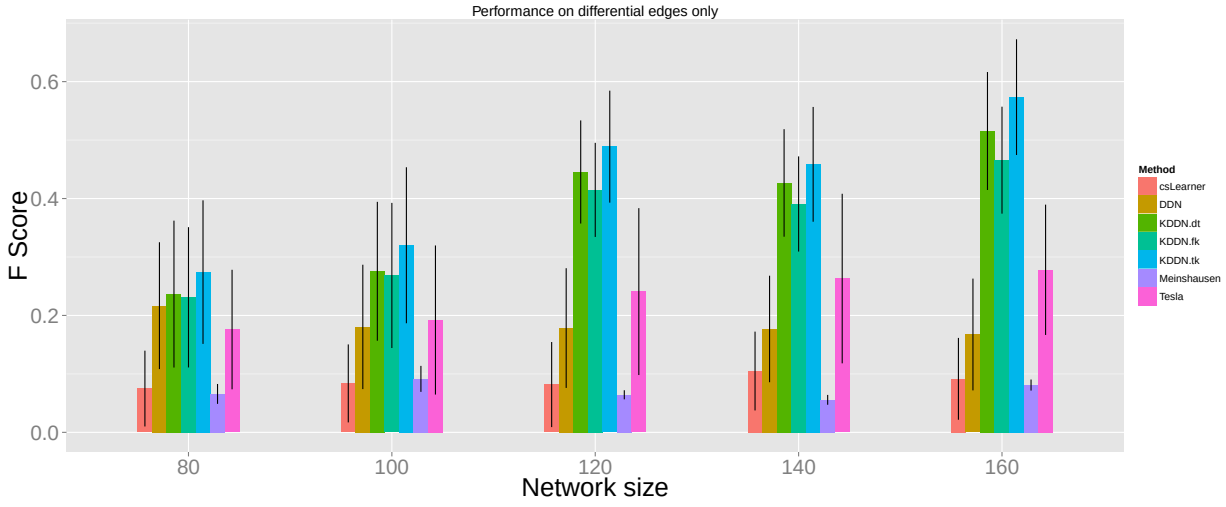
Through these comparisons, we show that KDDN performs better than peer methods in both overall and differential network learning. High-quality knowledge further improves KDDN performance, while KDDN is robust enough to even totally random prior knowledge. Joint learning, utilization of prior knowledge and attention on statistical significance help KDDN outperform.

We plot the performance comparison results in Figure 2.16 on the plane of precision and recall, with background heatmap indicating F score ranging from 0 to 1, in Figure 2.17. Besides the same observation of F score performance in the paper, we also see that KDDN performs best in both precision and recall. Tesla performs the second. Meinshausen's method performs third in overall network but poor in differential network.

In addition to the performance comparison on networks with different sizes, we also compared KDDN with DDN on the simulation example used in (Zhang *et al.*, 2009). The example was generated by SynTren and the comparison is shown in Figure 2.18. The nodes are placed at the same relative positions. The ground truth network is shown in Figure 2.18(b), with 20 nodes. The black edges indicate common edges. Red and green edges are condition-specific edges. The result learned by DDN is shown in Figure 2.18(c), in which only differential edges are learned, and 3 edges are error. The network learned by KDDN is shown in Figure 2.18(a). The number of erroneous edges is still 3 but 5 common edges are also identified. In this example the two methods achieved comparable performance in differential edge detection due to small network size, but the common edge identification can only be done by the proposed method.

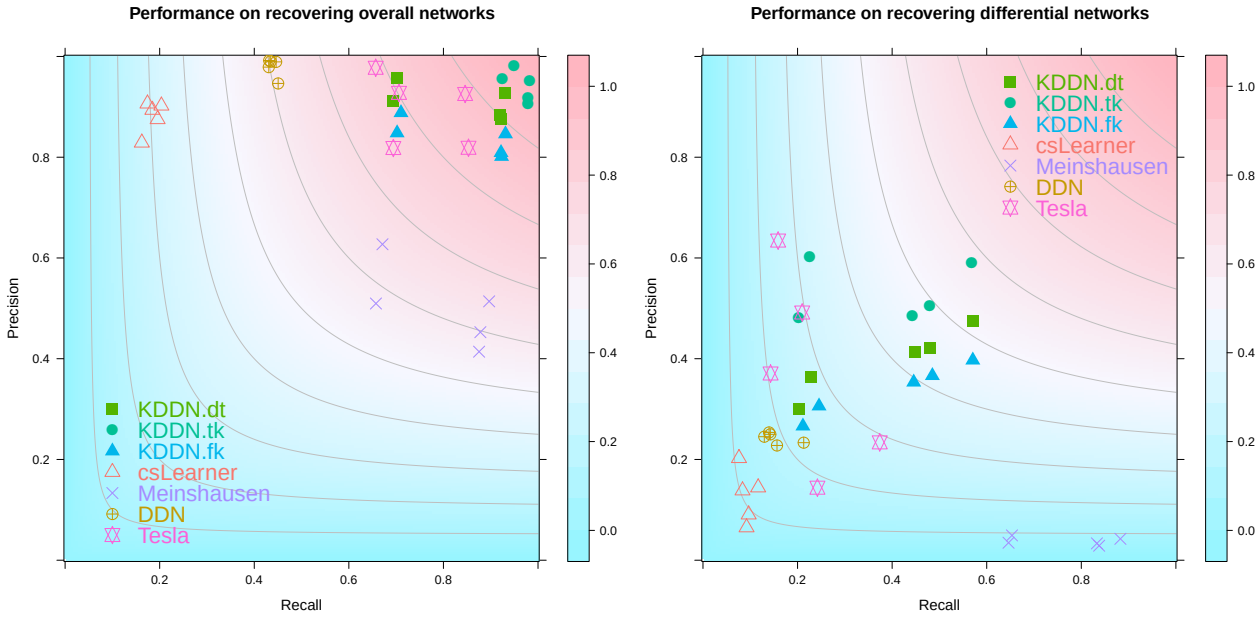


(a)



(b)

Figure 2.16: Performance comparison in F score. (a) Recovery of overall network. (b) Recovery of differential network.



(a) (b)

Figure 2.17: Performance of network recovery displayed on the plane of precision and recall with F score heatmap as background.

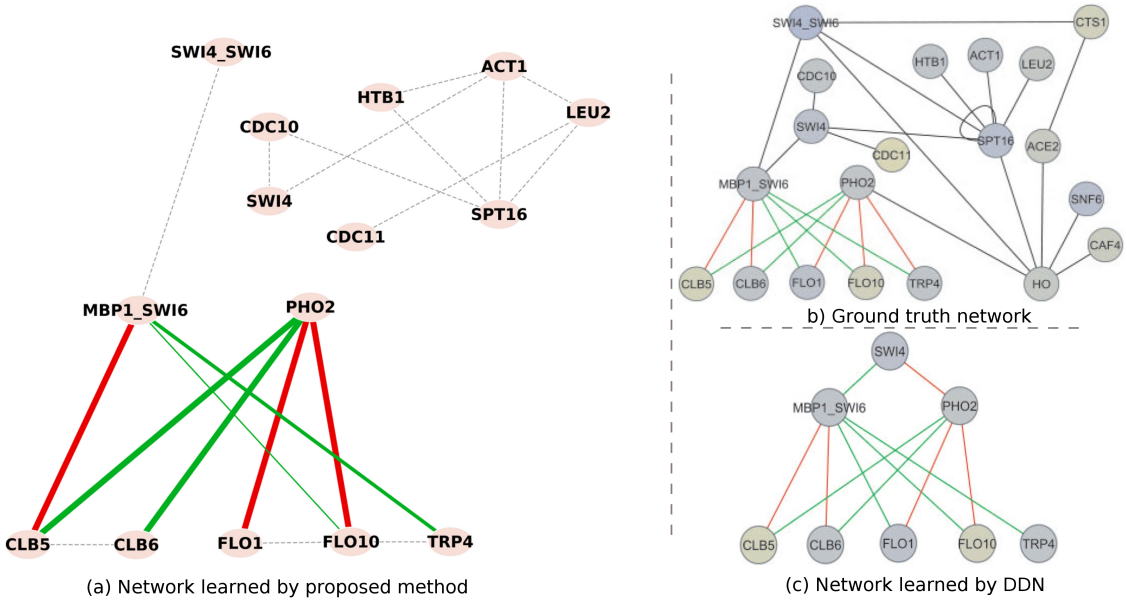


Figure 2.18: Comparison of results on simulation data generated by SynTren. (a) The result learned by the proposed method. (b) Ground truth network. (c) Network learned by DDN method.

2.5 KDDN Software

To facilitate the usage and adoption of KDDN in the research community, we develop a Cytoscape app to fully implement KDDN algorithm in a user-friendly way. KDDN algorithm is very efficient and provides the users with parallel computing capability utilizing ubiquitous multi-core machines. While the principal application here involves microarray gene expressions, our methodology and software can be readily applied to other types of quantitative molecular profiling data. Source code and compiled package of KDDN are freely available for download at `http://apps.cytoscape.org/apps/kddn`.

2.5.1 Introduction

KDDN learns the network structure as well as its significant changes using biological data and existing knowledge. It maximizes the synergy among experimental data, existing domain knowledge and rigorousness of mathematical modeling. KDDN addresses several unsolved issues concerning differential network inference using data-knowledge integrated approaches: (1) the solution (search) space is usually large while sample sizes are small, resulting in potential overfitting; (2) both conserved and differential biological networks are complex while currently lacking closed-form or efficient numerical solutions; (3) "structural" model parameters are assigned heuristically, leading to potentially suboptimal solutions; (4) prior knowledge is imperfect for inferring biological networks under specific conditions, *e.g.*, false positive "connections", biases, and non-specificity; and (5) current differential network learning (Mitra *et al.*, 2013) methods including our pioneer tool DDN (Zhang *et al.*, 2011) do not provide significance assessment on the differential connections and rigorous testing of the type I error rate.

KDDN is implemented as an open source Cytoscape 3.x app to facilitate the adoption in research community. In addition to the methodological advantages, KDDN software enjoys several practically valuable features: (1) it wraps up the complex methodology with user friendly interface requiring just a few clicks from installation to carrying out experiments; (2) it fully utilizes the Cytoscape (Cline *et al.*, 2007) framework by interacting with users and presenting information in various specialized windows and panels; (3) it adopts concurrent computing technology to benefit from the commonly found multicore/multiprocessor hardware, further speed up the notoriously complex network learning process.

KDDN has been tested thoroughly using both synthetic and real data. Comparison with peer methods demonstrated the advantage of KDDN joint learning framework and the prowess of knowledge incorporation ability. KDDN is successfully applied to a wide range of re-

search projects and case studies on yeast oxidative stress response, breast cancer recurrence, muscular dystrophy and estrogen receptors are presented in Chapter 3.

2.5.2 Implementation

KDDN algorithm jointly learns both the conserved network structure and statistically significant rewiring across different conditions. Condition-specific data and prior domain knowledge are systematically fused by a Lasso-type model with ℓ_1 regularized convex optimization formulation. By exploring the special characteristics of the problem, an efficient closed-form solution is derived for the embedded sub-problem of block-wise coordinate descent algorithm, essentially facilitates the computation and make the algorithm scalable to networks with thousands of nodes. To deal with the common concern with prior knowledge incorporation approaches that prior knowledge may be unspecific and conflict with the ground-truth, KDDN uses a novel strategy to estimate and control the adverse effect caused by the worst case scenario and incorporate maximally within that range. The strategy has been proven theoretically and experimentally to efficiently leverage the information in domain knowledge while remaining robust if the prior knowledge is inconsistent with the underlying ground-truth. KDDN also pays special attention to the statistical significance of the identified network by rigorously calibrating model parameters that quantitatively correspond to the expected significance level and assess edge-specific p-value of differential edges.

KDDN is developed as a Cytoscape 3.x app with parallel computing capability in Java and runs independently of platform. Conveniently distributed through Cytoscape App Store (Lotia *et al.*, 2013), KDDN is easily accessible through a few clicks. The comprehensive analysis of both conserved network structure and differential changes are aided by straightforward configurations. Networks are visualized seamlessly in Cytoscape and detailed numerical results are presented in the table panel and a dedicated results panel.

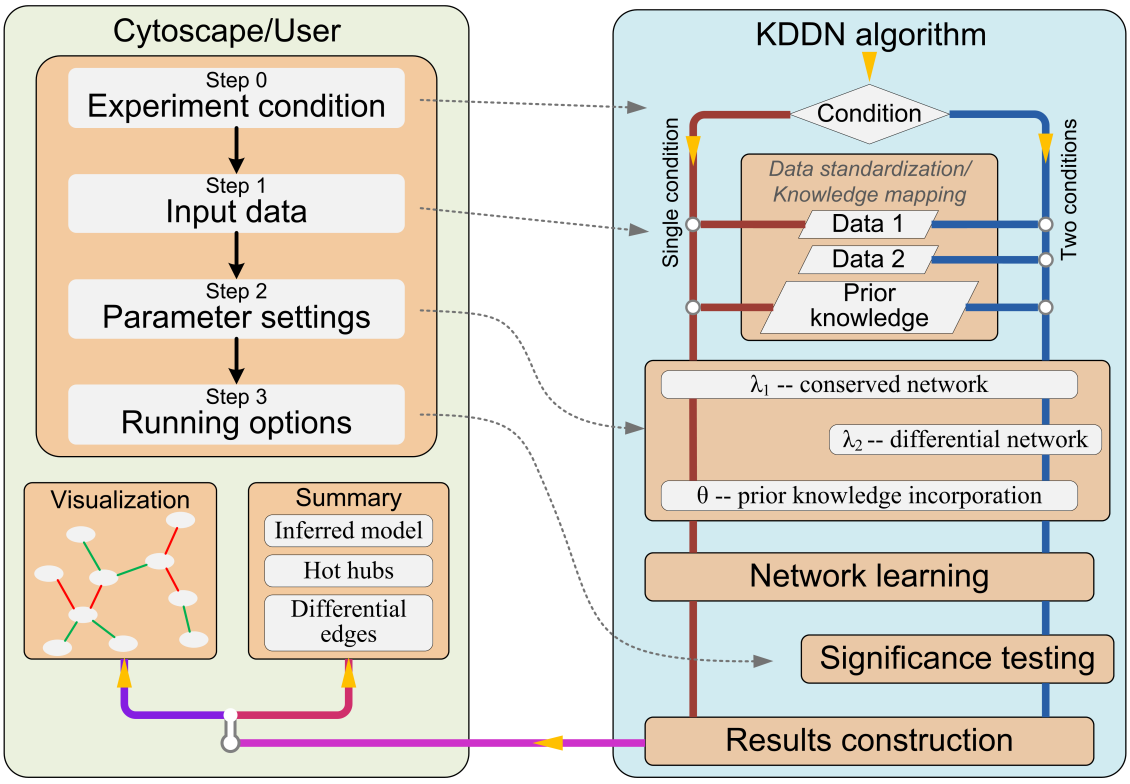


Figure 2.19: Architecture and flowchart of KDDN method and app.

Taking a list of genes or protein products as network nodes, and their measured data such as gene expression or protein expression as input, KDDN infers the static network structure if only single condition data is available, or detects the network rewiring and its statistical significance if data under two conditions are provided. Prior knowledge of the network can be supplied to guide the learning, otherwise the learning relies solely on data. KGML files downloaded from KEGG pathway website can be directly imported as prior knowledge. The architecture and design of KDDN app is illustrated in Figure 2.19 and Figure 2.20.

KDDN presents an integrated and systematic tool to explore the biological network structure and rewiring. Supported by well-grounded statistical framework, KDDN combines the strengths of versatile existing knowledge of network structure and abundant experiment data to depict an overall picture of biological networks and their dynamics. Tested on both simula-

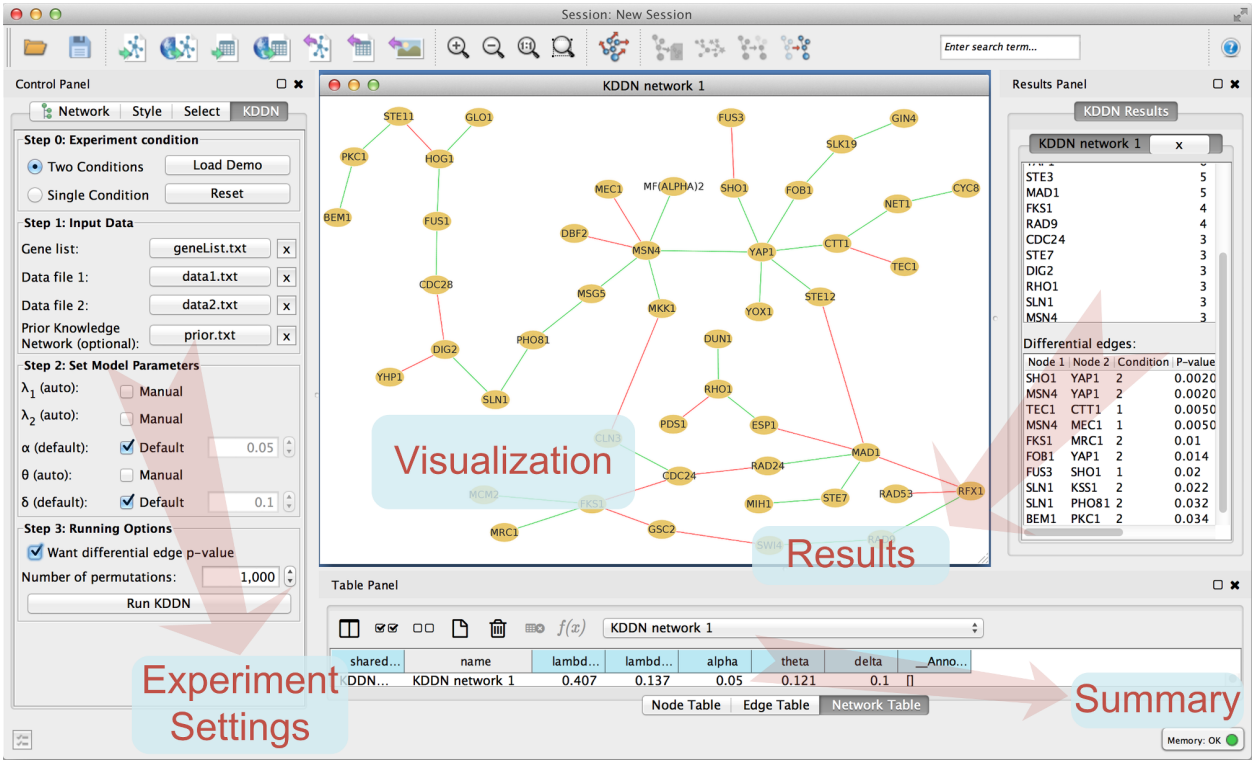


Figure 2.20: Interface design of KDDN app and a result of yeast oxidative stress response, see section 3.1 for more details.

tions and real data, KDDN demonstrates its effectiveness in revealing significant topological changes with knowledge incorporation capability. We expect KDDN to become a convenient tool to give a network perspective of biological data.

The complied package and source code can be found at `http://apps.cytoscape.org/apps/kddn`. There are three equivalent ways to install KDDN app to your Cytoscape 3.x.

1. Download the KDDN jar file and put it under directory $HOME/CytoscapeConfiguration/3/apps/installed;
2. Install from Cytoscape's App Manager by searching "KDDN";
3. Install directly from Cytoscape App Store following the instructions on `http://apps.cytoscape.org/help/getstarted_app_install`.

Once installed, KDDN can be accessed from Cytoscape Apps menu or control panel.

2.5.3 Workflow

KDDN takes input data and runs KDDN algorithm to construct the dependency network. When two conditions data are available, KDDN identifies the statistically significant condition-specific edges to provide insights into system dynamics.

A KDDN experiment can be performed following 4 steps:

1. Determine the experiment condition;
2. Provide input data;
3. Choose the way model parameters to be specified;
4. Configure running options.

We will use the interface shown in Figure 2.21 to explain the steps carrying out an experiment.

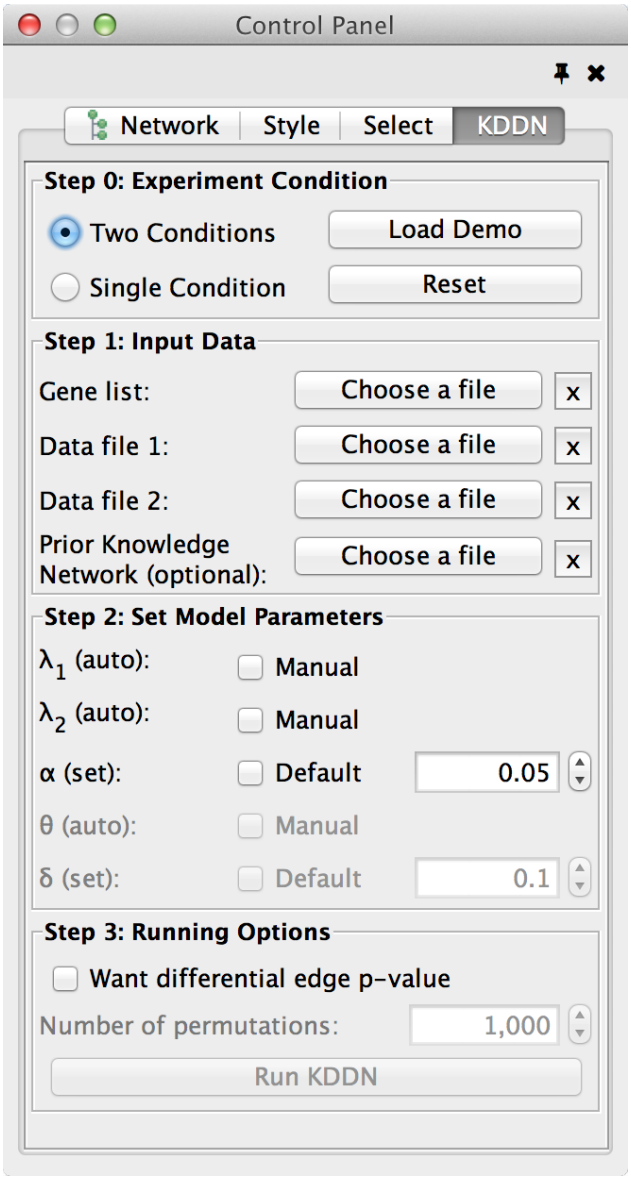


Figure 2.21: KDDN configuration interface.

Step 0: Experiment Condition

The left half of step 0 panel specifies experiment conditions. KDDN can work with both single condition data and two conditions data. Users can choose the proper experiment condition according to the needs.

Two Conditions

Data for two conditions are provided. KDDN learns the differential edges between the two conditions. Red edges highlight connections exist only under condition 1 and green edges highlight connections exist only under condition 2.

Single Condition

Only data of condition 1 is used (or provided). KDDN learns a static network structure with grey edges showing the connections.

The right half of step 0 panel provides buttons to load demo dataset and reset KDDN. A demo dataset is provided for users to understand the usage and explore the settings. All settings also can be easily reset to initial status.

Load Demo

Load the demo dataset including all required or optional files, and set all parameters to auto mode. The dataset contains 18 genes. Users can cancel the demo dataset or replace with new dataset (make sure the number of genes are the same).

Reset

Reset KDDN to initial status. This includes reset of all files choosers and reset of all parameters.

Step 1: Input Data

Biological system profiling data such as gene expression data and prior knowledge of network such as KEGG pathway are provided through the step 1 interface.

Gene list

A text file containing a list of nodes (genes, proteins) involved in the study. One node

per line. No row or column names. The node names will also be used in the visualized network.

Data file 1

Profiling data (gene expression, protein expression, RNA-seq RPKM, etc.) of condition 1 in a text file with no row or column names. One node per row, corresponding to the same order in gene list file. Tab or comma delimited. Data should be properly normalized.

Data file 2

Profiling data (gene expression, protein expression, RNA-seq RPKM, etc.) of condition 2 in a text file with the same format requirement with data file 1. Not required or ignored when experiment condition is set to single condition. In single condition experiment setting, data file 2 related fields are disabled.

Prior knowledge network

Prior knowledge of network topology in KGML format[1] or a text file with no row or column names. If text file is used, one edge per row with two node names separated by tab or comma. KGML xml file can be downloaded from KEGG website. This file is optional. If provided, KDDN will use the knowledge in learning, otherwise it relies solely on data.

Chosen files can be canceled using the cross button to the right of file chooser.

Step 2: Model Parameters

Step 2 guides users to tune model parameters of KDDN. Parameters can be set in auto/default mode or manual mode. Parameters are enabled or disabled according to specific settings.

$\boldsymbol{\lambda}_1$

[1] http://www.kegg.jp/kegg/xml

It controls the overall density of the learned network. Larger λ_1 leads to sparser networks. The default is in auto mode where it is calculated automatically from data, as explained in Section 2.3.3. When manual mode is chosen, a spinner shows up to allow user to set a value.

λ_2

It controls the density of identified differential edges. It is enabled only in two conditions experiments. Larger λ_2 leads to fewer differential edges. The default is in auto mode where it is found automatically according to the value of α, as explained in Section 2.3.3. When manual mode is chosen, a spinner shows up to allow user to set a value, and α setting is disabled.

α

The desired significance level of differential edges. It is used to find the corresponding value of $_2$ automatically. It is enabled only in two conditions experiments and $_2$ is set to auto. Check the default checkbox resets it to 0.05.

θ

The degree of prior knowledge incorporation. It is enabled only when the optional prior knowledge network file is provided. The default is in auto mode where it is found automatically according to the value of δ, as explained in Section 2.3.3. When manual mode is chosen, a spinner shows up to allow user to set a value, and δ setting is disabled.

δ

The maximum tolerable deviation in the worst case scenario of prior knowledge incorporation. It is used to find the corresponding value of θ automatically, as explained in Section 2.3.3. It is enabled only when the optional prior knowledge network file is provided and θ is set to auto. Check the default checkbox resets it to 0.1.

Step 3: Running Options

When the input data meets the minimum requirements corresponding to experiment condition, run button is enabled to allow users to carry out experiments.

In two conditions experiments, differential network is identified and users can opt in to calculate the significance of the differential edges, otherwise differential network is still calculated but the significance of differential edges are missing.

Want differential edge p-value

Enabled only in two conditions experiments. Check it will add more running time to perform permutation test to get edge-specific p-values for differential edges.

Number of permutations

The number of permutations used. It is enabled only in two conditions experiments and want differential edge p-value is checked.

After going through all steps, a final hit of run button will initiate the analysis. A progress bar informs users the progress of the three most time-consuming stages (if necessary according to settings): finding λ_2, finding θ and calculating p-value.

Output

After the calculation is completed, a network with an unique id is created and visualized, and summary information is provided in results panel and table panel. Results of demo data is shown in Figure 2.22.

Node names in the network view are the same with the ones in gene list file. In single condition experiment, static connections are displayed by gray edges. In two conditions experiment, differential network is displayed with red edges represent connections exist only under condition 1 and green edges represent connections exist only under condition 2.

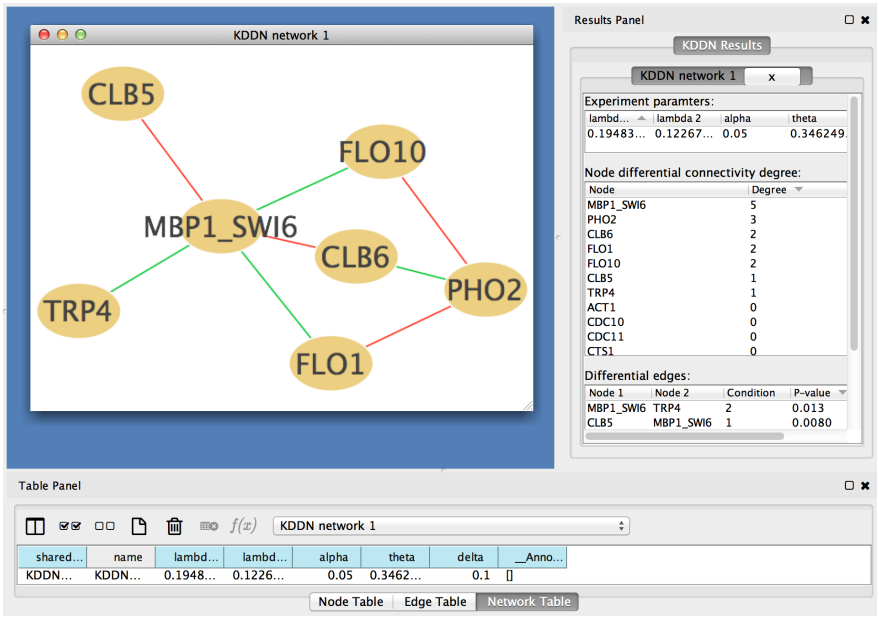


Figure 2.22: Demo results showing network view, table panel and results panel information.

There are three tables in tabbed results panel, experiment parameters, node connectivity degree and differential edges (in two conditions experiments),

Experiment parameters

A table summarized the parameters used in the experiment.

Node connectivity degree

A table lists the degree of all nodes. In single condition experiments the degree is the number of static edges connected to the node. In two conditions experiments the degree is the number of differential edges connected to the node.

Differential edges

A table lists all differential edges, if experiment is two conditions. If p-value is calculated, a p-value column is added to the table.

2.6 Conclusions

In this chapter we propose a general machine learning framework to tackle the structural learning of Gaussian graphical model with condition-specific topological changes. To address the challenges concerning differential network inference using data-knowledge integrated approaches, we formulated the problem of learning the conditionspecific network structure and topological changes as a convex optimization problem. Model regularization and prior knowledge were utilized to navigate through the vast solution space. An efficient algorithm was developed to make the solution scalable by exploring the special structure of the problem. Prior knowledge was carefully and efficiently incorporated in seeking the balance between the prior knowledge support and data-derived evidence. The proposed method can efficiently utilize prior knowledge in the network inference while remaining robust to false positive edges in the knowledge. The statistical significance of rewiring and desired type I error rate were assessed and validated. We evaluated the proposed method using synthetic data sets in various cases to demonstrate the effectiveness of this method in learning both common and differential networks, and the simulation results further corroborated our theoretical analysis. To make the method conveniently accessible for the research community, we developed a software suite that runs as a Cytoscape app. It provides straightforward configurations and comprehensive results to facilitate the adoption of the method in biomedical research.

Chapter 3

Applications of KDDN In Broader Biological Questions

KDDN has been adopted in a wide range of biomedical studies and provided insights into the biological problems from different aspects. This chapter gives a comprehensive landscape of the usage of KDDN through seven real studies: yeast stress response, breast cancer recurrence, T cell signaling in muscular dystrophy, transcription factor dynamic binding, network comparison of chromosome stable and unstable ovarian cancer patients, origin of ovarian cancer identification based on network distance, and brain tumor drug treatment study assisted by medical imaging.

3.1 Yeast Response to Oxidative Stress

To test the utility of KDDN in real biological study, we applied it to the public data set GSE7645. This data set used budding yeast *Saccharomyces cerevisiae* to study the genome-wide response to oxidative stress imposed by cumene hydroperoxide (CHP). Yeast cultures

were grown in controlled batch conditions, in 1 L fermentors. Three replicate cultures in mid-exponential phase were exposed to 0.19 mM CHP, while three non-treated cultures were used as controls. Samples were collected at t=0 (immediately before adding CHP) and at 3,6,12,20,40,70 and 120 min after adding the oxidant. Samples were processed for RNA extraction and profiled using Affymetrix Yeast Genome S98 arrays.

We analyzed the network changes of cell cycle related genes with structural information from the KEGG yeast pathway as prior knowledge. We added a well studied yeast oxidative stress response gene *Yap1* (Costa *et al.*, 2002; Ikner and Shiozaki, 2005; Jamieson, 1998; Kuge *et al.*, 1997) and related connections gathered from the Saccharomyces Genome Database (Cherry *et al.*, 2012) to the knowledge network. The learned differential network result is shown in Figure 3.1, in which nodes represent genes involved in the pathway rewiring, and edges show the condition-specific connections. Red edges are connections in control and green edges are connections under stress.

Oxidative stress is a harmful condition in a cell, tissue, or organ, due to the failure of antioxidant defense system to effectively remove reactive oxygen molecules and other oxidants. The result shows that *Yap1*, *Rho1* and *Msn4* are at the center of the network response to oxidative stress; they are activated under oxidative stress and many connections surrounding them are created (green edges). *Yap1* is a major transcription factor that responds to oxidative stress (Costa *et al.*, 2002; Ikner and Shiozaki, 2005; Jamieson, 1998; Kuge *et al.*, 1997). *Msn4* is considered as a general responder to environmental stresses including heat shocks, hydrogen peroxide, hyperosmotic shock, amino acid starvation (Causton *et al.*, 2001; Gasch *et al.*, 2000). *Rho1* is known to resist oxidative damage and facilitate cell survival (Lee *et al.*, 2011). The involvement of these central genes captured the dynamic response of how yeast cell sense and react to oxidative stress. The edge between *Yap1* and *Ctt1* under stress grants more confidence to the result. *Ctt1* acts as an antioxidant in response to oxidative stress (Grant *et al.*, 1998), and the coordination between *Yap1* and *Ctt1* in protecting cells

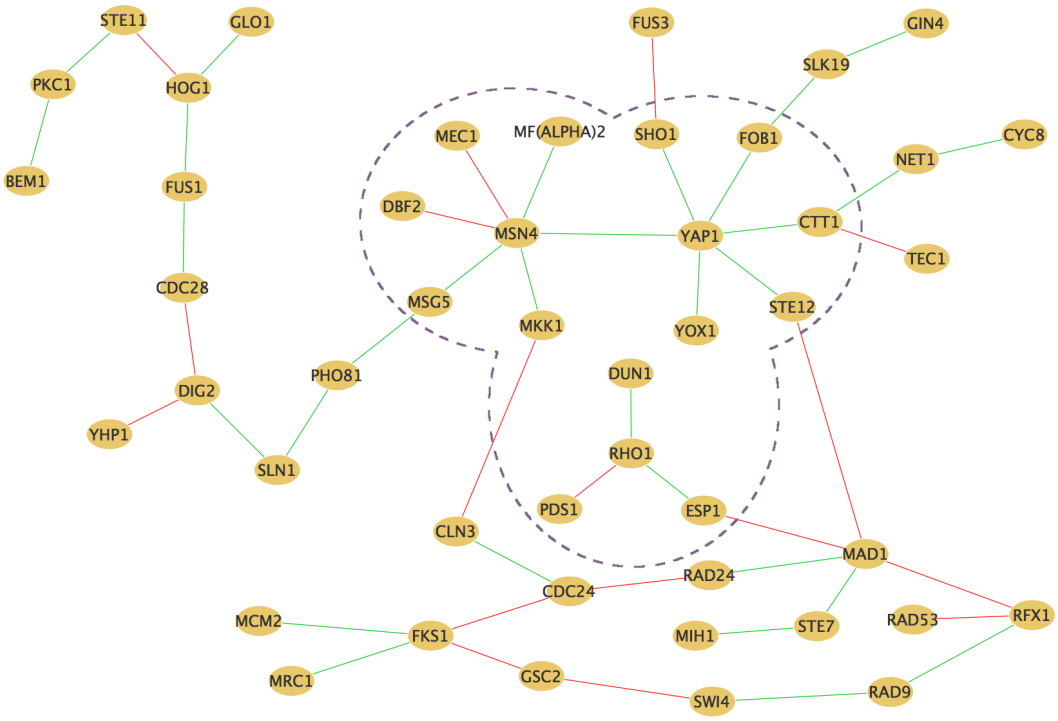


Figure 3.1: Differential dependency network in budding yeast reflects the cell cycle response to oxidative stress.

from oxidative stress is well established (Lee *et al.*, 1999). The establishment of interaction between Hog1 and Fus1 is also observed in prior study (Staleva *et al.*, 2004). This result depicted the dynamic response of yeast when exposed to oxidative stress and many predictions are supported by previous studies, which validated the effectiveness of the methods in revealing underlying mechanisms and providing potentially novel insights. These insights would be largely missed by conventional differential expression analysis as the important genes *Rho1*, *Msn4*, Yap1 and Ctt1 ranks 13, 20, 64 and 84 among all 86 involved genes based on t-test p-values.

3.2 Apoptosis Signaling in Recurred Breast Cancer

Network rewiring analysis can be utilized to study the mechanistic differences between long-term outcomes of a disease and help find the underlying key players that cause these differences. For example, 50% of estrogen receptor positive breast cancers recur, but the mechanisms involved in causing recurrence remain unknown. Understanding of the mechanisms of breast cancer recurrence can provide critical information for early detection and prevention. We used gene expression data from a clinical study (Loi *et al.*, 2007) to learn differences in the apoptosis pathway in primary tumors between later recurring and non-recurring patients. We compared the pathway changes in tumors obtained from patients whose breast cancer recurred within 5 years after treatment and from patients who remained recurrence free for at least 8 years after treatment. There were 47 and 48 tumor samples in the recurring and non-recurring groups, respectively. Gene expression data were generated using Affymetrix U133A arrays. We used the apoptosis pathway from KEGG as prior knowledge.

We overlaid the differential network over the KEGG apoptosis pathway to observe the signaling patterns. Following the same presentation, red edges are connections established in recurring patients, and green edges are connections unique to non-recurring patients. Dif-

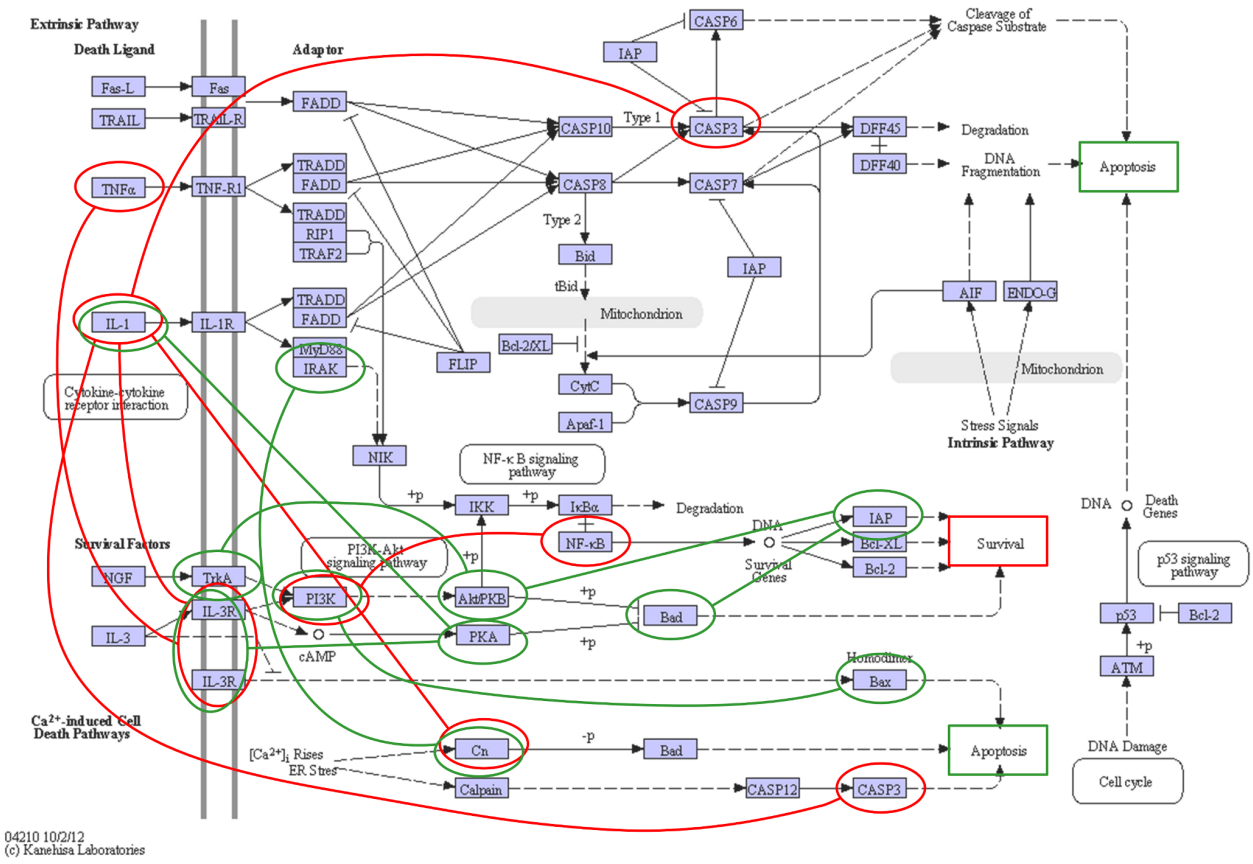


Figure 3.2: Differential dependency network of breast cancer recurrence aligned with KEGG apoptosis pathway.

ferences in the signaling among genes in the apoptosis pathway between tumors in patients that subsequently recurred or remained cancer free are shown in Figure 3.2.

Three inflammatory/immune response genes (*IL1B*, *NFκB* and *TNFα*) that are all linked to increased resistance to breast cancer treatment were identified in the recurring tumors. These genes formed a path to inhibit proapoptotic *CASP3* and *PPP3R1* (Su, et al., 2012), and to activate the pro-survival genes *PIK3R5* or *CSF2RB* that maintain cell survival. In contrast, green edges that were present in non-recurring tumors form paths to both anti-apoptotic *XIAP*/*AKT2* and proapoptotic *BAX* and *BAD* gene functions.

In the recurring breast cancers (red edges), the molecular activities mainly affect the initial

apoptotic signals outside the cell and within cell membrane (ligands and their receptors), while inside the cell there is no clear signaling cascade affected to determine cells fate. The only route affected within the cell is *IL1B*-induced inhibition of pro-apototic *CASP3*. In the non-recurring breast cancer, the affected network involves both signals received from activation of the membrane receptors and a cascade of signaling path inside a cell to promote apoptosis as well as survival. Since a balance between apoptosis and survival is necessary for damaged cells to be eliminated and repaired cells to survive (Murphy *et al.*, 2000), it is logical that both pathways would be activated concurrently. The apoptosis pathway rewiring analysis identified key mechanistic signaling differences in the tumors between recurring and non-recurring patients. These differences provide a promising ground for novel hypothesis to determine factors affecting breast cancer recurrence.

3.3 T Cell Signaling in Juvenile Dermatomyositis

We accessed a 125 patient muscle biopsy U133A mRNA profiling dataset containing 13 diagnostic groups of patients with specific muscle disorders (Bakay *et al.*, 2006). We applied KDDN to two of the largest groups: normal controls (NHM) with 18 samples and juvenile dermatomyositis (JDM) with 25 samples. Juvenile dermatomyositis is an autoimmune disorder of muscle in pediatric patients. T cell receptor signaling pathway is an important pathway in the pathophysiology of JDM and we analyzed the network based on it, shown in Figure 3.3.

CD8+ T cells are a well-recognized inflammatory infiltrate in JDM muscle. A novel and potentially important sub-network was detected by KDDN due to the role of NFAT proteins in mediating membrane signals to nuclear cellular reactions. *NFAT5* is upregulated in JDM. *NFAT5* is known to respond to osmostic stress, relaying this signal to the nucleus in muscle and other tissues (Zhang *et al.*, 2003; Hernndez-Ochoa *et al.*, 2012). The connection to T

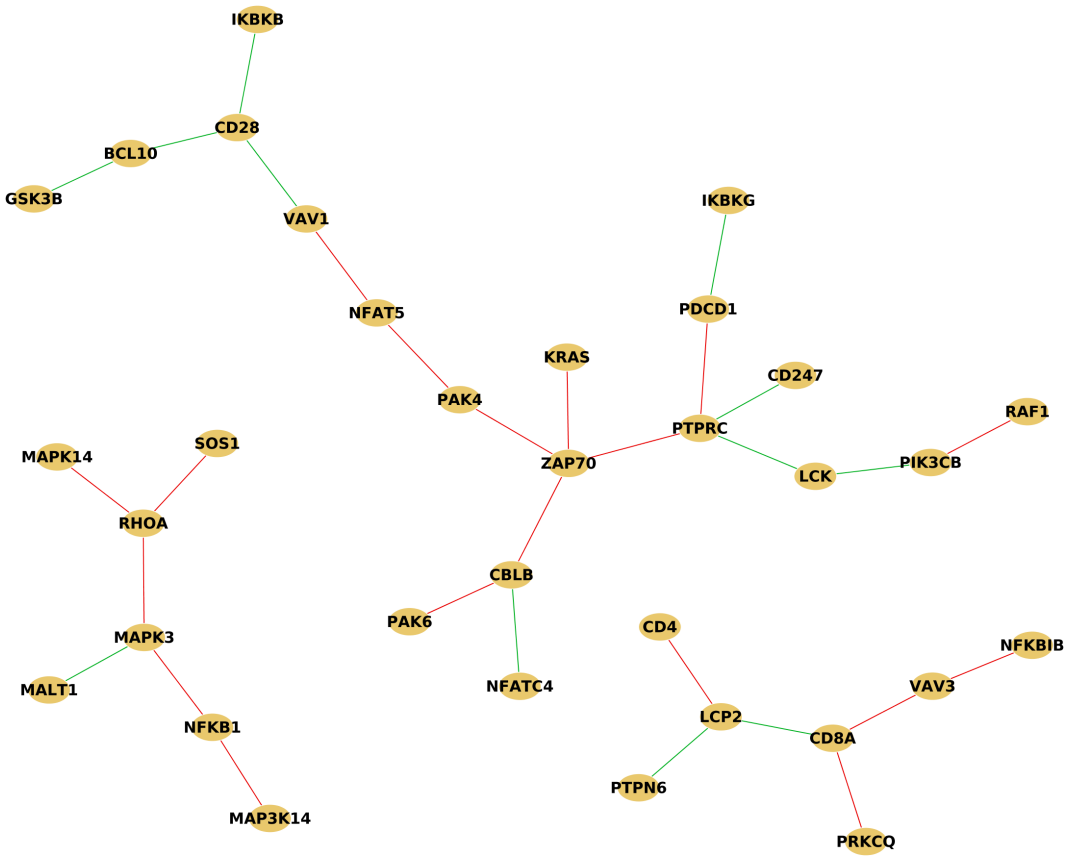


Figure 3.3: NHM *vs.* JDM in T cell receptor signalling pathway. Red edges are in normal, and green edges are in JDM.

cell pathways via *ZAP70* likely reflects an interaction between infiltrating T cells and pro-inflammatory myofibers, and points out a possible ischemic/osmotic pathway that may be an important contributor to JDM.

3.4 Estrogen Receptor α Binding Regulation

Nuclear receptor estrogen receptor alpha (ERα) controls the expression of target genes through either direct or tethered DNA binding. It is important to study the differential binding patterns of ERα under different conditions to understand the mechanisms of ERα

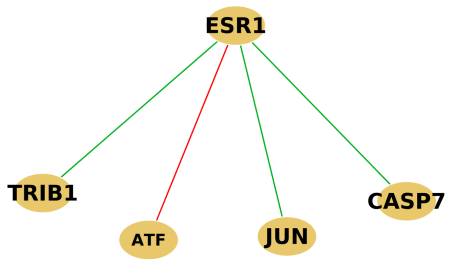


Figure 3.4: Wild type ERα *vs.* ERα mutated shows differential DNA binding patterns via direct and tethered binding. Red edges are in wild type, and green edges are in ERα mutated.

binding. We used a public data set (Stender *et al.*, 2010) profiling the gene expression of wild type and ERα mutated cell lines to discover the different binding targets.

The wild type ERα is expected to regulate target gene expression via direct binding while the ERα muated regulation is expected to be accomplished via tethered DNA binding. We selected a set of genes from literature documented ERα targets (Klinge, 2001; Lin *et al.*, 2004), expression inferred possible targets (Stender *et al.*, 2010) and targets found in database (Klinge, 2001), including *ESR1*, *TFF1*, *EBAG9*, *CASP7*, *GREB1*, *SP1*, *JUN*, *FOSB*, *ATF*, *CEBPB*, *PITX1*, *GADD45A*, *TRIB1*, *SOX9* and *HBEGF*. Results are shown in Figure 3.4.

The identified target binding relations pointed out the possible binding schemes as green - tethered and red - direct or tethered. *JUN* and *ATF* are known to be involved in tethered binding (Umayahara *et al.*, 1994; Kushner *et al.*, 2000), while *CASP7* is known as a direct target (Klinge, 2001; Lin *et al.*, 2004). These results demonstrated the ability of KDDN to work with transcription factor-target information and identify condition-specific transcription factor binding.

3.5 Impact of Chromosome Instability on Ovarian Cancer Gene Networks

We then apply our methods to analyze TCGA ovarian cancer data set to test the usefulness of our approach in biological studies. We define the Chromosome Instability Index (CIN Index) to characterize the genetic variations of samples. Using CIN Index, we divide the samples into groups that are genetically most unstable and most stable and compare their transcriptional network differences.

3.5.1 Chromosome Instability Index

Many diseases have genetic causes beneath the scene of transcriptional level abnormalities. It is believed that genetic aberrations play a role in tumorigenesis by increasing the rate of chromosome mutations, including deletion and amplification of genes involved in critical cellular processes. Chromosomal instability (CIN) is a defining characteristic of most human cancers (Nowak *et al.*, 2002). However the role of CIN in tumorigenesis remains poorly understood and people have few clues on the interaction between genetic variations and transcriptional outcomes that lead to phenotype differentiations (Ricke *et al.*, 2008).

We use TCGA level 3 copy number data to calculate the CIN Index for 556 samples. Each of the 22 chromosomes will be assigned a CIN Index to reflect its stability in terms of number of break points and amplitude of amplifications or deletions. The CIN Index is defined as the number of segments weighted by amplitude of alteration. For chromosome i, denote the largest amplitude of all samples as $A^{(i)}$ and the smallest amplitude of all samples as $D^{(i)}$. In sample j there are $N_j^{(i)}$ segments and $n_{j,k}^{(i)}$ probes in the k-th segment with amplitude $m_{j,k}^{(i)}$, $k = 1, 2, \cdots, N_j^{(i)}$. The cutoffs of amplification and deletion are τ_1 and τ_2 respectively. Then

the CIN of chromosome i for sample j is defined as:

$$CIN_j^{(i)} = \frac{\sum\limits_{m_{j,k}^{(i)}>\tau_1} (m_{j,k}^{(i)} - \tau_1) n_{j,k}^{(i)} + \sum\limits_{m_{j,k}^{(i)}<\tau_2} (\tau_1 + \frac{(\tau_2 - m_{j,k}^{(i)})(A^{(i)} - \tau_1)}{\tau_2 - D^{(i)}}) n_{j,k}^{(i)}}{\sum_{k=1}^{k=N_j^{(i)}} n_{j,k}^{(i)}}, \tag{3.1}$$

in which $\tau_1 = 2.5$ and $\tau_2 = 1.5$ are parameters we use here.

We use a vector of CIN to represent the chromosome instability of a sample.

$$\mathbf{CIN}_j = (CIN_j^1, CIN_j^2, \cdots, CIN_j^{22})^T. \tag{3.2}$$

3.5.2 Network Comparison between Chromosome Stable and Unstable Patients

We want to compare the transcriptional network differences between chromosome stable and unstable samples. To divide the samples, we first construct two reference vectors, **HighCIN** and **LowCIN**:

$$\begin{aligned} \mathbf{HighCIN} &= (\max \mathbf{CIN}^1, \max \mathbf{CIN}^2, \cdots, \max \mathbf{CIN}^{22})^T \\ \mathbf{LowCIN} &= (\min \mathbf{CIN}^1, \min \mathbf{CIN}^2, \cdots, \min \mathbf{CIN}^{22})^T. \end{aligned} \tag{3.3}$$

We select 20 samples with **CIN** closest to **HighCIN** in Euclidean distance as unstable group, and 20 samples closest to **LowCIN** as stable group.

Our method is applied to learn the differential network structure between early and late groups in different pathways using corresponding Level 2 gene expression data. Pathway structures retrieved from KEGG pathway database (Kanehisa and Goto, 2000) are used as prior knowledge.

Pathways analyzed include: Apoptosis, Cell Cycle, Wnt signaling and Tgf-β signaling. Same node and edge notations are use in the result presentation.

Identified differential networks are shown in Figures 3.5 to 3.8.

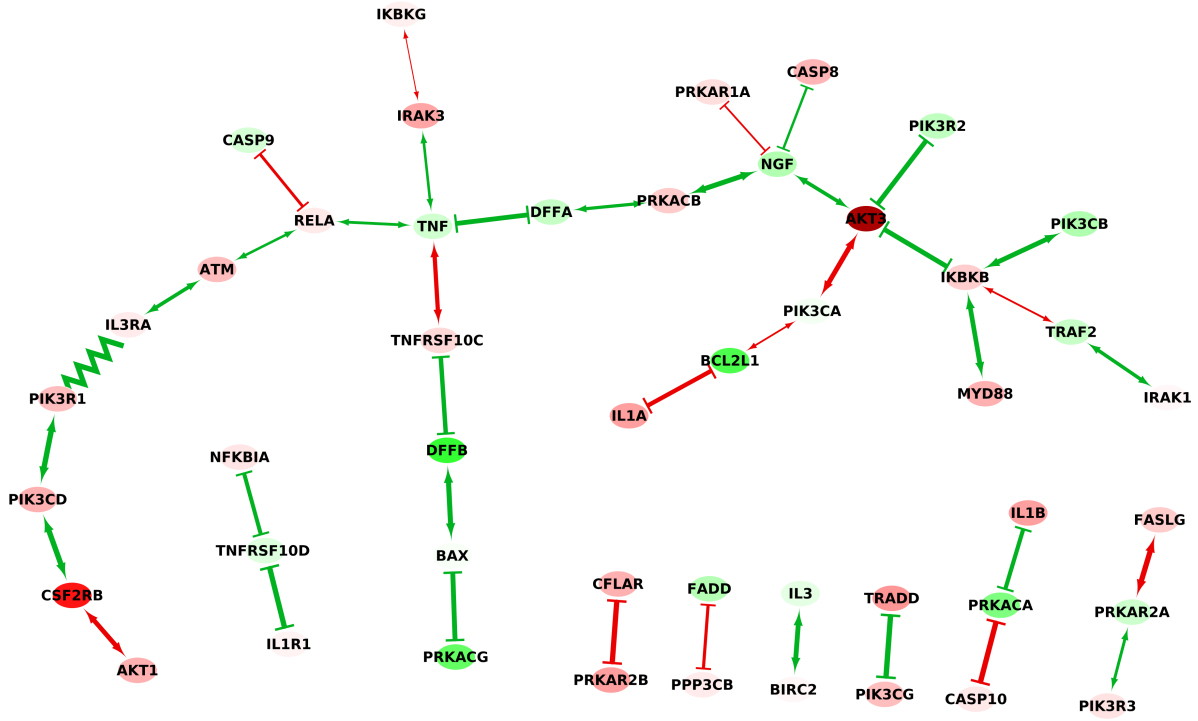


Figure 3.5: Differential network of ovarian cancer Apoptosis pathway between high-CIN and low-CIN patients.

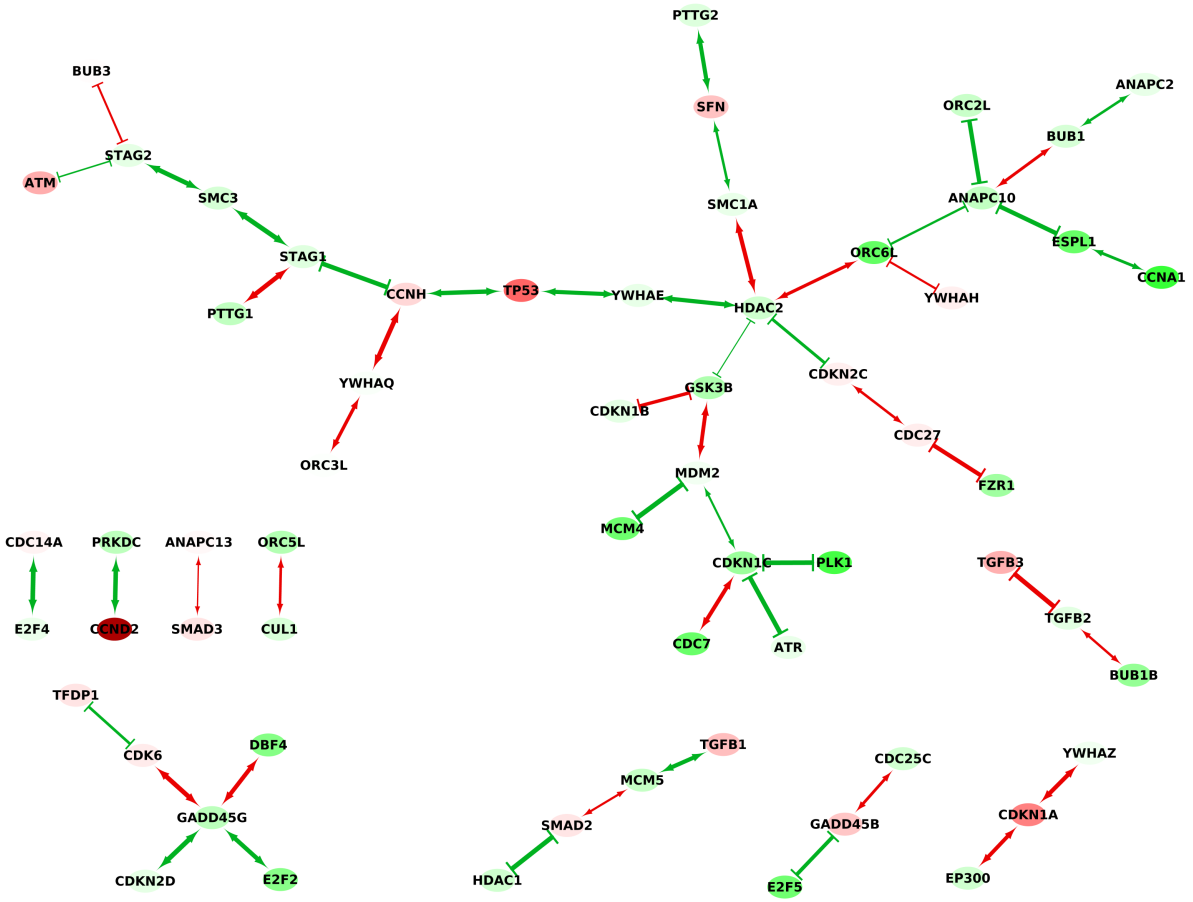


Figure 3.6: Differential network of ovarian cancer Cell Cycle pathway between high-CIN and low-CIN patients.

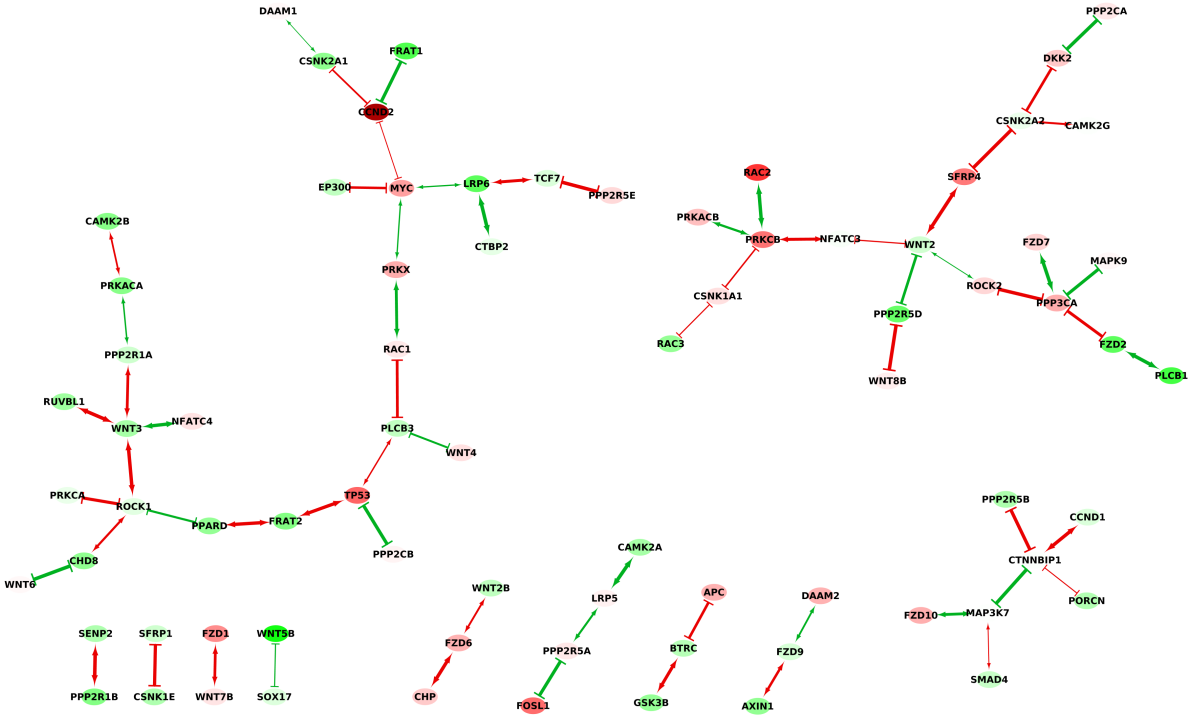


Figure 3.7: Differential network of ovarian cancer Wnt signaling pathway between high-CIN and low-CIN patients.

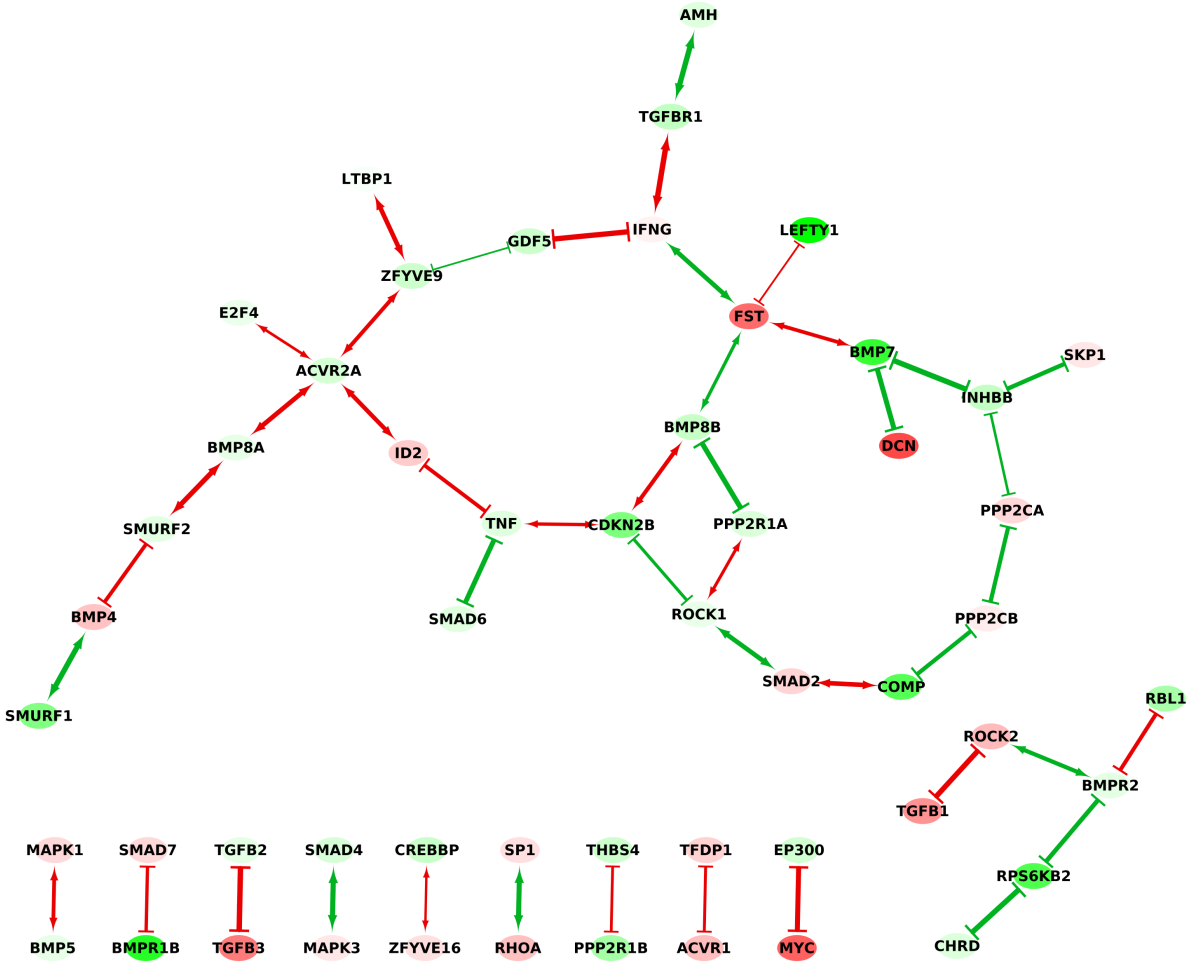


Figure 3.8: Differential network of ovarian cancer Tgf-β pathway between high-CIN and low-CIN patients.

3.6 Network Analysis to Identify Origin of Ovarian Cancer

Identifying the origin of tumor is a crucial yet difficult task in cancer research. So far, significant efforts have been made, through experimental or computational approaches, to reveal the origin of high-grade ovarian serous carcinoma. As evidences are needed to show how tumor cells evolve from some original normal cells, an intuitive reasoning is that the tumor cells and normal cells should have some biological traits in common. We look for hints along this direction by comparing gene expression profiles and related network topology of tumor and its origin candidates.

3.6.1 Hypotheses on the Origin of High-grade Ovarian Serous Carcinoma

The accepted view of ovarian carcinogenesis is that carcinoma begins in the ovary, and then spreads to the pelvic and abdominal cavities before metastasizing to distant sites. It has therefore been reasoned that survival for this highly lethal disease could be improved by developing screening methods that detect disease when it is confined to the ovary. To date, however, no prospective randomized trial of any ovarian cancer screening test(s) has demonstrated a decrease in mortality. Based on studies performed during the last decade, Kurman and Shih (2009) found that one of the rapidly growing highly aggressive ovarian cancers, "high-grade serous carcinoma" (HG) is only rarely confined to the ovary, and they believed that one of the main reasons for this is that the dogma underlying ovarian carcinogenesis is flawed. Although the reasons for this are not entirely clear, possible biological explanations exist, including that HG originates in some extra ovarian site, notably, the fallopian tube. The exploration of the controversies on the origin of HG may shed light on new medical

approaches to detection and treatment of ovarian cancer.

Given the importance, novelty and uniqueness of the related biological hypotheses and limitation of biomedical experiments, *in silico* methods are on demand to test the biological hypotheses. In the last decades, machine learning has become a fundamental research topic in the progression of computing applications in health care and biomedicine; while gene expression profiling is widely applied in cancer research, and is believed, by physicians, that expression profiling promises to be an important tool for identifying the unknown source of cancer (Varadhachary and Raber, 2009).

Although there are some studies (Marquez *et al.*, 2005; Bowen *et al.*, 2009) proposed to compare the gene expression profile of HG and those for normal organs, they are designed in an imperfect way by failing to measure the similarity between HG and normal organs directly based on expression value (Marquez *et al.*, 2005), incomplete comparison between HG and normal organs (Bowen *et al.*, 2009), and ignoring the tissue contamination which may bring noticeable bias to their results. Here the network-based pathway analysis compares the distances between HG and possible origin organs in terms of network structure differences across KEGG pathways. Hypothesis testing is carried out the test whether one of the distances are consistently smaller than the others across all pathways. The experiment results support the biological hypothesis that fallopian tube (FT) is more likely to be the origin of HG, which may shed light on new medical approaches for detection and treatment of ovarian cancer.

3.6.2 Transcriptional Network Comparison Shows HG and FT Have the Most Similar Gene Regulatory Networks

Gene networks are very complex, context-specific and subject to dynamic changes, and different network topology inference methods may lead to very different inference results. Instead

of identification of the underlying structures, detection of network differences will give more sparse and accurate results. In this section we use the knowledge guided differential dependency network in Chapter 2 to identify the most significantly different network topologies between tumor and origins to reveal how different their gene regulatory networks are.

We try to find some computational clues to identify which normal organ is the true origin where HG starts to evolve. It is prudent to hypothesize that HG tumor cells may retain certain gene network structure characteristics from their normal ancestral cells, and gene network topological differences between HG and its real origin will be slighter than with non-origins.

Transcriptional network distance is defined based on the result of KDDN. KDDN result of tumor and normal network comparison in a given pathway consists of a subset of genes in the pathway and certain number of connections between the genes. The connections in the differential network exist either in tumor or normal, and the size of the differential network reflects how different the two networks are. For a given underlying network, a pathway in our case, the nodes in the network are known, but the edges are unknown. Compared with nodes, differential edges are more informative, as the same nodes can have very different connections.

In a pathway if the networks of tumor and normal are identical, the differential network between them is empty, otherwise is composed by the K differential edges. We use graph edit distance as a measurement for the dissimilarity between the tumor and normal networks. The network distance $d_{network}$ between tumor and normal is defined as the ration between K and a normalization factor, which here is the size of the largest common network in all comparisons, C, which can reflect the size of underlying network.

$$d_{network} = \frac{K}{C} \tag{3.4}$$

The network distance is non-negative. It satisfies the conditions for a metric, including non-negativity, identity of indiscernibles, symmetry and subadditivity.

By extracting the topologically changed parts of transcriptional networks of tumor and normal samples, network distances between tumor and normal is assessed. The one among FT, OSE and Endo with the least distance to tumor HG has the most similar transcriptional network structure, and therefore is considered to be the origin of HG.

To make the network comparison more biologically meaningful, genes used in the test should have regular interactions at the first place. Biological pathway is a set of genes interacting with each other to form a network in order to achieve aggregated function that cannot be done individually. In cases when environment changes, pathway interaction can be destroyed or re-organized. KEGG (Kanehisa and Goto, 2000) is an open database collecting such kind pathways from reliable research results. Pathways of human from KEGG are used as test genes in the experiment. Gene number in 200 KEGG pathways varies from 1 to 257. We discard pathways with less than 10 genes, and use the left 171 pathways as 171 test gene sets to evaluated network distance separately.

In every pathway, the genes are assumed to function jointly and the network distances between HG and every one of the possible origins reflect their functional distances. Three distances are calculated, *i.e.*, the network distance between HG and FT, $d_{i:HG,FT}$, the network distance between HG and Endo, $d_{i:HG,Endo}$, and the network distance between HG and OSE, $d_{i:HG,OSE}$, and i is the index of pathways. For a particular pathway, the three distances can be considered as a group of repeated measures in different scenarios, each of which is an examination of the pathway under a given setting.

The aim of finding the origin of HG can be achieved by determining if one of the three normal organs is consistently the closest to HG in all pathways in terms of network distance. Suppose the structure of gene regulatory network of one normal organ tends to act more

similar to that of HG than the other two normal organs, the network distance between it and HG will be smaller than the network distances between the other two and HG in most of the pathways, which can serve as a strong evidence to support it to be the origin of HG.

We test if $d_{i:HG,FT}$ is consistently smaller than both $d_{i:HG,Endo}$ and $d_{i:HG,OSE}$. The test procedure fits into paired difference test well, by treating pathways as samples and network distances in each pathway as paired measures. The test null hypotheses are $d_{HG,FT} = d_{HG,Endo}$ and $d_{HG,FT} = d_{HG,OSE}$, and the alternative hypotheses are $d_{HG,FT} < d_{HG,Endo}$ and $d_{HG,FT} < d_{HG,OSE}$, as we need one-sided tests. Only if both two null hypotheses are rejected, we can reach a conclusion.

We adopted the Wilcoxon signed-rank test, a non-parametric statistical hypothesis testing method for the case of two repeated measurements on a single sample, which can be used as an alternative to the paired Student's t-test when the population cannot be assumed to be normally distributed, to test the hypotheses. The test rejected the null hypothesis $d_{HG,FT} = d_{HG,Endo}$ with $p = 0.00031$, and the null hypothesis $d_{HG,FT} = d_{HG,OSE}$ is also rejected, with $p = 4.93 \times 10^{-5}$. Test results show that FT has the smallest network distance to HG in abundant biological networks. The similarity between the transcription networks of HG and FT highlighted the possibility that FT is the origin of HG which supports the proposed hypothesis.

3.7 Integrated Study on Medulloblastoma Treatment using ATO

Medulloblastoma (MB) is the most common and highly malignant childhood brain tumor that has a tendency to spread throughout the brain and central nervous system. Current therapies leave a majority of survivors with serious auditory and neuro-cognitive deficits,

underscoring the need for the development of more effective treatments for MB. The origin of MB is complex and is often characterized by alterations in survival and cell cycle regulatory pathways and genes. For example, Sonic Hedgehog pathway is normally inactive in the mature cerebellar neuron, while its activation by the Smoothened (Smo) signal leads to unchecked tumor cell proliferation (Berman *et al.*, 2002).

ATO is an FDA approved, second line therapeutic used for treating certain cancers and we have shown it can effectively treat MB, preclinically via targeting hypothesized MB signaling mechanisms (de Bont *et al.*, 2008). There is a significant information gap in the literature on the efficacy and therapeutic molecular biology of MB treated by ATO. Furthermore, the previous *in vivo* studies using the ND2-SmoA1 MB model was primarily an endpoint study, and the information on kinetics of tumor growth was missing, accurate tumor growth analyses using longitudinal imaging was lacking, and a multiplatform systems biology approach was not rigorously applied.

We included KDDN into a data acquisition and analytics pipeline to integrate multiplatform data (survival, MRI, DNA copy number, gene expression, and protein) for studying the treatment of MB using ATO. The workflow among modules and their systematic cooperation is illustrated in Figure 3.9. MRI-imaging as a physical examination tool defines the phenotypic sample groups for all subsequent analysis and labels the gene expression data, RPMA protein data, survival data and tumor growth data. MRI-based tumor volume estimates were used to establish the efficacy of ATO treatment based on the known groups groups. Correspondingly, to corroborate with the other studies and to investigate possible mechanisms, KDDN analysis uses gene expression data or RPMA protein data to detect significant rewiring of transcriptional regulatory networks between different phenotypes or drug treatments. As a complete demonstration of the integrated approach, the animals on the ATO study were longitudinally interrogated by MRI for tumor growth kinetics and survival analysis, followed by KDDN analysis based on RPMA protein data for mechanistic reasons

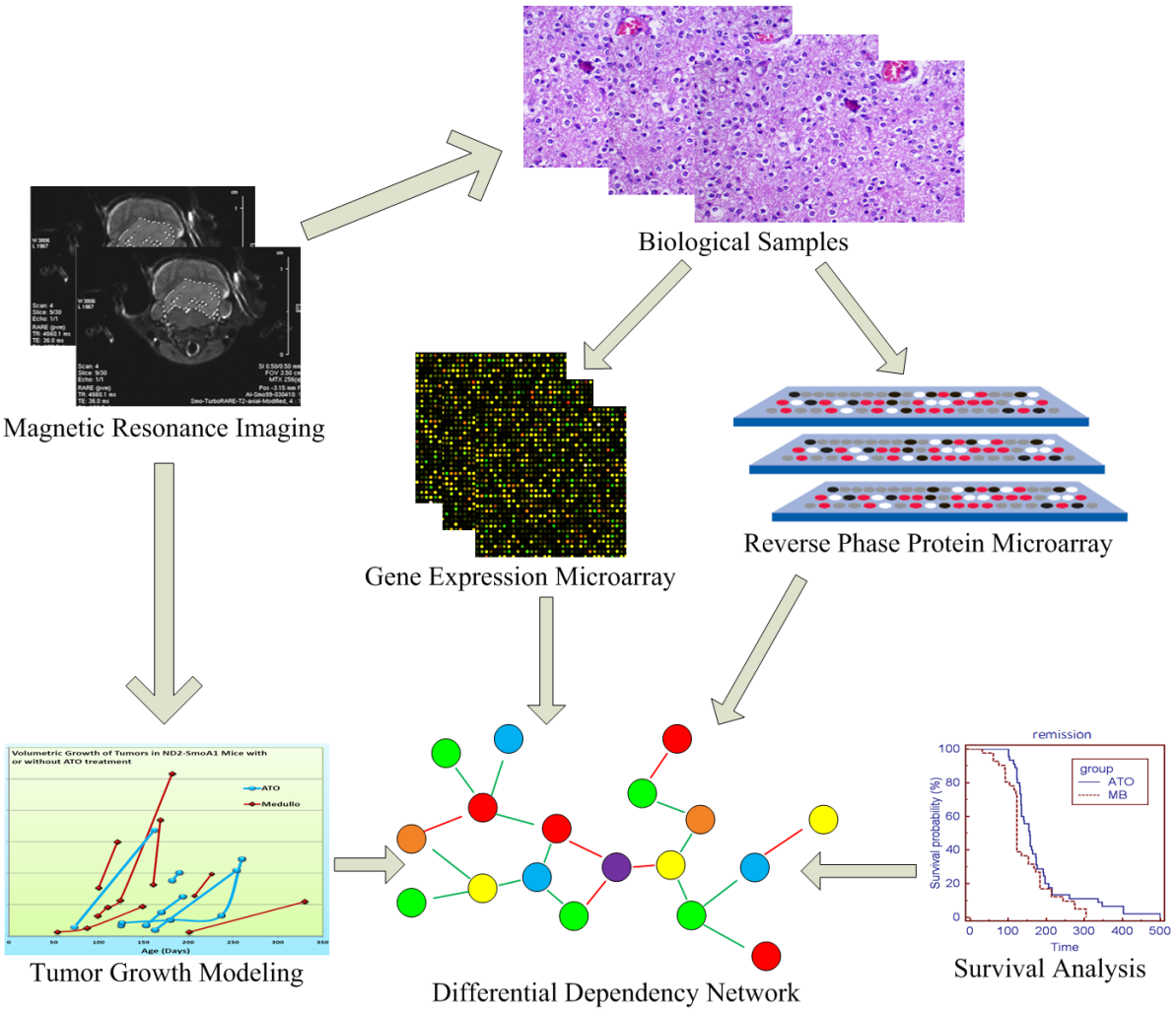


Figure 3.9: Overview of the data acquisition and analytics pipeline.

of drug efficacy.

We acquired the new survival data, MRI volumetric scans, and protein expressions from the ND1-SmoA1 MB mice, *in vivo*. This combination of precise survival-imaging based phenotyping with multiplatform molecular genotyping creates an opportunity to mechanistically elucidate the efficacy and mechanism of action by ATO on MB.

MRI based tumor volume estimates are summarized in Table 3.1. The ATO group had a mean growth rate of 0.0116 mm/day with a standard deviation of 0.0047 mm/day; the MB

Table 3.1: Tumor growth rate of treated and untreated groups.

Slope (mm/day)	MB	ATO
	0.003	0.005
	0.018	0.010
	0.129	0.011
	0.004	0.015
	0.028	0.017
	0.035	
Mean	0.036	0.0116
Standard deviation	0.047	0.0047

control group had a mean growth rate of 0.0361 mm/day with a standard deviation of 0.0472 mm/day. While our tumor growth results do not reach statistical significance (P-value of 0.1299 by two-sample unequal-variance t-test), the combination of the survival and MRI data strongly indicated, for the first time, that ATO reduced actual tumor growth rates, resulting in an increased lifespan in ATO-MB mice. Our results also support the utility of an imaging-based strategy for assessing and monitoring tumor response to therapy, potentially valuable in clinical trials of novel therapies using human subjects where imaging signatures as the endpoint provide a non-invasive and continuous monitoring capability (Zhao *et al.*, 2010).

It has been previously reported that *ErbB2*, a protein involved in growth of cells, was down-regulated with ATO treatment (Albanese *et al.*, 2013). Looking more extensively at the top 20 prioritized proteins (Figure 3.10), we examined their up/down-regulation between ATO-treated mice and MB mice without treatment. We subsequently searched relevant literature to infer/confirm the roles of ATO in regulating cell cycle machinery. Our results indicated that ATO down-regulated important anti-apoptotic (or pro-proliferation oncogene) protein expression. Our current analyses now indicate that ATO up-regulated/induced many pro-

apoptosis genes, including *Bad*, *Stat3*, *Smac/Diablo*, *β-catenin*, *Ask1*, and cleaved *Casp3*. Since for all these proteins, $P < 0.05$, our hypothesis that the apoptotic mechanisms are triggered or pro-proliferation mechanism are suppressed by ATO and is statistically significant at $\alpha = 0.05$.

In response to ATO treatment, different regulatory components and mechanisms are activated and the topology of the cell cycle signaling network changes accordingly. Using the RPMA expression data for 154 cell cycle related proteins, we applied KDDN, to detect the topological changes in protein-protein interaction (PPI) networks between MB and ATO. In a biological system or process, many genes work together in cohesion. In addition to ATO's effects directly on relevant proteins, the ATO-induced rewiring of cell cycle signaling pathway network represents another form of molecular therapeutic mechanisms. Our KDDN analysis, for the first time, detected many novel topological changes in biological network between the ATO and MB mice (Figure 3.11). *Cyclin D1* was previously identified being targeted by ATO (Albanese *et al.*, 2013), while *cPLA2* provides intracellular arachidonic acid to supply both cyclooxygenase and lipoxygenase pathways in cell growth and survival. Our DDN analysis result indicated that the connection between *Cyclin D1* and *cPLA2* (marked by arrow in Figure 3.11) existed in MB has been destroyed by ATO treatment. *MEK* and *EGFR* are the two cascaded components of *MAPK/ERK* pathway in regulating transcription and translation. Our result shows that the critical link between *MEK* and *EGFR* (marked by arrow in Figure 3.11) has been eliminated by ATO.

We further manually identified proteins closely involved in MB signaling and learned the KDDN on this focused subset by comparing the networks of normal samples, tumor samples and tumor samples after ATO treatment, the result of which is shown in Figure 3.12. Edges with different colors have unique meanings: blue - fully recovered by treatment (edges destroyed by tumor are regained); green - weakly created connection after treatment; red - not recovered; black: fully broken down tumor connection; gray: partially broken down

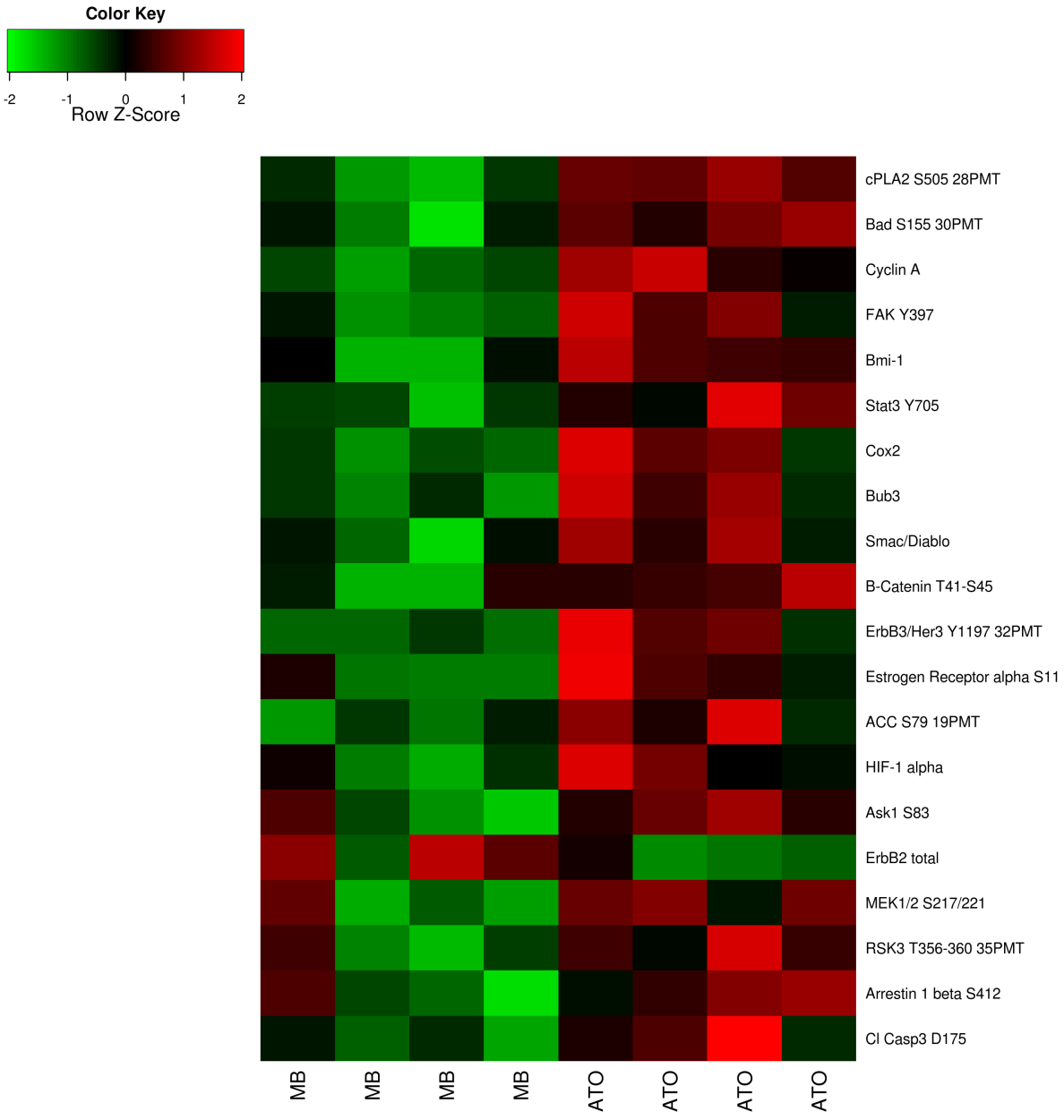


Figure 3.10: Differential analysis of RPMA data highlight the top 20 proteins modulated by ATO *in vivo*.

Edges	Hub protein in DDN
8	cPLA2 S505 28PMT
6	PKCabll T638/641
5	Cyclin B1
5	MEK1/2 S217/221
4	Cyclin D1
4	Bad total 30PMT
4	Bad S112 32PMT
4	ERM TTT 32PMT
4	A-Raf S299 32PMT
4	ErbB3/Her3 Y1197 32PM
4	cKit Y703 35PMT
4	Akt S473
4	4EBP1 T70 35PMT
4	EGFR Y1068 35PMT

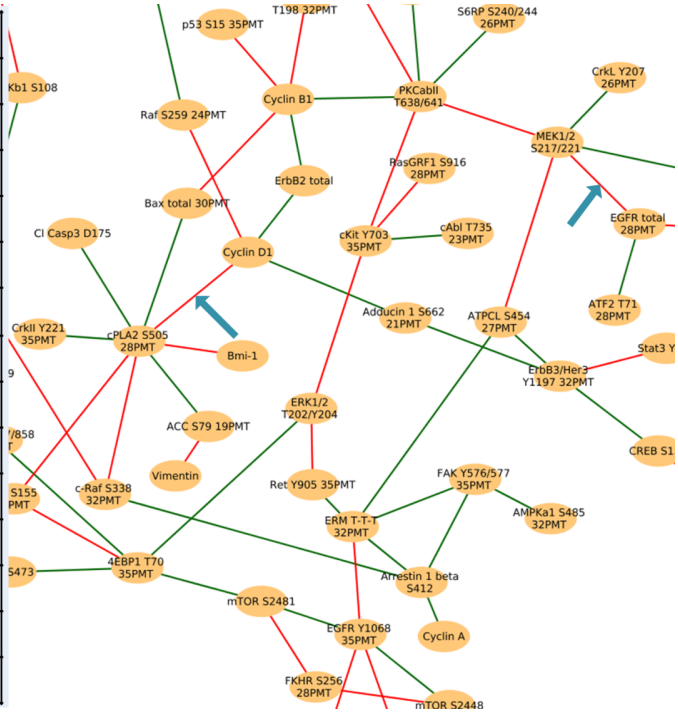


Figure 3.11: "Hubs" of the network rewiring by DDN analysis. Red edges are connections in MB destroyed by ATO treatment, and green edges are connections established after ATO treatment.

tumor connection. Our results may provide new insights into how ATO suppress cancer cell growth at systems level. In this more comprehensive study, KDDN identified a novel ATO driven link between induced *p38MAPK* activity and the spindle check-point protein *BUB3*. In addition, since the link between *p38MAPK* and *HIF1* remain partially active, further insights into the mechanistic insights into ATO anti-tumor activity are illuminated. Importantly, *cPLA2* provides intracellular arachidonic acid to supply both cyclooxygenase and lipoxygenase pathways in cell growth and survival. Our KDDN analysis result indicated that the levels *cPLA2* activity that existed in MB have been significantly affected by ATO treatment, possibly also helping to repress *HIF1* activity in the tumors.

3.8 Conclusions

In this chapter, KDDN is applied in varies case studies and demonstrated the usefulness of network analysis. In yeast stress response study, we found conclusions corroborate with existing literature, partially validated KDDN using real data. In breast cancer case study, KDDN highlights the significant network differences between recurrent patients and recurrence free patients. In T cell study new insights into the muscular dystrophy is gained. In transcription factor study the directional prior knowledge is demonstrated to facilitate the learning. In ovarian cancer case study, the differential networks identified by KDDN give hints on the impact of chromosome instability on the downstream gene network changes. Using gene network comparison to identify the origin of cancer is a unique strategy to formulate the problem and the finding can shed some light on the understanding of this complex mystery. In the imaging-integrated study KDDN is shown as an effective part in a comprehensive systems biology application.

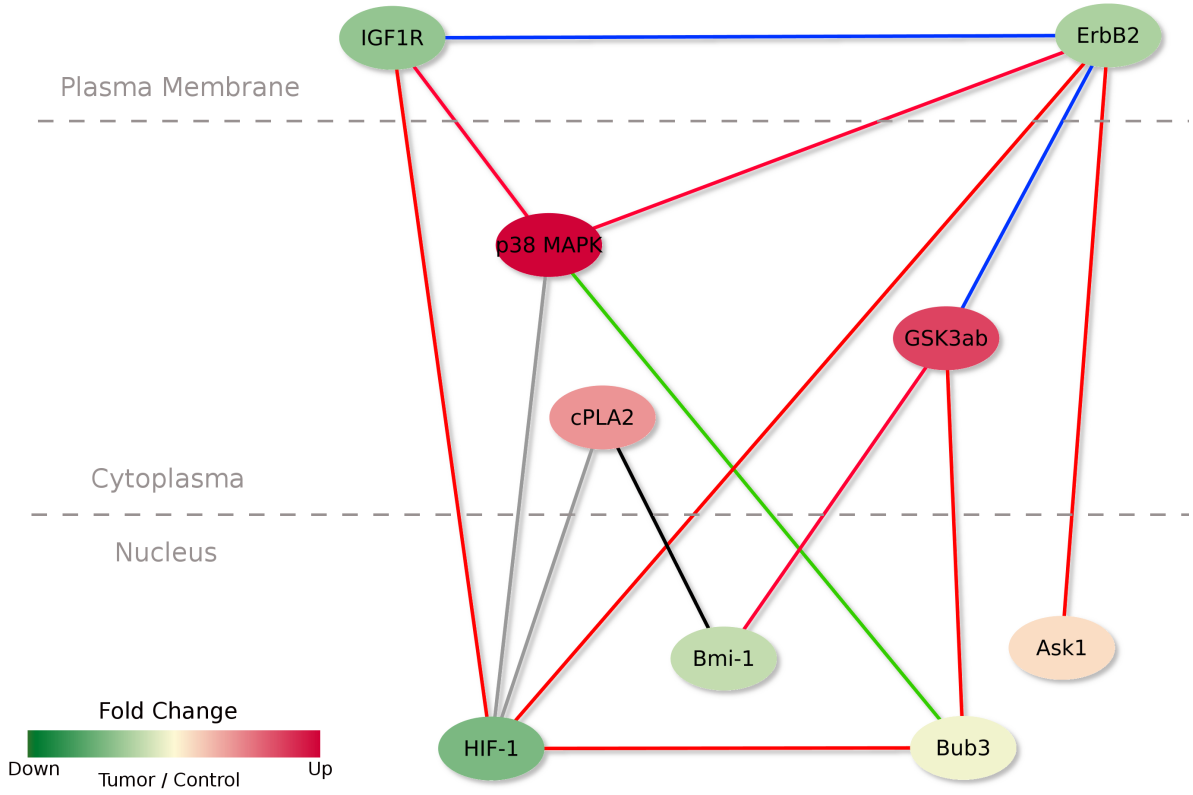


Figure 3.12: A focused network with proteins closely involved in MB signaling provides more insights into how ATO works to treat MB in vivo. Edges with different colors have specific meanings: blue - fully recovered by treatment (edges destroyed by tumor are regained); green - weakly created connection after treatment; red - not recovered; black: fully broken down tumor connection; gray: partially broken down tumor connection.

Chapter 4

Summary of Contributions and Future Work

4.1 Summary of Contributions

In this dissertation, we report a novel network modeling approach, namely knowledge-fused differential dependency network (KDDN), to learn the network structure and topological changes of Gaussian graphical models under different conditions. The KDDN method is statistically principled, and as illustrated by simulation studies and applications to real gene expression data, can be very effective at revealing network structure and rewiring within data. The effective incorporation of relevant prior knowledge is relatively simple to interpret, yet still convey considerable biological information. We have applied the methods to modeling biological networks on various gene/protein expression data sets and have obtained promising results. The contributions of this research work are summarized as follows:

Methodology

1. We propose a novel mathematical formulation that enables quantitative and effective incorporation of prior knowledge into network structural learning of Gaussian graphical models, utilizing the strengths of both condition-specific measurement data and relevant biological prior knowledge.

2. We propose a novel strategy to control the impact of unavoidable false positives in the prior knowledge. The strategy fully exploits the strong evidence from data while obtains "second opinion" by efficient consultations with prior knowledge.

3. We propose a significance assessment scheme to detect statistically significant rewiring of the learned differential dependency network, assign edge-specific p-values, and specify edge types to indicate one of six possible biological scenarios.

Application

1. We develop a comprehensive software suite that provides convenient and unique differential network analysis tool to the research community.

2. We apply KDDN to phenotype-conditioned data groups to study pathway-wide network rewiring, identify differential connectivity that may be responsible for determining different phenotypes at network level, and have obtained biologically plausible preliminary results.

3. We propose a phenotypic-distance over KDDN space between sample groups to study the origin of ovarian cancer. Our result confirms the previously proposed biological hypothesis.

4.2 Future Work

Identification and characterization of network structure and topological changes under different conditions are of great scientific significance in understanding the complex operating mechanisms of biological systems. We thoroughly examined the theoretical and practical aspects of differential network learning with prior knowledge under two conditions, however there are directions worth further exploration.

1. Generalize the two-condition learning to multiple-condition network learning to fully utilize the multi-task learning framework and reveal the network rewiring under this more complex scenario.

2. Study the strategy and methodology to integrate multi-platform biological profiling data and multiple sources of prior knowledge. The heterogeneity in the multiple sources is a major challenge.

3. Develop more accurate summary/test statistics for assessing the statistical significance of rewiring edges.

4. Explore alternative ways of weighted prior knowledge incorporation so as to improve the performance of KDDN.

Bibliography

Ahmed, A. and Xing, E. P. (2009). Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, **106**(29), 11878–11883.

Airoldi, E. M. (2007). Getting started in probabilistic graphical models. *PLoS Comput Biol*, **3**(12), e252.

Albanese, C., Rodriguez, O. C., VanMeter, J., Fricke, S. T., Rood, B. R., Lee, Y., Wang, S. S., Madhavan, S., Gusev, Y., III, E. F. P., and Wang, Y. (2013). Preclinical magnetic resonance imaging and systems biology in cancer research: Current applications and challenges. *The American Journal of Pathology*, **182**(2), 312 – 318.

Bakay, M., Wang, Z., Melcon, G., Schiltz, L., Xuan, J., Zhao, P., Sartorelli, V., Seo, J., Pegoraro, E., Angelini, C., Shneiderman, B., Escolar, D., Chen, Y.-W., Winokur, S. T., Pachman, L. M., Fan, C., Mandler, R., Nevo, Y., Gordon, E., Zhu, Y., Dong, Y., Wang, Y., and Hoffman, E. P. (April 2006). Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of rbmyod pathways in muscle regeneration. *Brain*, **129**(4), 996–1013.

Bandyopadhyay, S., Mehta, M., Kuo, D., Sung, M.-K., Chuang, R., Jaehnig, E. J., Bodenmiller, B., Licon, K., Copeland, W., Shales, M., Fiedler, D., Dutkowski, J., Gunol, A., van

Attikum, H., Shokat, K. M., Kolodner, R. D., Huh, W.-K., Aebersold, R., Keogh, M.-C., Krogan, N. J., and Ideker, T. (2010). Rewiring of genetic networks in response to dna damage. *Science*, **330**(6009), 1385–1389.

Banerjee, O., Ghaoui, L. E., D'Aspremont, A., and Natsoulis, G. (2006). Convex optimization techniques for fitting sparse Gaussian graphical models. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 89–96, New York, NY, USA. ACM.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, **9**, 485–516.

Barabasi, A.-L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, **12**(1), 56–68.

Bell, G., Hey, T., and Szalay, A. (2009). Beyond the data deluge. *Science*, **323**(5919), 1297–1298.

Berman, D. M., Karhadkar, S. S., Hallahan, A. R., Pritchard, J. I., Eberhart, C. G., Watkins, D. N., Chen, J. K., Cooper, M. K., Taipale, J., Olson, J. M., and Beachy, P. A. (2002). Medulloblastoma growth inhibition by hedgehog pathway blockade. *Science*, **297**(5586), 1559–1561.

Beyer, A., Bandyopadhyay, S., and Ideker, T. (2007). Integrating physical and genetic maps: from genomes to interaction networks. *Nature Reviews Genetics*, **8**(9), 699–710.

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition.

Bowen, N., Walker, L. D., Matyunina, L., Logani, S., Totten, K., Benigno, B., and McDonald, J. (2009). Gene expression profiling supports the hypothesis that human ovarian surface

epithelia are multipotent and capable of serving as ovarian cancer initiating cells. *BMC Medical Genomics*, **2**(1), 71.

Causton, H. C., Ren, B., Koh, S. S., Harbison, C. T., Kanin, E., Jennings, E. G., Lee, T. I., True, H. L., Lander, E. S., and Young, R. A. (2001). Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, **12**(2), 323–337.

Chen, L., Yu, G., Langefeld, C., Miller, D., Guy, R., Raghuram, J., Yuan, X., Herrington, D., and Wang, Y. (2011). Comparative analysis of methods for detecting interacting loci. *BMC Genomics*, **12**(1), 344.

Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., MacNeil, D. J., Zhang, C., Lamb, J., Edwards, S., Sieberts, S. K., Leonardson, A., Castellini, L. W., Wang, S., Champy, M.-F., Zhang, B., Emilsson, V., Doss, S., Ghazalpour, A., Horvath, S., Drake, T. A., Lusis, A. J., and Schadt, E. E. (2008). Variations in DNA elucidate molecular networks that cause disease. *Nature*, **452**(7186), 429–435.

Cherry, J. M., Hong, E. L., Amundsen, C., Balakrishnan, R., Binkley, G., Chan, E. T., Christie, K. R., Costanzo, M. C., Dwight, S. S., Engel, S. R., Fisk, D. G., Hirschman, J. E., Hitz, B. C., Karra, K., Krieger, C. J., Miyasato, S. R., Nash, R. S., Park, J., Skrzypek, M. S., Simison, M., Weng, S., and Wong, E. D. (2012). Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, **40**(D1), D700–D705.

Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., and Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nat Rev Cancer*, **8**(1), 37–49.

Cline, M., Smoot, M., Cerami, E., Kuchinsky, A., Landys, N., Workman, C., Christmas, R., Avila-Campilo, I., Creech, M., Gross, B., Hanspers, K., Isserlin, R., Kelley, R., Killcoyne,

S., Lotia, S., Maere, S., Morris, J., Ono, K., Pavlovic, V., Pico, A., Vailaya, A., Wang, P., Adler, A., Conklin, B., Hood, L., Kuiper, M., Sander, C., Schmulevich, I., Schwikowski, B., Warner, G., Ideker, T., and Bader, G. (2007). Integration of biological networks and gene expression data using cytoscape. *Nature Protocols*, **2**(10), 2366–2382.

Consortium, A. I. M. (2011). Evidence for network evolution in an arabidopsis interactome map. *Science*, **333**(6042), 601–607.

Costa, V. M. V., Amorim, M. A., and Quintanilha, A. (2002). Hydrogen peroxide-induced carbonylation of key metabolic enzymes in Saccharomyces cerevisiae: the involvement of the oxidative stress response regulators Yap1 and Skn7. *Free Radical Biology & Medicine*, **33**(11), 1507–1515.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition*. Wiley-Interscience, 2 edition.

de Bont, J. M., Packer, R. J., Michiels, E. M., Boer, M. L. d., and Pieters, R. (2008). Biological background of pediatric medulloblastoma and ependymoma: A review from a translational research perspective. *Neuro-Oncology*, **10**(6), 1040–1060.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, **28**(1), pp. 157–175.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, **1**(2), 302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**(3), 432–441.

Friedman, N. (2004). Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, **303**(5659), 799–805.

Friedman, N., Linial, M., and Nachman, I. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**, 601–620.

Gao, S. and Wang, X. (2011). Quantitative utilization of prior biological knowledge in the bayesian network modeling of gene expression data. *BMC Bioinformatics*, **12**(1), 359.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, **11**(12), 4241–4257.

Genz, A. and Bretz, F. (2009). *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., and Hothorn, T. (2012). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-9994.

Grant, C. M., Perrone, G., and Dawes, I. W. (1998). Glutathione and catalase provide overlapping defenses for protection against hydrogen peroxide in the yeastsaccharomyces cerevisiae. *Biochemical and Biophysical Research Communications*, **253**(3), 893 – 898.

Hara, S. and Washio, T. (2011). Common substructure learning of multiple graphical gaussian models. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part II*, ECML PKDD'11, pages 1–16, Berlin, Heidelberg. Springer-Verlag.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2001). Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, **1**, 49–75.

Hernndez-Ochoa, E. O., Robison, P., Contreras, M., Shen, T., Zhao, Z., and Schneider, M. F. (2012). Elevated extracellular glucose and uncontrolled type 1 diabetes enhance nfat5 sig-

naling and disrupt the transverse tubular network in mouse skeletal muscle. *Experimental Biology and Medicine*, **237**(9), 1068–1083.

Hood, L., Heath, J. R., Phelps, M. E., and Lin, B. (2004). Systems biology and new technologies enable predictive and preventative medicine. *Science*, **306**(5696), 640–643.

Hudson, N. J., Reverter, A., and Dalrymple, B. P. (2009). A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation. *PLoS Comput Biol*, **5**(5), e1000382.

Ideker, T. and Krogan, N. J. (2012). Differential network biology. *Molecular Systems Biology*, **8**(1).

Ikner, A. and Shiozaki, K. (2005). Yeast signaling pathways in the oxidative stress response. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, **569**(12), 13 – 27. Stress Responses.

Jamieson, D. J. (1998). Oxidative stress responses of the yeast saccharomyces cerevisiae. *Yeast*, **14**(16), 1511–1527.

Jensen, S. T., Chen, G., Stoeckert, C. J., and Jr (2007). Bayesian variable selection and data integration for biological regulatory networks. *The Annals of Applied Statistics*, **1**(2).

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, **28**(1), 27–30.

Klinge, C. M. (2001). Estrogen receptor interaction with estrogen response elements. *Nucleic Acids Research*, **29**(14), 2905–2919.

Kolar, M., Song, L., and Xing, E. P. (2009). Sparsistent learning of varying-coefficient models with structural changes. *Advances in Neural Information Processing Systems*, **22**(11), 1–9.

Kolar, M., Song, L., Ahmed, A., and Xing, E. P. (2010). Estimating time-varying networks. *Annals of Applied Statistics*, **4**(1), 94–123.

Kuge, S., Jones, N., and Nomoto, A. (1997). Regulation of yap1 nuclear localization in response to oxidative stress. *The EMBO Journal*, **16**(7), 1710–1720.

Kurman, R. J. and Shih, I.-M. (2009). Pathogenesis of Ovarian Cancer: Lessons From Morphology and Molecular Biology and Their Clinical Implications. *International Journal of Gynecologic Pathology*, **27**(2), 151–160.

Kushner, P. J., Agard, D. A., Greene, G. L., Scanlan, T. S., Shiau, A. K., Uht, R. M., and Webb, P. (2000). Estrogen receptor pathways to ap-1. *The Journal of Steroid Biochemistry and Molecular Biology*, **74**(5), 311 – 317.

Lee, J., Godon, C., Lagniel, G., Spector, D., Garin, J., Labarre, J., and Toledano, M. B. (1999). Yap1 and skn7 control two specialized oxidative stress response regulons in yeast. *Journal of Biological Chemistry*, **274**(23), 16040–16046.

Lee, M. E., Singh, K., Snider, J., Shenoy, A., Paumi, C. M., Stagljar, I., and Park, H.-O. (2011). The rho1 gtpase acts together with a vacuolar glutathione s-conjugate transporter to protect yeast cells from oxidative stress. *Genetics*, **188**(4), 859–870.

Li, H., Xuan, J., Wang, Y., and Zhan, M. (2008). Inferring regulatory networks. *Frontiers in bioscience : a journal and virtual library*, **13**, 263–275.

Lin, C.-Y., Strom, A., Vega, V., Li Kong, S., Li Yeo, A., Thomsen, J., Chan, W., Doray, B., Bangarusamy, D., Ramasamy, A., Vergara, L., Tang, S., Chong, A., Bajic, V., Miller, L., Gustafsson, J.-A., and Liu, E. (2004). Discovery of estrogen receptor alpha target genes and response elements in breast tumor cells. *Genome Biology*, **5**(9), R66.

Loi, S., Haibe-Kains, B., Desmedt, C., Lallemand, F., Tutt, A. M., Gillet, C., Ellis, P., Harris, A., Bergh, J., Foekens, J. A., Klijn, J. G., Larsimont, D., Buyse, M., Bontempi, G.,

Delorenzi, M., Piccart, M. J., and Sotiriou, C. (April 1, 2007). Definition of clinically distinct molecular subtypes in estrogen receptorpositive breast carcinomas through genomic grade. *Journal of Clinical Oncology*, **25**(10), 1239–1246.

Lotia, S., Montojo, J., Dong, Y., Bader, G. D., and Pico, A. R. (2013). Cytoscape app store. *Bioinformatics*, **29**(10), 1350–1351.

Luscombe, N. M., Madan Babu, M., Yu, H., Snyder, M., Teichmann, S. A., and Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, **431**(7006), 308–312.

Marquez, R. T., Baggerly, K. A., Patterson, A. P., Liu, J., Broaddus, R., Frumovitz, M., Atkinson, E. N., Smith, D. I., Hartmann, L., Fishman, D., Berchuck, A., Whitaker, R., Gershenson, D. M., Mills, G. B., Bast, R. C., and Lu, K. H. (2005). Patterns of Gene Expression in Different Histotypes of Epithelial Ovarian Cancer Correlate with Those in Normal Fallopian Tube, Endometrium, and Colon. *Clinical Cancer Research*, **11**(17), 6116–6126.

Mazumder, R. and Hastie, T. (2012). Exact covariance thresholding into connected components for large-scale graphical lasso. *J. Mach. Learn. Res.*, **13**, 781–794.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, **34**(3), 1436–1462.

Mitra, K., Carvunis, A.-R., Ramesh, S. K., and Ideker, T. (2013). Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, **14**(10), 719–732.

Murphy, K. M., Ranganathan, V., Farnsworth, M. L., and Lock, R. B. (2000). Bcl-2 inhibits bax translocation from cytosol to mitochondria during drug-induced apoptosis of human tumor cells. *Cell Death and Differentiation*, **7**(1), 102–111.

Nowak, M. A., Komarova, N. L., Sengupta, A., Jallepalli, P. V., Shih, I.-M., Vogelstein, B., and Lengauer, C. (2002). The role of chromosomal instability in tumor initiation. *Proceedings of the National Academy of Sciences*, **99**(25), 16226–16231.

Ochs, M. F. (2010). Knowledge-based data analysis comes of age. *Brief Bioinform*, **11**(1), 30–39.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, **104**(486), 735–746.

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D. L., and Falciani, F. (2004). Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, **20**(9), 1361–1372.

Reverter, A., Hudson, N. J., Nagaraj, S. H., Prez-Enciso, M., and Dalrymple, B. P. (2010). Regulatory impact factors: unraveling the transcriptional regulation of complex traits from expression data. *Bioinformatics*, **26**(7), 896–904.

Ricke, R. M., van Ree, J. H., and van Deursen, J. M. (2008). Whole chromosome instability and cancer: a complex relationship. *Trends in Genetics*, **24**(9), 457 – 466.

Roy, S., Werner-Washburne, M., and Lane, T. (2011). A multiple network learning approach to capture system-wide condition-specific responses. *Bioinformatics*, **27**(13), 1832–1838.

Shen, C., Huang, Y., Liu, Y., Wang, G., Zhao, Y., Wang, Z., Teng, M., Wang, Y., Flockhart, D., Skaar, T., Yan, P., Nephew, K., Huang, T., and Li, L. (2011). A modulated empirical bayes model for identifying topological and temporal estrogen receptor alpha regulatory networks in breast cancer. *BMC Systems Biology*, **5**(1), 67.

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic boolean

networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**(2), 261–274.

Song, L., Kolar, M., and Xing, E. (2009). Time-Varying Dynamic Bayesian Networks. *Advances in Neural Information Processing Systems 22*, **22**, 1732–1740.

Staleva, L., Hall, A., and Orlow, S. J. (2004). Oxidative stress activates fus1 and rlm1 transcription in the yeast saccharomyces cerevisiae in an oxidant-dependent manner. *Molecular Biology of the Cell*, **15**(12), 5574–5582.

Stender, J. D., Kim, K., Charn, T. H., Komm, B., Chang, K. C. N., Kraus, W. L., Benner, C., Glass, C. K., and Katzenellenbogen, B. S. (2010). Genome-wide analysis of estrogen receptor alpha dna binding and tethering mechanisms identifies runx1 as a novel tethering factor in receptor-mediated transcriptional activation. *Molecular and Cellular Biology*, **30**(16), 3943–3955.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, **109**(3), 475–494.

Umayahara, Y., Kawamori, R., Watada, H., Imano, E., Iwama, N., Morishima, T., Yamasaki, Y., Kajimoto, Y., and Kamada, T. (1994). Estrogen regulation of the insulin-like growth factor i gene transcription involves an ap-1 enhancer. *Journal of Biological Chemistry*, **269**(23), 16433–42.

Varadhachary, G. R. and Raber, M. N. (2009). Gene Expression Profiling in Cancers of Unknown Primary. *Journal of Clinical Oncology*, **27**(25), e85–e86.

Werhli, A. and Husmeier, D. (2007). Reconstructing Gene Regulatory Networks with

Bayesian Networks by Combining Expression Data with Multiple Sources of Prior Knowledge. *Statistical Applications in Genetics and Molecular Biology*, **6**(1), 612–633.

Yuan, M. and Lin, Y. (2007a). Model selection and estimation in the gaussian graphical model. *Biometrika*, **94**(1), 19–35.

Yuan, M. and Lin, Y. (2007b). On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **69**(2), 143–161.

Yuan, Y., Rueda, O. M., Curtis, C., and Markowetz, F. (2011). Penalized regression elucidates aberration hotspots mediating subtype-specific transcriptional responses in breast cancer. *Bioinformatics*.

Zhang, B. and Wang, Y. (2010). Learning structural changes of gaussian graphical models in controlled experiments. In *Conference on Uncertainty in Artificial Intelligence (UAI 2010)*.

Zhang, B., Li, H., Riggins, R. B., Zhan, M., Xuan, J., Zhang, Z., Hoffman, E. P., Clarke, R., and Wang, Y. (2009). Differential dependency network analysis to identify condition-specific topological changes in biological networks. *Bioinformatics*, **25**(4), 526–532.

Zhang, B., Tian, Y., Jin, L., Li, H., Shih, I.-M., Madhavan, S., Clarke, R., Hoffman, E. P., Xuan, J., Hilakivi-Clarke, L., and Wang, Y. (2011). Ddn: a cabig analytical tool for differential network analysis. *Bioinformatics*, **27**(7), 1036–1038.

Zhang, Z., Ferraris, J. D., Brooks, H. L., Brisc, I., and Burg, M. B. (2003). Expression of osmotic stress-related genes in tissues of normal and hyposmotic rats. *American Journal of Physiology - Renal Physiology*, **285**(4), F688–F693.

Zhao, B., Oxnard, G. R., Moskowitz, C. S., Kris, M. G., Pao, W., Guo, P., Rusch, V. M., Ladanyi, M., Rizvi, N. A., and Schwartz, L. H. (2010). A pilot study of volume measure-

ment as a method of tumor response evaluation to aid biomarker development. *Clinical Cancer Research*, **16**(18), 4647–4653.

Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *J. Mach. Learn. Res.*, **7**, 2541–2563.

Zhou, S., Rütimann, P., Xu, M., and Bühlmann, P. (2011). High-dimensional covariance estimation based on gaussian graphical models. *J. Mach. Learn. Res.*, **12**, 2975–3026.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, **101**(476), 1418–1429.

Appendix A

Appendix

A.1 Biographical Sketch

Ye Tian received his BE and MS degrees in the Department of Automation at Tsinghua University, Beijing, China, in 2007 and 2009, respectively. Since August 2009, he has been a doctoral student and graduate research assistant in the Bradley Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University (Virginia Tech), Virginia, United States of America, under the supervision of Dr. Yue Wang. His research interests include machine learning and its applications to bioinformatics and computational biology.

A.2 List of Publications Related to the Dissertation

[1] **Y. Tian**, B. Zhang, E.P. Hoffman, R. Clarke, Z. Zhang, I.M. Shih, J. Xuan, D.M. Herrington and Y. Wang, "KDDN: An open-source Cytoscape app for constructing differential dependency networks with significant rewiring", *Bioinformatics*, 2014. (in revision)

[2] **Y. Tian**, B. Zhang, E.P. Hoffman, R. Clarke, Z. Zhang, I.M. Shih, J. Xuan, D.M. Herrington and Y. Wang, "Knowledge-fused differential dependency network models for detecting significant rewiring in biological networks", *BMC Systems Biology*, 8:87, 2014.

[3] **Y. Tian**, S. Wang, Z. Zhang, O. Rodriguez, E. Petricoin, I.M. Shih, D. Chan, L. Hilakivi-Clarke, M. Avantaggiati, G. Yu, S. Ye, R. Clarke, C. Wang, B. Zhang, and C. Albanese, "Integration of network biology and imaging to study cancer phenotypes and responses", *IEEE-ACM Trans Computational Biology and Bioinformatics*, 2014. (in press)

[4] B. Zhang, **Y. Tian**, and Z. Zhang, "Network Biology in Medicine and Beyond", *Circulation: Cardiovascular Genetics*, 2014. (in press)

[5] **Y. Tian**, L. Chen, B. Zhang, G. Yu, I-M Shih, J. Xuan, and Y. Wang, "Genomic and network analysis to study origin of ovarian cancer", *Systems Biomedicine*, 1:55-64, 2013.

[6] L. Chen, **Y. Tian**, S. Bova, I.M. Shih, and Y. Wang, "Discriminant and network analysis to study origin of cancer", *Statistical Diagnostics of Cancer: Analyzing High-Dimensional Data*, Wiley-VCH Verlag GmbH & Co. KGaA, 2013.

[7] S. Assis, A. Warri, M.I. Cruz, O. Laja, **Y. Tian**, B. Zhang, Y. Wang, T.H. Huang, and L. Hilakivi-Clarke, "Maternal dietary exposure to synthetic estrogen during pregnancy leads to transgenerational increase in mammary cancer risk", *Nature Communications*, 3:1053, 2012.

[8] S. Wang, O. Rodriguez, **Y. Tian**, S. Ye, E. Petricoin, and C. Albanese, "Integration of Imaging and Systems Biology to Study Treatment of Medulloblastoma", *Proc. of IEEE International Workshop on Genomic Signal Processing and Statistics*, 22-25, Washington, DC, 2012.

[9] **Y. Tian**, B. Zhang, I.M. Shih, and Y. Wang, "Knowledge-Guided Differential Dependency Network Learning for Detecting Structural Changes in Biological Networks", *Proc. of ACM International Conference on Bioinformatics and Computational Biology*, 254-263,

Chicago, 2011.

[10] B. Zhang, **Y. Tian**, L. Jin, H. Li, I.M. Shih, S. Madhavan, R. Clarke, E.P. Hoffman, J. Xuan, L. Hilakivi-Clarke, and Y. Wang, "DDN: A caBIG analytical tool for differential network analysis", *Bioinformatics*, 27(7):1036-1038, 2011.