Inferring Regulatory Networks from Multiple Sources of Genomic Data

by

Chen-Hsiang Yeang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2004

Author .....
Department of Electrical Engineering and Computer Science
August 30, 2004

Certified by .....
Tommi S. Jaakkola
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Inferring Regulatory Networks from Multiple Sources of Genomic Data

by

Chen-Hsiang Yeang

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses the problems of modeling the gene regulatory system from multiple sources of large-scale datasets. In the first part, we develop a computational framework of building and validating simple, mechanistic models of gene regulation from multiple sources of data. These models, which we call *physical network models*, annotate the network of molecular interactions with several types of attributes (variables). We associate model attributes with physical interaction and knock-out gene expression data according to the confidence measures of data and the hypothesis that gene regulation is achieved via molecular interaction cascades. By applying standard model inference algorithms, we are able to obtain the configurations of model attributes which optimally fit the data. Because existing datasets do not provide sufficient constraints to the models, there are many optimal configurations which fit the data equally well. In the second part, we develop an information theoretic score to measure the expected capacity of new knock-out experiments in terms of reducing the model uncertainty. We collaborate with biologists to perform suggested knock-out experiments and analyze the data. The results indicate that we can reduce model uncertainty by incorporating new data. The first two parts focus on the regulatory effects along single pathways. In the third part, we consider the combinatorial effects of multiple transcription factors on transcription control. We simplify the problem by characterizing a combinatorial function of multiple regulators in terms of the properties of single regulators: the function of a regulator and its direction of effectiveness. With this characterization, we develop an incremental algorithm to identify the regulatory models from protein-DNA binding and gene expression data. These models to a large extent agree with the knowledge of gene regulation pertaining to the corresponding regulators. The three works in this thesis provide a framework of modeling gene regulatory networks.

Thesis Supervisor: Tommi S. Jaakkola
Title: Associate Professor

Acknowledgments

There are many persons to thank for their kind help in fulfilling this dissertation. The first person is my advisor, Professor Tommi Jaakkola. He brought me into the field of computational biology that I was eager to work with but did not have chance to. All the works appeared in this dissertation are the results of our intensive discussions during the past five years. I appreciate his great insight about problems and the remarkable knowledge in machine learning and mathematics. I also respect his insistence on the preciseness and the rigor about research works as well as scientific writings. Moreover, I believe all his students including myself benefit from his dedication to help students and the tremendous freedom allowed in shaping our research agenda. It is my great pleasure to work with him.

I should also thank Professor David Gifford at MIT CSAIL. He builds an intellectual alliance between biologists at Whitehead Institute and computer scientists at MIT CSAIL. Thanks to his effort, people in these very distinct disciplines start to communicate their views, appreciate each other's contributions, share certain perspectives and interests and collaborate in some pioneering works in computational biology. I benefit greatly from interacting with people in this unique environment. He also provided constructive advice about interfacing my works in biological contexts.

I also want to thank Professor Richard Young at Whitehead Institute as an encouraging mentor to facilitate building this inter-disciplinary and interactive environment. Being a biologist, he understands the importance of computational modeling in this field and is willing to collaborate with computer scientists. He granted us the privilege of using various pioneering high-throughput data generated from his laboratory before they were published. He also helped me to pin-point biological problems which are important in this field and relevant for computational works.

I want to thank Professor Trey Ideker at UCSD for the close collaboration with us. We share the same perspective of building mechanistic models based on molecular interactions. Therefore, the works of physical network models can be attributed to our fruitful discussions when he was a postdoctoral fellow at Whitehead Institute.

Furthermore, he was willing to dedicate substantial amount of resource and time to perform knock-out experiments in order to validate the automated experimental design methods. I appreciate this collaboration because I believe it is a right scenario of doing research in this field. In addition, he also pointed out the sources of high-throughput protein-protein interaction database to be incorporated in the physical network models.

I would also like to thank members of Ideker Laboratory at Whitehead and at UCSD: Owen Ozier, Scott McCuine, Chris Workman and Ying-Ja Chen. They provided the visualization software (Cytoscape) which draws all network figures in this thesis and generated new knock-out data following our suggestions. The experimental design part of this dissertation (Chapter Five) is indeed a joint work with them.

I shall thank members of Young Laboratory at Whitehead for the kind share of their data, valuable discussions about our works and instructions about the basic biology of the problems I work on. Nicola Rinaldi provided high-throughput CHIP-chip data and gene expression data. She also answered my questions regarding the experimental procedures of CHIP-chip assays and yeast transcription regulatory systems. Tony Lee offered insights about mechanisms of combinatorial control for multiple regulators and helped me to clarify certain assumptions about this problem. Julia Zeitlinger kindly reviewed our work of automated experimental design before its submission. She also pointed out references regarding the biological knowledge about yeast mating pathways. I thank other members of Young Laboratory for the share of their research works during the joint meeting sessions. I also thank Ernst Fraenkel and Ben Gordon at Whitehead for our communications with each other's research works and the kind share of the binding motif data generated by them.

I would like to thank former and current members of Professor Gifford's group at MIT CSAIL for the valuable comments and suggestions about my research works. Ziv Bar-Joseph and Georg Gerber discussed with me about the technical aspects of physical network models and the possibility of extending the current models into several different directions. Their work of the GRAM algorithm also inspired the idea of physical network models. Karen Sachs offered help in explaining the technologies

of various high-throughput assays and their limitations. She also provided comments about the presentations of our works, pointed out references in the thesis, and proof-read my thesis chapters. Ken Takusagawa and Tim Danford helped me to get high-throughput data of binding motifs. Alex Rolfe shared the discussions about inferring combinatorial effects from binding and expression data and the error models of CHIP-chip assays. Alex Hartemink's works of Bayesian network models and the discussions with him helped me to shape the framework of physical network models. In addition to personal help, I also appreciate the valuable discussions of relevant works in this field during the weekly group meetings with these people.

I appreciate the dynamic interactions with my colleagues of Tommi's group at CSAIL during the past five years. John Barnett shared with me substantial common interests in computational biology and machine learning. We had frequent discussions regarding the topics directly related to our own works and beyond. He also cordially proof-read my dissertation chapters. Jason Rennie had shared the same office with me for nearly five years. Aside from the communications about our works and discussions about machine learning problems, he also provided great help in sorting out various technical issues of networks and computers and proof-read my thesis chapters. Martin Szummer reviewed my thesis proposal and provided suggestions about research methodology and writings. We also had collaboration beyond this thesis. Harald Steck offered great help in technical problems of Bayesian networks. We also communicated intensively about inferring gene regulatory networks due to the common interests. In addition, I should thank other former and current members of Tommi's group: Tony Jebara, Adrian Corduneanu, Nathan Srebro, Claire Monteleoni, and Romer Rosalez. I benefit greatly from the discussions with them in the group meetings of discussing machine learning literature. Many of them also sat in my preparation talks and offered useful suggestions about the presentations of our works.

I would like to thank other members of MIT CSAIL community for their help during my study. Professor Tomas Lozano-Perez graciously agreed to be in my thesis committee. He helped me to clarify some questions in the original models and refine

our current works. He also provided useful guidance for interfacing the disciplines of computer science and biology. Huizhen Yu offered help in suggesting the improvement of my presentations and discussing relevant problems in machine learning. Sayan Mukherjee provided useful information and suggestions about postdoctoral jobs. I should also thank my officemates at G585 in the Stata Center: Meg Aycinena, Natalia Hernandez-Gardiol, Kurt Steinkraus, Nick Matsakis, James McLurkin, Sarah Finney, Terry Koo, Michael Ross, Luke Zettlemoyer. They were responsible for reconfiguring the space to be more amenable than the original setting.

I would like to thank DARPA and NIH for partially funding the research projects in my thesis.

I shall thank many friends whom I have met in Boston during these years. They make this period of time a unique experience in my life.

I want to thank my brother Chen-Pang Yeang and my sister-in-law Wen-Ching Sung. We live together under the same roof, and they are always considerate and close to me. It is them that make me feel Boston is home.

Finally, I would like to dedicate this dissertation to my parents, Mr. Chia-Chu Yeang and Mrs. Lan-Chun Hsu. Their dedication to children's education shapes my life and values, and their selfless support allows me to explore possible directions of my career. I cannot finish this dissertation without their effort.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prelude

Life has been the inspiration for engineering and information science since the dawn of human civilization. The earliest record (or more likely, the earliest mythology) of humanoid robots dates back to 500 B.C. in ancient China ([116]). The flying machine appeared in Leonardo DaVinci's manuscript ([38]) was a remarkable imitation of bird wings. Computers, whose constitutional materials – beads, gears, wheels, vacuum tubes, silicon chips – cannot be further from biological life, have a very biological meaning in the etymological sense. The earliest use of this word denotes the clerks who did computational works for military purposes ([18]). This word is also semantically translated into Chinese as "electrical brain", which bears the connotation of a living creature. This translation, although imprecise, partly reflects the subconsciousness of pioneers of information and computer science during the early to mid twentieth century. Alan Turing not only laid out the theoretical foundation of computations and computability, but also coined the Turing test that could operationally define artificial intelligence, and proposed the earliest quantitative model elucidating embryo development – diffusion-reaction models. Norbert Wiener studied human brain waves and established cybernetics which applied control theory to explain the dynamic behavior of biological systems. Frank Rosenblatt instilled the abstract notion of neural information processing into a computational model – perceptrons –

which eventually evolved into extremely fruitful applications ranging from consumer appliances to spam email filters. These big names are just few in the large echelon of scientists/engineers who have spent substantial efforts bridging computational and biological science. A wide range of sub-disciplines – including computer vision, robotics, brain and cognitive science, computational linguistics, machine learning, biomechanics, medical imaging, computational biology, and many others – have been developed toward this direction since the beginning of the computer era.

However, it is not until the end of the twentieth century and the beginning of the twenty first century that biological science per se is perceived as an information science. Beforehand *bench work* comprised the whole life of biologists who work at laboratories. As experimental technologies improve and more high-throughput technologies are developed, the time and effort spent on data collection are reduced while the analysis, extraction and processing information become more important parts of biological research. It is not a wild conjecture that in the future experimental work will be completely automated in a large scale. The main job of biologists will be designing experiments, analyzing and processing the large amount of information gathered from different sources. Information and engineering science are expected to play important roles in the development of this "new" biology due to their expertise in the processing of large amount of information.

This trend already becomes prominent in the fields of many "omics" (genomics, proteomics, interactomics, etc.) and systems biology. A large amount of data covering different aspects of the biological system have been collected: DNA sequences, structures of proteins or other molecules, mRNA and protein expressions, molecular interactions, protein modifications and localizations, metabolic substrates fluxes, and many others. The need to extract meaningful information from these data creates many computational problems. The infrastructure of information storage, processing and transfer is certainly an important aspect; for instance, the standardization and management of biological databases, the platform of streamlining the data collection processes, visualization and representation of information. However, more important aspects are the quantitative techniques of studying a complex biological system:

modeling or simulating a complex system, extracting statistically significant patterns from data, and so on. These problems create tremendous opportunities for computer scientists to contribute in biological science.

This thesis presents one of the many works that attempt to understand gene regulation by building computational models on large-scale genomic data. Due to the complexity of the gene regulation system and insufficient data, current progress in this field are still preliminary. We view the works in this thesis as an effort of tackling an important biological problem with a principled computational method.

1.2 Gene regulation

The goal of this thesis is to develop computational methods of inferring aspects of gene regulation from large-scale genomic datasets. In this section I will give a crude overview about the biological processes of gene regulation. This overview is incomplete and general. It serves the purpose of providing the biological background for the discussions in subsequent chapters. Most of the content in this overview is excerpted from [101].

It is now a commonly known that proteins are fundamental building blocks of life and genes are the DNA segments encoding proteins. Proteins are important for life because they participate in diverse biological processes. Few instances include muscles and inter-cellular matrix, hemoglobins, enzymes, antibodies, transcription factors, and signal transducers. The aggregate of these biological processes determines the structures, morphologies and functions of organisms and their fitness in a specific environment.

The information about proteins is encoded in DNAs. DNA (Deoxynuleic Acid) is a long, double-helix shaped polymer composed of nucleotides with base pairs of purines and pyrimidines. Like bits for digital computers, nucleotides are the basic information units for a DNA. Each position along a DNA is filled with one of the four bases: adenine (A), thymine (T), cytocine (C) and guanine (G). The bases along the two strands of a DNA need to match each other in order to form a stable double-

helix structure: A matches T and C matches G. The composition of bases from the 5-end to the 3-end of a DNA is called the sequence of this DNA. Because DNAs are duplicated whenever cells divide via mitosis and all the cells of the same organism originate from a single cell (zygote), the DNA sequences of different cells are almost identical (with certain exceptions, for instance, the DNA sequence of a cancer cell undergoes a series of mutations thus is significantly different from a normal cell). The entire DNA sequence of an individual organism is called the *genome* of this organism. The size of a genome varies from a few kilo bases to several giga bases.

The synthesis of protein products from DNA sequences is called *gene expression*. The synthesis procedure of most contemporary organisms follows the *central dogma*: DNA source $\rightarrow$ RNA template $\rightarrow$ protein product. The sequence information in a DNA (A, T, C, G) is first *transcribed* into another type of polymer called messenger Ribonucleic Acid (mRNA). mRNA is much smaller yet less stable than DNA. The base thiamine (T) in a DNA is replaced by uracil (U) in an mRNA. After transcription mRNAs in most eukaryotes are *spliced* by removing the sequences which do not encode proteins (introns) and ligating together the separated sequences encoding the same protein (exons). Spliced mRNA molecules are transported from nucleus to cytoplasm. A spliced mRNA is then *translated* into a protein by ribosome and transfer RNAs (tRNAs). A triplet of bases in the spliced mRNA (a codon) corresponds to a specific amino acid. The tRNAs carrying the RNA bases complementary to mRNA codons (anti-codons) are then sequentially recruited and the chain of amino acids (poly-peptide chain) is elongated. The poly-peptide chain is released from ribosomes upon completion and is folded into the proper protein conformation. The information flow DNA source $\rightarrow$ RNA template $\rightarrow$ protein product is not universal for all organisms. Retroviruses, for instance, store genetic information in RNAs and reverse-transcribe the genetic information into the DNA sequence of the host. The reversely transcribed genes in the host DNAs are then expressed along with other host genes.

The protein synthesis mechanisms described above have been studied in a great detail during the past fifty years. These mechanisms, although essential for understanding how genes control the biological processes, are not sufficient. They are

common to all genes hence do not elucidate why the functions of some proteins are manifested in specific cell types under certain conditions. In order to understand how genes function under a specific internal and external condition, it is necessary to know the fundamental processes of modulating the functions of genes. The modulation of gene functions is called gene regulation.

The functions of many genes are modulated by the quantities of their products. Obviously the number of protein molecules in a cell can affect the activities of this protein. In analogy to computers, the activity of a gene is at "ON" state when its protein level is high and at "OFF" state when the protein level is low. In practice many genes modulate their functions through this simple mechanism. For instance, the quantities of enzymes Gal2p and Gal3p catalyzing galactose metabolism significantly increase in a galactose-rich environment ([103]); heat shock proteins Hsp30p remain low at room temperature but are expressed when the temperature rises ([137]). Since genes are expressed through transcription and translation, the control of expression levels operates on these processes.

The quantities of most gene products are modulated via the control of transcription initiation. The transcription of genes is undertaken by a multi-unit complex – RNA polymerase II holoenzyme – constituted of about ten proteins surrounding the RNA polymerase II (RNAP II) ([77]). In eukaryotes, the RNAP II holoenzyme binds to the TATA-box upstream of the transcription start site of DNA, reads its sequence, synthesizes and elongates the mRNA molecule by sliding along the DNA. The RNA polymerase II holoenzyme is a general and insufficient apparatus for transcription initiation. To initiate transcription the RNAP II holoenzyme must interact with the proteins which bind on the DNA regions upstream of the transcription start site – the promoter regions of genes. Those DNA-binding proteins are gene-specific transcription factors. The modulation of mRNA quantities of genes under a specific condition can be achieved by the bindings of gene-specific transcription factors. A few instances of transcription initiation control have already been studied in a great detail. For example, genes encoding galactose metabolism enzymes are activated when transcription factor Gal4 binds to their promoters. Under glucose-rich or

galactose-poor conditions, repressor Gal80 binds to Gal4 and inhibits its interaction with the general transcription apparatus, hence represses the expression of galactose metabolism genes. When galactose is enriched, Gal80 is disassociated from Gal4 and those genes are activated ([103, 68]).

Although transcription factor bindings are widely conceived as a major mechanism of modulating transcription initiation, there are many unresolved problems. An important problem is how the expression of the entire genome is controlled by a relatively small number of transcription factors. Because most genes are bound by multiple transcription factors on their promoters, coordinated and competitive interactions among transcription factors are believed to be responsible for the diverse control (for instance, [29, 125]). However, except a few cases (e.g., the interaction of Gal4 and Gal80) most combinatorial control mechanisms are unknown. The specificity of transcription factor bindings on promoters is under intensive study. The specificity of some transcription factors are achieved by the DNA sequence alone, for the promoters bound by those factors are enriched with sequences of specific patterns called motifs. However, the binding sites of many transcription factors do not seem to have simple patterns. It is also unclear how important the non-transcription factor proteins which indirectly bind to DNAs – chromatin modifying factors, MAP kinases, and so on – are in terms of regulating transcription initiation.

The latency of transcription initiation control ranges from minutes (for example, cell cycles of protozoan) to hours and days (for example, embryo development of metazoan). It may not be fast enough to respond to some very dynamic environmental changes. In order to respond to those changes cells have developed other mechanisms of modulating protein functions. Protein modification is another common way to alter protein functions. A protein often contains multiple "docking sites" which can accommodate small molecular groups. It is chemically modified when the molecular groups are recruited to or disassociated from the docking sites. The activity of a protein may depend on its modification state, for the presence of small molecular groups may alter the conformation of the protein or provide energy for its activities. Phosphorylation, for example, activates the protein by adding the energy in the phosphodiester bond

of phosphate into the protein ([101]). Other post-translational modifications include methylation, acetylation, ubiquitination, and adenization ([101]).

Protein modifications play important roles in certain biological processes such as signal transduction pathways ([89, 101]). Because the transcription initiation control takes place in the nucleus, there must be a mechanism of propagating an external stimulus into the nucleus. This mechanism is called signal transduction. The signal transduction goes through a series of amplification stages like analog electronic circuits. The external stimulus often changes the conformation of receptor proteins embedded on the cellular membrane (for instance, the binding of antigens on the surface of T-cells or the binding or pheromones on the pheromone receptors). This change in turn modifies the protein at the next stage (for instance, G-protein). The signal of the external stimulus is then propagated along the pathway. The protein at the preceding step modifies the protein at the subsequent step. Signals are amplified since one protein at the preceding step is capable of modifying many proteins at the subsequent step. Signal proteins are transported into the nucleus and modify transcription factors. Gene expression is affected in response to the modification of transcription factors. A well-studied example of protein modification driven signal transduction is Mitogen Activation Phosphorylation (MAP) kinase pathway ([133]). The transduction of signals is achieved via a cascade of phosphorylations. This mechanism is responsible for the mating signal transduction of yeasts, and we will discuss it in more detail in Chapter Four.

Although the functional roles of protein modifications are as prevalent as transcription initiation control, they are not as intensively studied. The primary reason is the difficulty of studying protein modifications in a high-throughput fashion. High-throughput assays of detecting chemical modification states of proteins are actively under development. High-resolution mass spectrometry is a promising technology, but it is not yet able to efficiently measure the quantities and modification states of all proteins in the proteome. On the other hand, the mRNA or protein levels alone are not reliable indicators of the activities of protein modifications. Some studies showed that the expression levels of MAP kinases along the same signal transduc-

tion pathway were correlated (perhaps due to the feedback transcription control from the end products of signal transduction, [136]). However, this observation does not hold in general. Protein-protein interactions are more reasonable indicators for they are the necessary conditions for certain types of protein modifications such as phosphorylations. However, most large-scale assays of protein-protein interactions do not capture transient interactions ([132]), and the quality of these large-scale datasets is often questioned ([31, 86]). Therefore, studying the gene regulatory effects of protein modifications remains an open problem.

Transcription initiation and post translational modifications do not cover all mechanisms of gene regulation. In addition to post translational modifications, cells also respond to abrupt environmental changes by localization. Some transcription factors are occluded from the nucleus under normal conditions and are transported into the nucleus under the environmental change. Gene expression can also be regulated at translation levels in addition to transcription levels.

The mechanisms of gene regulation establish relations among all the genes which can be viewed as a complex network. Understanding and reconstructing gene regulatory networks are one of the leading problems in contemporary biology ([25]).

1.3 Problem statement

The goal of this thesis is to study computational methods of reconstructing the gene regulatory network from multiple sources of (primarily high-throughput) data. Due to the availability of data and simplicity of models, we focus on the gene regulation mechanisms which can be revealed by physical (molecular) interactions – protein-DNA and protein-protein bindings. These mechanisms cover the control of transcription initiation and signal transduction pathways.

We postulate that the effects of gene regulation are realized via pathways of molecular interactions. For example, the effect of deleting a gene can be propagated along a specific pathway and eventually alters a downstream gene. Therefore, the network of physical interactions is essential for understanding gene regulation. However, the

physical interaction network alone does not provide sufficient information about gene regulation. To understand gene regulation we need to annotate the physical network with various properties pertaining to the functions of gene regulation: the function of a physical interaction as activation or repression, the direction of a protein-protein interaction in a signal transduction cascade, the activity of a pathway, and so on. These attributes are important since a specific setting of their values provides a simple yet self-consistent model about gene regulation. For example, if the direction and function of each interaction along the pathway is known, then we can predict the effect of deleting each gene along the pathway.

The first part of this thesis focuses on learning the annotations of the physical network from multiple sources of data. Some annotated properties (such as the presence of a physical interaction) are directly observed via noisy measurements. However, most properties are indirectly constrained from multiple data rather than directly observed. For example, the expression response in a gene deletion experiment informs us about the aggregate effect along the pathways connecting deleted and affected genes, but does not reveal the effects of individual interactions. The computational model needs to incorporate both direct observations and indirect constraints under the same framework. Once we can express both direct observations and indirect constraints within the same modeling framework, we want to find the algorithm which efficiently infers the annotated properties that satisfy the constraints from data.

In addition to model formulation and inference algorithms, there are several other sub-problems for annotating the physical network models. First, due to the sparse constraints from existing knock-out data, there are often an astronomical number of annotations which fit the data equally well. We want to efficiently represent these annotations, for example, decompose a configuration into the product of subconfigurations of independent subnetworks. Second, once optimal annotations are inferred from existing datasets, we want to systematically validate inferred results. This includes quantitative tests such as cross validation tests on the predictive accuracy of the inferred models and qualitative tests such as literature survey on inferred subnetworks.

Due to the sparse constraints from existing data, there are likely many optimal annotations which fit the data equally well. Further experiments are required in order to discriminate the true annotation from other candidates. The second part of this thesis focuses on automated experimental design and analyzing the data from new experiments to reduce the uncertainty of inferred models. The purpose of experimental design is to suggest new experiments which would provide the maximal expected information for discriminating existing models. Various computational problems are linked to experimental design for physical network models: how to choose the loss (objective) function for prioritizing new experiments, how to incorporate the property of physical network models in the loss function, how to simplify the computation in order to make it tractable, and so on.

The effectiveness of an experimental design scheme can be internally validated by methods such as cross validation or learning curve analysis. However, the ultimate test is to perform the suggested experiments and analyze the new data. We collaborate with biologists to perform gene knock-out experiments according to our experimental design criteria and analyze the new data. The purpose of data analysis is two fold: to verify the consistency of model predictions regarding gene expression changes and to reduce the uncertainty of annotations along these pathways. A number of computational problems emerge when achieving these goals in data analysis. We will discuss these problems in Chapter Five.

The physical network models described in the first part only consider the effects of single gene deletions on downstream genes. This approach can only extract the gene regulation properties under the scenario when multiple transcription factors independently control downstream genes. However, in many cases multiple transcription factors control gene regulation in a coordinated fashion. Therefore, it is essential to study how multiple transcription factors coordinately control gene regulation. This is a challenging problem since there are many possible mechanisms involved in coordinated gene regulation. For example, multiple proteins form a complex, or the presence of one protein may block the recruitment of another protein on the same promoter. For simplicity we only consider the functional aspect of regulation from

multiple transcription factors. This means modeling the dependencies between the observed quantities pertaining to gene regulation. The third part of the thesis focuses on modeling the mRNA expression data between transcription factors and their regulated genes.

The problem of modeling the dependencies of gene expression data for multiple transcription factor control is still complicated due to the combinatorial nature of the functions. Consider a very simple scenario that mRNA expression data are quantized to two levels (on or off) and the mRNA levels of regulated genes are deterministic functions of regulators (transcription factors). With respect to the input size, there are an exponential number of possible input configurations and a super-exponential number of possible Boolean functions. The large size of the functional class makes the inferred functions highly susceptible to over-fitting data of a limited size. Moreover, most of these combinatorial functions are also hard to interpret in terms of simple and fundamental mechanisms. Therefore, we want to simplify the combinatorial functions to a reduced class such that they can be efficiently enumerated but still capture essential properties of gene regulatory control.

In addition to simplifying the class of combinatorial functions, it is also important to reduce possible assignments between regulators and regulated genes. In principle, it is possible that each gene is controlled by a large number of transcription factors with a distinct function. This scenario, however, is very unlikely due to the physical limitation of proteins and DNAs and the lack of economy which is against evolution. Instead, many biologists postulate that genes of similar biological functions are often co-regulated by a small number of transcription factors. The combinatorial function from the expression states of regulators to the expression state of each gene is identical. The set of regulators and regulated genes that are tied together with a specific function constitute the basic unit of gene regulation. Our goal is to infer these basic units – or *regulatory modules* – from protein-DNA binding and mRNA expression data. In other words, we want to identify pairs of regulator sets and gene sets and the combinatorial functions which optimally fit binding and expression data. Once these modules are identified, we also want to validate the results by applying external

information such as the biological functions of regulated genes and the regulatory functions of regulators.

1.4 Overview of previous works

There has been a rich literature of computational methods of inferring gene regulatory networks during the past few years. Most early works rely on mRNA expression data alone. As more types of high-throughput data become available, most recent works incorporate multiple data sources in their models. In this section I will give an overview of previous works which are relevant to this thesis. This overview is divided into three parts. The first part introduces different approaches of modeling gene expression data, including some of the works that address the combinatorial aspects of multiple transcription factor control. The second part discusses efforts of integrating expression data with other types of data sources. The third part covers works about experimental design which are beyond modeling gene regulation but are relevant to the experimental design framework in Chapter Five.

1.4.1 Modeling gene expression data

Classical methods of studying the gene regulatory circuitry rely on detailed investigations on a small system through deletions of genes or cis-regulatory elements. The works by Davidson et al. on sea urchins ([175, 29]) present a remarkable example. In the earliest work, they focused on the regulation of a single gene Endo16 responsible for endoderm development. The promoter region of Endo16 was subdivided into 7 cis-regulatory modules (modules A to G). Each module contains several transcription factor binding sites. Because the transcription factors bound on Endo16 promoter were not completely identified, they studied the single and combinatorial functions of the cis-regulatory elements by deleting single or double modules from Endo16 promoter. The effects of module deletions were observed by measuring the time-course activities of CAT reporter genes. The computational model reported in their work is a logical circuit-like model containing switches, multipliers and basic Boolean op-

erators. Although the resulting model is very accurate, reliable and conceptually clear, this approach is very difficult to extend into the genome-wide scale. A complete deciphering of the regulatory circuitry of one gene requires recognition of all cis-regulatory elements on its promoter, mutations on all cis-regulatory elements and many combinations of cis-regulatory elements, and time-course measurements of the target gene under these conditions. Therefore, it would be too costly to apply the same approach to the entire genome.

Another approach of studying gene regulatory networks is to construct detailed models about the physical/chemical properties of gene expression. Each step of gene expression – transcription initiation, elongation and termination, mRNA degradation, translation initiation, elongation and termination, proteolysis, etc. – can be viewed as a chemical reaction. The static properties (for example, concentrations of reactants at equilibrium) and dynamic properties (for example, the rate of mRNA or protein synthesis) can be modeled using statistical mechanics or molecular dynamics. This approach achieves partial success in small subsystems involved with few genes. Chen et al. constructed linear differential equation models of transcription and translation ([20]):

$$\begin{aligned} d\mathbf{r}/dt &= C\mathbf{p} - V\mathbf{r} + \mathbf{s}, \\ d\mathbf{p}/dt &= L\mathbf{r} - U\mathbf{p}. \end{aligned} \tag{1.1}$$

where $\mathbf{r}$ denotes mRNA levels of all genes and $\mathbf{p}$ protein levels of all genes. Thattai et al. modeled noisy processes of transcription and translation with stochastic differential equations ([154]). A detailed and sophisticated computational model was proposed by Arkin et al. to characterize the bifurcated fate of λ phage ([6]). λ phage can either parasite on the host bacteria genome and remain dormant (lytic phase) or re-program the host cell to massively reproduce its genome and kill the host (lysogenic phase). The state of the virus is indirectly affected by environmental conditions and directly determined by few proteins (Cro, CrI, CrII, etc.). These proteins regulate each other and form a double feedback loop. Arkin et al. constructed detailed models at every step of gene regulation. They applied discrete stochastic processes to model the highly fluctuating processes involved with small molecular quantities. According to this

sophisticated model, they simulated the dynamics of those proteins and demonstrated their consistency with empirical measurements.

These bottom-up approaches from fundamental physical/chemical laws are perhaps the most principled methods of modeling gene regulation. However, their use is limited from both learning and modeling perspectives. From the learning perspective, the bottom-up models often contain a large number of unknown parameters which need to be estimated from a limited dataset. Examples stated above show that even modeling a very small subsystem involved with few (less than five) genes requires hundreds of parameters about the reaction coefficients at every step. Because biochemical assays for measuring these parameters are very expensive and time-consuming, the use of bottom-up models is restricted to either very well characterized systems (such as the λ phage) or very small systems (such as the expression of one gene). The accurate values of these parameters often cannot be reliably learned due to the overfitting problem. Instead, most studies emphasized the qualitative properties emerged from the quantitative models. For instance, the work by Arkin et al. focused on the initial condition (relative concentrations of proteins Cro and CrII) which would lead to different fates of λ phage – lytic or lysogenic phase. However, from the modeling perspective, the qualitative properties may be very sensitive to specific values of parameters. It is well known that a small system of non-linear differential equations can yield very complex behaviors, such as exponential growth or decay, regular or irregular oscillations, or chaos. The "territories" in the space of parameters which would yield different behaviors can be intermingled, and their boundaries are often irregular. Therefore, we may not be able to predict the qualitative behavior of a system without knowing the accurate parameter values.

The bottom-up models are simplified in order to tackle the genome-wide data. For example, the rate equation specifying transcription and translation can be simplified as a set of linear differential equations.

$$\frac{dx_i(t)}{dt} = -r_i x_i(t) + \sum_{j \neq i} a_{ij} x_j(t). \tag{1.2}$$

where r_i denotes the rate of mRNA degradation of gene i and a_{ij} denotes the catalytic efficiency of transcription factor j on gene i. At steady state the left hand side becomes 0, and the rate equations become a system of linear equations. Gardner et al. applied gene expression data to to reverse engineer the rate coefficients and verified the models by the new expression data with specified perturbation on input genes ([60]). Cheng et al. clustered gene expression profiles according to the steady state responses in equations 1.1([21]). Although these methods significantly simplify the bottom-up models, their results are also less reliable due to the simplified assumptions. For example, the mRNA levels of regulated genes are often not a linear function of the mRNA levels of regulators.

Bottom-up models of gene regulation require detailed assumptions about the underlying mechanisms, thus are difficult to construct from available data. In contrast, there are statistical models which require very few hypothesis about the underlying mechanisms. Clustering gene expression data is an extreme example of the data-driven, mechanism-free modeling. The fundamental concept underlying clustering gene expression data is very simple: genes involved in the same regulatory process (or other biological processes) respond to some environmental signals in a similar (or opposite) fashion. Therefore, the expression profiles of these genes are correlated. In practice there are many problems associated with clustering data, and a whole branch of machine learning is dedicated to solving these problems. We will not review all previous works of clustering gene expression data but only discuss some important variants.

The most commonly used clustering methods on gene expression data are hierarchical clustering ([45]), k-means ([143]) and self-organizing maps (SOM, [148]). Hierarchical clustering merges two data points at each iteration and replaces the data of the merged cluster with the average expression profiles of its members. The merging continues until all data points are in the same cluster. The result is a hierarchy of clustering instead of one specific clustering. K-means clustering starts with a fixed number of clusters and iteratively updates the centers of clusters and the cluster memberships of data points. The iteration continues until a stationary

solution is attained. Self-organizing map places clusters on a topological structure (for instance, grids) and updates cluster centers and data point memberships that respect the topological structure. Other clustering methods which have been applied on gene expression data include the maximum likelihood membership allocation of parametric models ([76]) and graph theoretic based clustering methods ([14]).

The most important element of distance-based clustering methods is the choice of the distance or similarity metric. It is natural to treat a gene expression profile as a data point in a high dimensional Euclidean space (the dimension is the number of experiments) and the distance metric as the Euclidean distance between two vectors. Many works of clustering gene expression data use the Euclidean distance (for instance, [148]). In addition, Pearson correlation coefficients are also commonly used as the similarity metric (for example, [45]). It is straightforward to show that the Pearson correlation coefficients are equivalent to Euclidean distances when the expression data are normalized and centered. Other distance or similarity metrics applied in clustering include mutual information ([140]), Markov random walk distances ([147]), and various types of kernels.

The assumption that genes participate in the same biological process are correlated across many different conditions is too simple and requires refinement. This is because the correlation is manifested only under the conditions that perturbed the underlying process. By relaxing the correlation assumption to a subset of experiments, clustering is extended to bicluster both genes and experiments concurrently. Biclustering allows us to extract correlated genes and the experimental conditions on which they are correlated. There have been works of biclustering genes and experiments based on graph theoretic criteria ([21], [151]) or geometric criteria ([80]).

Clustering gene expression data is a summarization of data rather than a mechanistic model about gene regulation processes. In contrast to the bottom-up models described above, the use of clustering is to explore new data rather than understanding the biological processes. Therefore, it can be exploited as a pre-processing step to facilitate more sophisticated models, but the clustering per se can hardly be viewed as a model of gene regulation processes.

Clustering is unable to express the relations beyond cluster memberships (whether genes belong to the same cluster) and pairwise distances (whether two genes are close or apart). To uncover these relations, various statistical models are applied on gene expression and other types of data. Bayesian networks are a class of models which can capture statistical dependencies and independencies beyond pairwise relations. In the Bayesian network formulation, the joint probability of a number of random variables is expressed as the product of conditional probabilities.

$$P(x_1, \cdots, x_n) = \prod_{i=1}^{n} P(x_i|Pa_i). \tag{1.3}$$

where Pa_i denotes the parent variables of x_i. In the context of gene expression data, each random variable denotes the expression level of a gene. The formulation in equation 1.3 seems to suggest the Bayesian network encodes the causal relation of variables: Pa_i is the cause of x_i. This semantics is incorrect when inferring the Bayesian network for there are multiple equivalent causal models which yield the same joint probability distribution ([74]).

The joint probability in equation 1.3 can be graphically represented as a directed acyclic graph (DAG): edges are connected from the parents of x_i to x_i. The graph of a Bayesian network is called its *structure*. The most important information contained in a Bayesian network structure is the *conditional independence* relation. Variables x_i and x_j are conditionally independent given variables X_k if the following equality holds:

$$P(x_i, x_j|X_k) = P(x_i|X_k)P(x_j|X_k). \tag{1.4}$$

In other words, the dependency between x_i and x_j is mediated by X_k. This can be interpreted as either X_k are the intermediate causes between x_i and x_j or X_k are the common causes of x_i and x_j.

There are two types of computational problems on Bayesian networks. Inference denotes computing the conditional probabilities of some variables given the evidence of other variables. This task is often performed by recursively propagating the evidence regarding a specific variable to the variables which are directly dependent on

it (its neighbors in the network), until all evidence have been incorporated by each variable. Learning the network structure denotes finding the conditional independence relations which best fit the data. When expressing a probability distribution, a Bayesian network is decomposed into two components. The structure of the network pertains to the factorization of the joint probability function into smaller terms. The factorization structure encodes the conditional independence relations of variables. The parameters of the network specify the exact function of each term. To learn the structure of a Bayesian network we need to define the objective function with respect to the data. Typically this objective function is the likelihood function marginalized over parameter values: ([73, 74]):

$$P(X, G; D) = \int \prod_i \prod_j \prod_k P(x_i = d_{ij} | Pa_i^G = d_{Pa_i k}, \theta_{ijk}) P(\theta|G) P(G) d\theta. \qquad (1.5)$$

where G is the Bayesian network structure, θ its parameters, d_{ij} and $d_{Pa_i k}$ specific values of x_i and Pa_i. $P(\theta|G)$ and $P(G)$ are the priors of model parameters and structure. The marginal likelihood function is less susceptible to overfitting compared to the joint likelihood function since the effect of a particular parameter setting is smoothed out by averaging. For discrete random variables the joint probability is expressed as the multinomial distribution where θ are the multinomial probabilities, and the parameter prior follows the Dirichlet distribution ([74]). $P(G)$ is often chosen to penalize the complexity of the model, i.e., the number of edges in the graph. Finding the model structure which maximizes equation 1.5 is known to be NP-hard ([22]). Henceforth various heuristics are applied to find suboptimal solutions ([74, 106, 56]).

Bayesian networks are advantageous over clustering in their capability to reveal causal or functional relations from the conditional independence relations. One example was proposed by Hartemink et al. on galactose metabolism genes ([71]). The authors compared the scores of two gene regulation models illustrated. In model 1, the function of repressor Gal80 on galactose metabolism genes (such as Gal2) is mediated by Gal4; in model 2, Gal4 and Gal80 jointly affect Gal2. Model 2 yields a higher

likelihood score, which matches the fact that the complex Gal4-Gal80 regulates the expression of Gal2.

Despite the usefulness in limited examples, it is difficult to trust a single Bayesian network model learned from the data as the underlying model of gene regulation. The problems are due to the quality and size of gene expression data and the reliability of the heuristic search methods. Friedman et al. alleviated these problems by applying the bootstrap method to enlarge the effective size of data and reporting graph theoretic properties extracted from multiple high-scoring models ([56]). The two reported properties (features) are the order of genes and the Markov properties (whether a gene is in the *Markov blanket* – the minimal set of variables that shield the rest of the variables in the model). Hartemink et al. adopted a similar approach by reporting the consensus of the multiple Bayesian network structures of high scores ([72]).

Bayesian networks have been extended along several directions on the analysis of gene expression data. For example, Pe'er et al. applied the method proposed by Cooper et al. ([26]) of combining observational and perturbational data to learn the model structure ([123]). Imoto et al. extended Bayesian network learning to continuous variables by applying splines on the conditional probabilities ([85]). Similarly, Friedman et al. proposed using Gaussian processes to model the functional relations of continuous random variables ([57]). Murphy et al. proposed using dynamic Bayesian networks to study the time course gene expression data ([114]). Steck et al. and Tong et al. proposed different active learning methods on sequentially selecting/suggesting new experiments which could best disambiguate candidate models ([159, 144]).

While Bayesian networks have much stronger expressive power and a clearer objective function than clustering, they suffer from several shortcomings in representing gene regulatory models. First, the graph semantics of Bayesian networks does not necessarily match our intuition about the representation of causal relations. Arrows in a Bayesian network may not denote causal relations, for we can create a class of models with different edge directions but are equivalent in the likelihood function ([73]). It is possible to extract causal relations from conditional independence relations and a substantial number of works were devoted in this field (for example,

[122]). However, these works often required strong assumptions (for example, there were no hidden variables) and could often extract only a small number of causal relations from a large number of dependency relations. Second, because of the convoluted semantics of Bayesian networks and incomplete data, it is often difficult to interpret the learned models in terms of mechanisms. For example, a directed edge in a Bayesian network may contain many intermediate steps of a biological pathway. Third, the quality of learned Bayesian networks depends on the quality and the size of the data. Theoretical and empirical studies indicate that the size of the dataset which guarantee the learning method converges to the true model is exponential in terms of the model size ([37, 149]). This result suggests currently available data can hardly yield any confident Bayesian network model at genomic scale, though it is possible to learn reliable subnetworks from it. Fourth, current methods of learning fully parameterized Bayesian networks emphasize the dependency of variables rather than the functions of them. More often we are interested in the specific functions of genes such as the results in ([175, 29]). To extract the functions in a Bayesian way we have to define the marginal likelihood function pertaining to a specific function pertaining to conditional probabilities. This requires averaging over a restricted class of parameters (for example, all the conditional probabilities which indicate variable A activates variable B) rather than the entire parameter space. Current works of applying Bayesian networks in gene regulation rarely address important problems in this direction, such as how to divide the parameter space appropriately and how to perform integration over the restricted space efficiently.

In order to exploit the functional aspects of gene regulation, another class of models – Boolean networks and their variants – are also widely used in modeling gene regulation. The semantics of a Boolean network is a collection of Boolean functions with possibly cascaded inputs and outputs ([94, 155]). The mappings from input variables to output variables can be expressed as lookup tables of discrete variables. In gene expression analysis, variables of the Boolean networks are quantized gene expression levels. The score of a Boolean network is simply the fitness of the model on the data: the number of model predictions contradicting with the data. One

can also impose a complexity penalty to the score analogous to the score used for Bayesian networks. Boolean networks have the advantage of being able to represent the combinatorial functions of gene regulation in a very simple form. The gene regulation model of cis-regulatory elements of sea urchins ([175, 29]), for example, can be expressed as an extended Boolean network. However, the primary bottleneck is the learning of Boolean network models from data. To uniquely learn a Boolean network all possible input configurations must appear in the data. For a small network with few inputs, input configurations may be manipulated by deleting and over-expressing input genes, for example, [175]. Combinations of perturbations are expensive when applying to a large system. In most cases not only the combinatorial functions but also input variables are unknown. Similar to learning Bayesian networks, it is expensive to directly learn the network structure, and the results are not reliable when the data size is small. There have been efforts of incorporating prior knowledge about network structure. Tanay et al. started with the core networks extracted from literature and expanded the network by incrementally adding input variables and adjusting combinatorial functions which best fit the data ([149]). Many more works attempted to incorporate the information about transcription factor binding motifs and protein-DNA interactions in the model construction. We will introduce some of those works in subsequent paragraphs. Finally, the learned Boolean network structures and functions are subject to noise in the data. Instead of using the deterministic model, some works modeled gene regulation with combinatorial functions plus noise and partially improved the learned models, for instance, [151].

1.4.2 Data integration

Expression data alone do not suffice to provide mechanistic information about gene regulation. As stated above, there are many possible explanations for the correlation of gene expression profiles. To further elucidate gene regulation processes, it is necessary to incorporate other types of data in the models. Since most gene regulation models focus on the transcription initiation aspect, many works of data integration focus on combining gene expression data with the evidence about transcription factor-

promoter bindings. Other types of data – such as protein-protein interactions and metabolic fluxes – are incorporated as more biological processes are taken into account.

We categorize previous works of data integration in terms of the following aspects: the purpose of combining data sources, the relevance to the combinatorial control of multiple regulators, the mechanisms these works are targeted, and the types of data incorporated.

The most straightforward purpose of data integration is to use the information extracted from one data source to verify the model generated by another type of data. Many early works of gene expression analysis fall into this category. These works applied the hypothesis that genes co-expressed under some conditions were co-regulated by the same transcription factor(s). Hence they clustered genes according to expression profiles and sought external evidence that members within the same clusters are co-regulated. For example, Spellman et al. clustered cell cycle gene expression data and verified the known binding motifs (SCB and MCB) appear on the promoters of genes expressed at G1 and S phases ([143]). Tavazoie et al. applied the motif finding algorithm (AlignACE) on the promoters of cluster members and identified more known motifs of transcription factors ([152]). In addition to motif sequences, the functional categorization of genes is often used to verify the clustering results of expression data. For example, [152, 143, 136] all use the Munich Information Center for Protein Sequences (MIPS) functional categories of yeast genes [1] to verify the results.

The use of one data source to validate the models inferred from other sources does not "fuse" data in a strict sense. In contrast, most recent works of data integration extract information jointly from multiple sources. A common approach is to construct the model which jointly fits multiple sources of data. This approach has been realized in several different ways. One can use one type of information as supporting evidence to facilitate the learning from other types of data. Typically the "primary" information is expression data and the supporting information pertains data about sequences, physical interactions or gene functions. For instance, Hartemink et al. im-

[1] http://mips.gsf.de/

posed the protein-DNA interaction data from location analysis as the prior on the structure of the Bayesian network gene expression model ([72]). Graphs containing transcription factor-gene edges extracted from location data were assigned with higher confidence. Segal et al. used information about gene functions to construct decision tree-like modules explaining gene expression data ([136]). Transcription factors and signal transduction proteins are the candidates whose expression profiles could partition the gene expression data under different conditions. One can also construct models pertaining to different data separately and combine them by multiplying their likelihood functions. For instance, Holmes et al. built a joint model of gene expression and promoter sequence to cluster genes ([76]). The expression model is the Gaussian noise model with an uninformative prior. The sequence model is the multinomial distribution with a Dirichlet prior. Tanay et al. combined gene expression data and sequence data to infer the unobserved variables of transcription initiation ([150]). Sequence data was used to model the binding affinity from a transcription factor to a promoter. The activity of a transcription factor on a promoter was dependent on its binding affinity and the mRNA concentration of the transcription factor. The mRNA levels of regulated genes were functions of the transcription factor activities. Segal et al. combined gene expression data, location data and sequence motif information to construct a rich probabilistic model – probabilistic relation model (PRM) ([135]). The model predicted the expression level of a gene under a specific experiment according to the sequence motifs and protein-DNA bindings on its promoter and the information about the experimental types. In addition, one may combine weak evidence from multiple sources to obtain more confident results. For example, Bar-Joseph et al. developed the GRAM algorithm that captured co-expressed genes which are also supported by the location data ([12, 100]). They obtained an initial module of genes by thresholding the confidence values of protein-DNA binding and the correlation coefficients with respect to the average expression profile within the module. The initial module was extended by relaxing the thresholds on location data confidence to incorporate more genes.

Data fusion has been used by several important works to explore the combinato-

rial aspects of gene regulation. The modules inferred from the algorithm in ([136]) implicitly contain information about combinatorial control. The expression levels of regulators in the module affect the expression levels of regulated genes in a combinatorial fashion. For example, regulated genes do not change when regulator A is down-regulated or unchanged. When both regulators A and B are up regulated, module members are down regulated. When A is up regulated and B is down regulated, module members are up regulated. In a decision tree representation A "masks" the influence of B on module members. An interpretation is that A is an activator but is also necessary for the function of repressor B. Another work by Pilpel et al. attempted to identify sequence motif pairs that exhibit stronger regulation effects than single motifs ([125]). The synergistic effect of motif pairs is measured by comparing the coherence of expression data against the hypothesis that each motif independently affects expression coherence. In addition to identifying a set of dependent motif pairs, they also discovered the relative strength of several motifs under some specific experimental conditions.

In terms of the mechanisms, most works of data fusion focus on the transcription initiation aspect of gene regulation. This is reasonable since most data (mRNA expression, sequence motifs, protein-DNA bindings) pertain to this aspect. With the availability of other types of data, some works start to explore the signal transduction or metabolic pathway aspects of gene regulation. Ideker et al. perturbed (deleted) genes along galactose response pathways and reconstructed the partial order of genes along the pathways ([83]). Following this approach, they proposed a statistical score to measure the significance of an *active subnetwork*: a collection of genes connected via molecular interactions that are co-expressed ([82]). They also developed a greedy search algorithm to find active subnetworks. Steffen et al. adopted a similar assumption that genes along the signal pathway were co-expressed and identified several meaningful pathway members ([145]). Ihmels et al. adopted a similar assumption on metabolic pathways and discovered that a significant number of genes along the same metabolic pathways are co-expressed ([84]).

The types of data incorporated depend on the purpose of data integration, the tar-

get mechanisms, and the availability of data sources. Currently most works incorporate combinations of the following types of data: mRNA expression, DNA sequences, various protein-DNA and protein-protein interaction assays, protein expression, and annotations of gene functions. As more types of large-scale data become available, the data integration works are expected to include these new data very soon.

1.4.3 Experimental design

Biological experiments are the ultimate test of all computational models. The choice of experiments, however, is a computational problem. This problem is not intensively addressed in current computational models of gene regulation. We give an overview of the experimental design works relevant to our methods. Most of these works do not necessarily tackle the problems of modeling gene regulation.

Experimental design is an important topic in statistics. It is also termed as active learning in the area of machine learning. Experimental design has been applied to various theoretic and practical problems. Classical examples can be found in the textbook by Fedorov ([48]). It formulates the criteria of choosing experiments (loss functions) for typical statistical problems such as regression, classification, and model discrimination. Different loss functions on different problems are all based on similar principles: to reduce the uncertainty of the learned models based on the hypothetical data generated from new experiments. Because actual data from new experiments are not acquired, we can only compute the expected loss function according to current models. For example, suppose the problem is to estimate the parameter θ in multi-dimensional regression:

$$Y = \theta^T \cdot X + n. \tag{1.6}$$

where X, θ are multi-dimensional vectors of input variables and parameters, and Y, n are scalars of the output variable and noise. We have the freedom of setting the values of X in order to estimate θ. The uncertainty of estimated parameters was characterized by its dispersion matrix $V(\hat{\theta}(D))$ which is simply the covariance matrix of estimated parameters. V is in general a function of new data, thus it cannot

be calculated a priori. Instead we evaluate the expected covariance matrix over the hypothetical data generated by current models. Under the linear form in equation 1.6, V only depends on inputs X:

$$V(\hat{\theta}(D)) = E_D\{\hat{\theta}\hat{\theta}^T\} = \sum_{i=1}^{|D|} X_i X_i^T. \tag{1.7}$$

Thus we can choose the set of inputs X_i which minimize the uncertainty generated from the dispersion matrix, for instance, by minimizing the determinant of V.

Experimental design is naturally applied in machine learning problems where data collection is expensive. Remarkable examples are learning from a vast number of World Wide Web documents. It is easy to automatically collect many web documents but relatively expensive to ask people to manually label the documents or answer the queries about their preferences of documents. Therefore, active learning algorithms are applied to generate queries which are critical to the target problems. Examples include the query-by-committee algorithm ([54]) and active selection of document clusters for information retrieval ([87]). The former method maintains a list of learning "experts" (for example, binary classifiers) and chooses the queries where the prediction results from these experts disagree. The latter method presents clusters of documents which maximize the mutual information between the user's preference about documents and the selection of document clusters.

Active learning for standard machine learning problems such as regression ([48]), model discrimination ([48]), parameter estimation ([158]), and learning the structure of Bayesian networks ([159, 144]) has been intensively studied recently. In learning the Bayesian network structures, an experiment denotes the perturbation of one or multiple variables to fixed values. A key problem is to define the loss function of an experiment. In [159], the uncertainty of model structure is characterized by the uncertainty of edge presence + direction in a graph. The presence + direction of an edge between A and B can be modeled by a three-state random variable (no edge, from A to B, from B to A), and the uncertainty of the graph is the sum of entropies of these variables for each pair. In [144], new experiments were generated from a

committee of experiments which yielded the highest disagreement in terms of the average KL divergence of distributions.

Automated experimental design has recently been applied to biological experiments. A remarkable example is the robot scientist which automated each step in scientific discovery ([96]). The system focused on a small aromatic amino acid metabolic pathway and sought for experiments which would best reveal the relations between the identities of proteins and their enzymatic functions. They looked for auxotrophic experiments which observed cellular phenotypes of a knock-out genotype under different nutritional conditions. The loss function of prioritizing new experiments was based on the monetary cost of experiments and expected reduction of model uncertainty (entropy) according to hypothetical experimental results. In addition to experimental design, they also automated the processes of experimentation. Despite of the very promising outcomes, the system currently works on a very small and well known biological system. How to extend the principled approach of experimental design to a large system with many unknowns and noise would be a primary challenge in this area.

1.5 Roadmap

Following the questions proposed in Section 1.3, we structure the remaining parts of the thesis as follows. Chapter Two lays out the general concepts of the physical network models. It first states the objectives of the modeling framework, then describes each element of the model at a broad level. This includes a skeleton graph of putative physical interactions, attributes (annotated properties) of the model and their configurations, data association and model inference algorithms, experimental design for model discrimination, and combinatorial regulation of multiple transcription factors.

Following the conceptual framework in Chapter Two, Chapter Three discusses the integration of different types of data to a great detail. It starts with introducing the datasets used in this work, then describes the methods of incorporating different types of data as constraints of the model. We will then discuss the model inference algorithm

which efficiently approximates an annotation that satisfies the constraints from data. We will also introduce the algorithm which decomposes optimal annotations into annotations of subnetworks.

Chapter Four focuses on the empirical analysis of the physical network models on several datasets. We first verify the inference results on a relatively small network of yeast mating pathways. Various quantitative and qualitative methods are applied to verify inference results. For instance, cross validation tests on the predictive accuracy of inferred models. We then analyze the inference results on the genome-wide data. Since the system is much larger and less well known, we focus on presenting the inferred subnetworks and linking them to existing knowledge about yeast gene regulation.

Chapter Five discusses experimental design and the analysis of new experimental data. We will first introduce the concept of experimental design for model discrimination, then formulate the objective function for prioritizing new experiments. We then apply the experimental design method to the physical network models obtained in Chapter Four and rank new knock-out experiments accordingly. Suggested experiments are first validated internally by showing their importance in the physical network. We then perform some of the suggested experiments and analyze the new data generated from these experiments. We will show how putative pathways are verified and how the uncertainty of their annotations is reduced by incorporating the new data in the model.

Chapter Six discusses the computational method of inferring regulatory models involved with multiple transcription factors. It first states the problems and the specific assumptions for the computational methods. We then define the elements of a regulatory model and establish a criterion of fitting a regulatory model to protein-DNA binding and gene expression data. We then describe an algorithm that identifies regulatory modules from existing data. It is followed by the analysis and discussions of inferred modules from real datasets.

Chapter Seven draws the conclusion about these works. It also points out limitations of current models and possible extensions for future work.

Chapter 2

Physical Network Models

Modeling the mechanisms and functions of gene regulation is a difficult task due to the complexity of the system and insufficient data. We address these problems by making a simplified assumption that the effect of gene regulation is propagated via cascades of molecular interactions. Accordingly we build a computational model which incorporates evidence from multiple data sources compatible with the underlying hypothesis. This model, which we call the *physical network model*, annotates physical interactions with various attributes and links these attributes with constraints generated from empirical data.

In this chapter, we will describe the framework of physical network models, which is the basis of all the works in this thesis. First we will describe the objectives of our models and discuss the level of detail and the gene regulation mechanisms that the physical network models are aimed to capture. Following this discussion, we will introduce elements of the physical network models, including a skeleton graph composed of pairwise molecular interactions, functional attributes associated with molecular interactions, and the biological interpretations of attribute configurations. Methods of selecting and verifying models – including how to constrain attribute values from empirical data, how to infer attribute values, how to design new experiments to discriminate degenerate models, and how to incorporate combinatorial aspect of transcription control – will be briefly introduced in this chapter and covered in more depth in subsequent chapters.

2.1 Objectives of physical network models

In the realm of gene regulation, a key problem is to identify the regulatory relations of genes and their physical mechanisms. Some mechanisms – such as transcription factor bindings or protein modifications – appear in many gene regulation processes and can be viewed as their building blocks. Using these building blocks to construct the entire network of gene regulation remains an open problem due to the incomplete knowledge in biology and the complexity of the underlying system.

We focus on molecular interactions – protein-DNA and protein-protein bindings – as the primary physical mechanisms of gene regulation. In a brief review in Chapter One, we have seen the important roles of transcription factor bindings and cascades of protein modifications in transcription control. The bindings of transcription factors on DNA promoters serve as a major mechanism of interacting with the RNA polymerase II holoenzyme, which directly controls transcription initiation. On the other hand, an external stimulus is often transduced into the nucleus via a cascade of protein modification events (phosphorylations, methylations, ubiquizations, acetylations, and so on). Hence protein modifications serve as an indirect mechanism of relaying external signals to the transcription apparatus. Many of these modifications are undertaken by protein-protein interactions. For example, phosphorylation, the acceptance of a phosphate group, can take place by interacting with another protein named kinase. Since large-scale protein-protein binding data are currently more accessible than protein modification data, we use protein-protein binding data as a proxy to protein modifications and attempt to infer properties of signal transduction from them. These mechanisms certainly do not cover the entire picture of gene regulation, but they are likely to be the necessary components for controlling gene expressions.

It is important to understand what aspect of gene regulation can be attributed to the effects of molecular interactions and what aspect requires other mechanisms. By knowing the functional roles of molecular interactions, biologists are able to form testable hypothesis and investigate other mechanisms.

We intend to build a computational model to capture the physical interaction aspect of gene regulation. There are several key questions pertaining to this kind of model. First, we want to identify the active pathways and subnetworks which are responsible for transcription regulation. This is important since the datasets pertaining to molecular interactions are noisy and many molecular interactions are not involved in transcription regulation. Second, we want to know the functional attributes associated with individual interactions or pathways. Functional attributes are important because they provide interpretable abstractions to understand the underlying biological processes of gene regulation. These attributes include the presence, the causal directionality (whether one gene affects the other or vice versa) and the functional directionality (whether the upstream gene activates or represses the downstream gene) of an interaction and the activity of a pathway. Third, the inferred attribute values reflect the beliefs about the mechanisms given the current data. Due to insufficient data, there may be multiple putative mechanisms which explain the data equally well. It is also possible that none of the putative mechanisms are indeed accurate. The only way to discriminate or verify these models is to perform new experiments. Therefore, a guiding principle of designing new experiments is critical for the models. Fourth, empirical results indicate the promoters of most eukaryote genes are bound by multiple transcription factors. To understand transcription regulation, it is essential to know the functional roles and combinatorial control schemes.

In order to answer these questions, we develop a computational framework of modeling transcription regulation through molecular interactions. We name this framework a *physical network model* because it is based on a network of physical (protein-DNA and protein-protein) interactions. Various attributes associated with the physical network are defined (they will be described in Section 2.3). These attributes annotate the mechanisms of transcription control via cascades of molecular interactions. These attributes are constrained by various types of data including high-throughput chromatin immunoprecipitation (CHIP-chip) assays for protein-DNA interactions ([100, 78]), yeast two-hybrid systems for protein-protein interactions ([160, 86]), gene expression microarray data ([143, 23, 152, 80, 61]), and po-

tentially many others. We encode these data as probabilistic constraints of model attributes and apply an efficient model inference algorithm to infer the optimal annotations of these attributes. Since the physical network is often sparsely constrained by data, there are likely many attribute annotations which fit the data equally well. We define an information score of new experiments pertaining to their capacity of discriminating these degenerate models. New experiments are performed following the rankings of their information scores. Finally, we incorporate the combinatorial effects of multiple transcription factors in the model, so that it is capable of explaining the expression data beyond the pairwise knock-out interactions.

It is important to realize the limitations of this modeling framework before describing computational algorithms and analyzing data. First, our model focuses on the aspect of molecular interactions. Other mechanisms certainly play important roles and can not be ignored in understanding gene regulation. We have discussed some of these mechanisms in Chapter One, such as chromatin modifications, protein localization, and protein and mRNA degradations. Our model currently does not include these mechanisms. Second, we treat the regulatory network as a circuit of discrete states, hence all the physical (such as molecular bindings) and functional (such as the change of gene expression levels) events are discretized. Quantitative differences of these events – such as the number of transcription factors staying at a promoter region or the number of mRNA copies generated within a time interval – are not considered. Third, we ignore the spatial and temporal aspects of gene regulation. The spatial aspect refers to the localization of gene regulation processes and the variations of gene expression across space. The temporal aspect refers to the temporal (and causal) order of regulatory events and the temporal variations of gene expression. Spatial and temporal effects are the determining factors of development and cell differentiation in multi-cellular organisms. Despite their importance, we ignore these effects for the simplicity of building models and the availability of data. Fourth, we use the error models or error measures provided in the data rather than building refined error models for each dataset. Many previous works are dedicated to this problem, for instance, [42, 44, 79]. We focus on the data integration framework

in this thesis and leave building accurate error models as an external task whose improvement can be "plugged in" the data integration framework.

In the following sections we will introduce each element of the physical network modeling framework. They include a skeleton graph of putative physical interactions, model attributes, data association and model inference, experimental design, and combinatorial regulation of multiple transcription factors. Each element will be discussed in details in the following chapters.

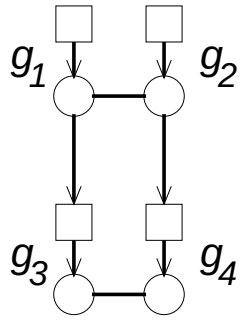
2.2 A skeleton graph of putative physical interactions

Networks (graphs) are the most common metaphor in describing gene regulation. They are widely used for the following reasons. First, networks are easy to visualize and understand compared to texts and equations. Second, networks as a mathematical entity have been studied since Euler, and a wide range of tools are available to study them. Third, graphs have a simple yet versatile representation. Therefore, they are used to express many different relations in different contexts.

The versatility of graphs also creates the possibility of mis-communication if they are not properly defined and illustrated. The networks appeared in the works in this field all have different definitions. For example, an edge in a network can denote a concrete physical interaction ([100, 82]), a statistical dependency ([72, 56, 123, 125]), or a functional relation ([157]). It is thus necessary to clarify the meaning of a network before we use it in the rest of the thesis.

Because we focus on molecular interactions as the physical mechanisms of gene regulation, it is natural to construct our model on top of the network of protein-DNA and protein-protein interactions. We define a *skeleton graph* of physical network models as a collection of "likely" protein-DNA and protein-protein interactions. This network serves as a template or superset for the interactions in the physical network models. Whether a physical interaction is enlisted as a "likely" interaction is judged

Figure 2-1: Toy example of skeleton graph



by direct measurements of protein-DNA or protein-protein interactions. For example, we threshold on the measurement p-values of CHIP-chip data ([100]) and include the protein-DNA interactions which pass the threshold value. Notice each edge in the skeleton graph is not necessarily a true interaction because false positives of measurements are expected. On the other hand, we do not include all possible interactions to eliminate false negatives because of the high cost of encoding this large model and carrying out its inference. Formally, a skeleton graph is defined as:

$$G = (V, E), V = V^d \cup V^p, E = E^{pd} \cup E^{pp}, E^{pd} \subseteq V^p \times V^d, E^{pp} \subseteq V^p \times V^p. \quad (2.1)$$

There are two types of vertices and two types of edges. V contains vertices of proteins V^p and DNA promoters V^d. Hence, a gene is mapped to two vertices in the skeleton graph. E contains edges of protein-DNA interactions E^{pd} and protein-protein interactions E^{pp}. A toy example of a skeleton graph is shown in Figure 2-1. Squares denote DNA promoters and circles denote protein products of genes. An edge between two circles denotes a protein-protein interaction and an edge from a circle to a square denotes a protein-DNA interaction. An edge from the DNA promoter to the protein product of the same gene denotes the functional relation of gene expression (transcription $\rightarrow$ translation).

The directionality of edges needs to be specified. There is no ambiguity in the orientation of an edge of protein-DNA interaction. The direction of a protein-DNA edge is from the protein vertex to the DNA promoter vertex for this is always the

direction of the information flow in gene regulation: a DNA-binding protein controls the transcription initiation of a gene by binding to its promoter region. Notice the source vertices of all protein-DNA interactions belong to the set of DNA-binding proteins (transcription factors) in the genome.

The orientation of an edge of protein-protein interaction, in contrast, is not clear without knowing the function of this interaction. If a series of protein-protein interactions convey signal transduction such as the MAP kinase cascade introduced in Chapter One, then the direction of each protein-protein interaction is clear from the functional perspective: it is from the protein at the previous step of signal transduction (e.g., a MAP kinase kinase) to the protein at the subsequent step (e.g., a MAP kinase). However, although the directionality of protein-protein interactions is clear in this context, the functional direction cannot be determined by the physical interaction data alone. On the other hand, many gene regulation mechanisms do not have a sequential control flow. The direction of a protein-protein interaction thus becomes ambiguous in those schemes. For instance, when two proteins form a dimer in order to bind to a DNA promoter, it may not be clear about the meaning of the functional/causal direction of their protein-protein interaction. Furthermore, many protein-protein interactions are either artifacts or not playing any roles in gene regulation. It is also inappropriate to specify the directions of those edges. For the ambiguities in these aspects, we leave the directions of protein-protein interactions unspecified when constructing a skeleton graph. In subsequent chapters, we will narrow down the definition of protein-protein interaction directions and discuss how to infer them from functional data.

It is straightforward to express pairwise relations in a graph by simply encoding each pairwise relation as an edge. In biology, it is common that multiple genes are involved in the same regulation mechanism. Sometimes these multi-way interactions can be reduced to aggregate effects of pairwise interactions, but more often such reductions are not appropriate. Consider the following two cases: (1) three proteins bind together and form a complex, (2) they bind pairwisely but do not form a complex. Pairwise interactions of these cases are identical, but they correspond to very

different mechanisms. To incorporate multi-way interactions such as complexes, the skeleton graph should become a *hypergraph*: there are *hyper-edges* associated with multiple (instead of two) vertices in the graph. We restrict our discussions to pairwise interactions in this thesis for the simplicity of the model and the lack of high quality high-throughput data revealing complex formations.

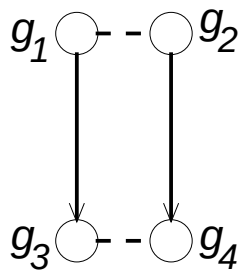
In addition to these simplifications, we also collapse the protein product and DNA promoter vertices of the same gene into the same node. They certainly correspond to different physical entities and should not be confused. However, whether a vertex denotes a protein or a DNA promoter is self evident according to the edges it appears: a protein-DNA edge is always from a protein node to a DNA node, and a protein-protein edge is always between two protein nodes. For the economy of notations and the convenience of conducting inference, we combine protein and DNA nodes in the skeleton graph. The simplified definition becomes

$$G = (V, E), E = E^{pd} \cup E^{pp}, E^{pd} \subseteq V \times V, E^{pp} \subseteq V \times V. \tag{2.2}$$

With this definition, at most three edges are allowed between two distinct nodes: protein-DNA interactions at both directions (if both vertices correspond to transcription factors) and a protein-protein interaction. Moreover, self edges (both protein-DNA and protein-protein) are allowed since a protein may be auto-regulated or bind itself to form a homo-dimer. The collapsed version of the toy skeleton graph in Figure 2-1 is shown in Figure 2-2. A solid line denotes a protein-DNA interaction and a dash line denotes a protein-protein interaction. When both a solid line and a dash line are incident to a vertex g, it denotes that both the promoter of gene g is the target of a DNA-binding protein and protein g interacts with another protein. We adopt this collapsed representation of the physical network throughout the thesis.

The knowledge of pairwise interactions that appear in the skeleton graph comes from experimental data. Since all experiments are subject to error, edges in the skeleton graph may not reflect the ground truth. Therefore, we should view the skeleton graph as a collection of *likely* physical interactions instead of *true* physical

Figure 2-2: Collapsed toy example of skeleton graph



interactions. We expect to include many physical interactions which are false positives or irrelevant to the gene regulatory mechanisms we are probing. The "likelihood" of the presence of an interaction can be inferred by incorporating evidence from multiple data sources. If the presence of an interaction is compatible with evidence from multiple sources, then it is more likely to exist and be involved in a gene regulation process. Conversely, if we exclude an interaction from the skeleton graph, we will not be able to use it to explain data or infer the properties associated with that edge.

Because false negative interactions deteriorate the explanatory power of the physical network model but false positive interactions do not, it seems natural to incorporate all possible pairwise interactions in the skeleton graph. In other words, the skeleton graph which does not have any false negative interaction is a complete graph which contains three edges (two protein-DNA interactions of opposite directions and one protein-protein interaction) between every pair of distinct vertices and two self-edges (one protein-DNA and one protein-protein interaction) for each vertex. Such a dense graph is neither a reasonable characterization of physical interactions nor computationally tractable at genomic scale. Therefore, we must trade off false negative edges and include only the interactions which are reported with decent confidence levels. If the data is already a list of likely physical interactions (such as the protein-protein interaction data from curated databases), then we directly incorporate these interactions in the graph. If the data reports the confidence (strength, affinity) of bindings among all pairs of genes (such as the protein-DNA interaction data from high-throughput chromatin immunoprecipitation experiments), then we choose

a threshold and incorporate those pairs whose confidence values pass the threshold. We will discuss the likelihood of a physical interaction, the effect of false negative edges and the choice of threshold values in Chapters Three and Four.

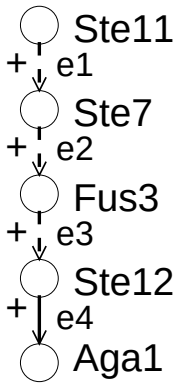
2.3 Model attributes and configurations

As the name of the skeleton graph suggests, likely physical interactions only provide a template to accommodate functions of gene regulation. In the analogy of an electronic system, the skeleton graph corresponds to the wiring diagram of a circuit. It informs us about the connections of devices in the circuit but does not specify the functions of these devices. One cannot reverse-engineer the function of the circuit by investigating its wiring diagram alone. We must either possess the knowledge about these devices by checking their serial numbers or figure out their functions by probing the system at various testing points. In biology, the functions of many genes remain unknown. Therefore, we need to infer the functions of these "devices" from empirical data.

However, genes may not be the basic functional units. First, many genes possess multiple functions. Which function is exercised depends on the interactions with other genes, cellular compartments, and environmental conditions. For example, the transcription factor Sok2 possesses both activating and inhibitory functions ([138]). Second, many genes need to cooperate with other genes in order to perform a specific function. Gene modules are very common in transcription regulation ([77]), signal transduction ([133]) and protein synthesis ([101]). For example, one of the largest module, ribosome complex, comprises two submodules. Each submodule is composed of a large number of ribosome RNAs and proteins.

What is the appropriate characterization of regulatory functions if genes are not the basic units? By linking the skeleton graph of physical interactions with the actual transcription regulation processes, we notice that the information about gene regulation is contained not only in vertices, but also in other features of the graph. For example, when specifying that a transcription factor f regulates a gene g by binding to its promoter region, the information that "f controls g" is associated with

Figure 2-3: A toy example of a physical network model



the edge from f to g. Furthermore, if we specify that f activates g, then this positive regulation can be represented as the positive sign of the edge $\vec{fg}$. Similarly, properties of transcription regulation can be associated with other aspects of the graph such as paths, cliques and clusters.

We enrich the network of physical interactions with various properties which allow us to describe certain transcription control mechanisms and to explain certain types of experimental data. In this thesis, the transcription control mechanisms are the cascades of transcription initiation control and signal transduction pathways. The experimental data of interest include physical interaction (protein-DNA and protein-protein interactions) data and gene expression data. By specifying these graph-related attributes, we are able to depict the transcription control via molecular interactions. In the toy example of the yeast mating pathway illustrated in Figure 2-3, pathway Ste11 $\rightarrow$ Ste7 $\rightarrow$ Fus3 $\rightarrow$ Ste12 specifies the Mitogen Activated Phosphorylation (MAP) pathway which transduces the mating signal generated by pheromone from the cellular membrane to the nucleus. Ste12 is a transcription factor and activates the genes related to mating responses including Aga1. Each gene in the preceding step of the pathway triggers the activity of the next step in the positive direction, thus all edge signs are positive. Notice the activity of a gene is not necessarily reflected in its mRNA or protein level, but can also be the chemical modification state (for example, phosphorylation). The directions and signs of edges in this simple graph

provide functional information about this pathway.

Our physical model contains the following types of attributes.

1. The presence or absence of a physical interaction denoted as x_e, where $e \in E$ is an edge in the skeleton graph. As mentioned previously, the presence of an edge is uncertain due to the false positives in the physical data. Thus we treat it as a random variable whose value will be inferred from the data. $x_e = 1$ if the interaction exists, and $x_e = 0$ otherwise. We denote $X_{E^{pd}} = \{x_{e_i} : e_i \in E^{pd}\}$ as the collection of protein-DNA edge presence attributes and $X_{E^{pp}} = \{x_{e_j} : e_j \in E^{pp}\}$ as the collection of protein-protein interaction attributes.

2. The causal direction of a physical interaction denoted as d_e. d_e specifies the direction of information flow in a cascade of molecular interactions. In a protein-DNA interaction, the direction of control flow is always from a protein (transcription factor) to a DNA (promoter). In contrast, the causal direction of a protein-protein interaction is undetermined from the physical data alone. We need to observe the functional processes in which this interaction is involved in order to infer its causal direction. Since the direction of a protein-protein interaction depends on the pathways in which this interaction participates, a protein-protein interaction can be bi-directional. It may possess one orientation in one pathway and the opposite orientation in the other. For simplicity we exclude this possibility in our preliminary physical model and assign each edge a unique direction. The following convention is adopted in order to map a causal direction into a binary value ($+1$ or -1). First we select a directed path as a reference. Then we assign $+1$ to the edge directions which are along the reference path, and -1 to the directions which are against the reference path. The selection of reference paths is arbitrary as long as their directions do not contradict with each other. Denote $D_{E^{pp}} = \{d_{e_i} : e_i \in E^{pp}\}$ as the collection of protein-protein edge direction attributes. The directions of protein-DNA edges are fixed, thus they do not need to be modeled as unknown attributes.

3. The immediate effect of a physical interaction denoted as s_e. s_e specifies whether

the the source node activates or represses the activity of the destination node. It can be treated as the sign of an edge. The semantics of s_e for a protein-DNA interaction designates the function of a transcription factor as an activator or a repressor. In contrast, the semantics of s_e for a protein-protein interaction designates the function of a signal transduction protein (such as a protein kinase) on the next step of the signal transduction pathway. Notice that the "activity" of a gene does not necessarily indicate its mRNA or protein level. For example, the activity of a transcription factor can be the conformation of the protein molecules, and the activity of a protein kinase can be the phosphorylation state of the molecules. Therefore, the signs of many edges are not directly observable from the gene expression data. The edge sign is also mapped into a binary value: $s_e = +1$ if the function is activation and $s_e = -1$ if the function is repression. Denote $S_{E^{pd}} = \{s_{e_i} : e_i \in E^{pd}\}$ and $S_{E^{pp}} = \{s_{e_i} : e_i \in E^{pp}\}$ as the collections of protein-DNA and protein-protein edge signs.

These attributes are treated as variables whose values are uncertain. We define a *configuration* of the physical network model as an instantiation of the settings of all variables associated with the skeleton graph. A configuration entails a specific model of gene regulation in terms of molecular interactions. As seen in the example in Figure 2-3, a configuration specifies the relevant molecular interactions, the causal orders of genes in the pathways and the regulatory functions of genes or gene modules. Furthermore, we can predict the consequence of perturbing the system (deleting or over-expressing genes) given a configuration. Because the physical network model is able to express the gene regulation mechanism corresponding to any of the possible configurations, it can be viewed as a *meta-model* of gene regulatory models.

The primary differences between the physical network models and previous works introduced in Chapter One are the assumptions about gene regulation mechanisms and the properties they attempt to capture. Most previous works focus on modeling the attributes associated with genes. For example, Bayesian networks on gene expression modeling in [56, 71, 123] all treat the mRNA expressions of genes as the variables in the model. Properties about the structure of a Bayesian network emerge from the

dependencies of the expression data. In contrast, we assume the structure of the network is pre-determined by physical interaction data and explicitly model various properties associated with the structure. Learning the attributes with a fixed structure is typically undertaken by model inference algorithms such as max-product or sum-product, while learning the model structure often resorts to sampling or greedy search algorithms. Both problems have no efficient algorithms that guarantee to find the solutions, but empirically structure learning is considered more difficult than model inference. However, it should be pointed out that the difference between physical network models and previous works of Bayesian network gene expression models is not computational. One can formulate a physical network model as a Bayesian network with variables (nodes in the Bayesian network) defined as attributes of vertices, edges and paths in the underlying physical network.

2.4 Data association and model inference

The purpose of a model is to study a complex subject with simplified and generalized assumptions. Therefore, a model is useful only when it links to data obtained from observations or experiments. In the context of physical network models, the configurations of annotated attributes are determined by empirical data. Without any data each configuration is equally likely. As more data are included the uncertainty of model configurations reduces; in other words, there are fewer configurations compatible with existing data. The reduction of uncertainty can be understood from two aspects. The uncertainty pertaining to a specific variable is reduced if there are multiple data points probing it. For example, a protein-DNA interaction is more likely to be real if it is supported by both CHIP-chip assays and promoter sequence analysis. In addition, the overall uncertainty of model configurations is reduced when the model is constrained by distinct yet related data. For example, the edge signs along a pathway can be uniquely determined if each gene along the pathway is deleted and the deletion effect on downstream genes are measured.

Linking an observed data with variables in the model is called a *data association*

problem. When applying to physical network models, we categorize observed data into two types: physical and functional data. A physical data is directly tied to a single variable in the physical network model. In other words, the experiments which report a physical data directly measure an attribute in the physical network model. The data of protein-DNA interactions, protein-protein interactions, and protein complexes all fall into this category. There is no ambiguity of interpreting the data because they directly measure the attributes. When there are multiple data probing the same interaction, we can assign the confidence of this interaction by combining the confidence measures from multiple data. Therefore, the association with physical data is direct and does not depend on other attributes in the model.

In contrast, a functional data is related to multiple variables in the physical network model. We use the term *functional* because they probe the relations between inputs (perturbations) and outputs (the changes of states such as gene expression levels) of a system but do not specify the mechanisms underlying such relations. However, since we are interested in modeling the mechanisms, we need to assign the functional relations with attributes listed above.

All the gene expression data – including mRNA and protein level measurements – fall into the category of functional data. Many gene expression experiments measure the differential changes of deletion or over-expression mutants with respect to the wild type. The cause and effect in these data are clear, though the intermediate steps remain unknown. The changes of gene expression are the effects of the perturbation, and the causes are the changes (deletion or over-expression) of the perturbed genes. We can decompose a knock-out or over-expression data into these cause-effect pairs. For instance, if deleting a gene g up-regulates one hundred genes $g_1, \cdots, g_{100}$ and down-regulates fifty genes $g_{101}, \cdots, g_{150}$, then we can decompose this expression data into 150 triplets $(g, g_1, +), \cdots, (g, g_{150}, -)$, where the first and the second elements stand for the cause and the effect of the experiment and the third element reports the direction of the expression change. For simplicity we only consider the qualitative changes of knock-out experiments (up or down regulations or no change). Quantitative changes – the strength of expression changes – can be transformed into the

probabilities of qualitative changes.

Other gene expression data probe the response of the system in different types of cells or under different environmental conditions. It is not straightforward to assign the cause of expression changes to specific genes as in knock-out or over-expression data. Nevertheless, with certain assumptions we can assign the candidate causes of a gene expression change to the bindings of transcription factors at its promoter region. Other functional data include the phenotypical observations under perturbations. For instance, in budding yeasts the complete list of lethal genes (the deletion of a single gene is lethal) and many pairs of synthetic lethal genes (the deletion of individual genes is not lethal, but the deletion of both genes is lethal) are readily available ([64, 169, 157]). We will not utilize this type of functional data in this thesis, despite it is possible to incorporate them in the physical network model in the future work.

We can interpret a functional relation (cause and effect) in a functional data with the mechanisms of molecular cascades. To make these mechanisms explain a functional relation, the model attributes along the paths connecting the cause and the effect must satisfy certain constraints. Using the example in Figure 2-3 again, suppose knocking out Ste11 down-regulates the mRNA expression of Aga1. To make the path in Figure 2-3 explain this relation, the protein-protein interactions should follow the direction Ste11 $\rightarrow$ Ste7 $\rightarrow$ Fus3 $\rightarrow$ Ste12 $\rightarrow$ Aga1 and the aggregate sign of the four edges must be +1. These constraints usually do not uniquely determine the value of an attribute, but they narrow down the space of possible configurations.

Both confidence values derived from physical data error models and hard constraints obtained from explaining the functional data can be represented as *potential functions*. A potential function maps each configuration of a set of variables into a non-negative real number pertaining to the constraint. The potential function of a physical data observation contains a single argument of the model attribute and the returned value is proportional to the likelihood ratio of measurements. For example, if the conditional likelihoods of observing a binding affinity value in a chromatin IP experiment are $\mathcal{L}_1 = P(y_e|x_e = 1)$ and $\mathcal{L}_0 = P(y_e|x_e = 0)$ respectively (where x_e denotes the actual interaction and y_e the observed binding affinity), then we can

construct the potential function of this interaction x_e as

$$\phi(x_e) = \left(\frac{\mathcal{L}_1}{\mathcal{L}_0}\right)^{x_e}. \tag{2.3}$$

The potential function of a functional data involves multiple attributes to explain the functional data. It returns a relatively large value (close to 1) if the configurations of these variables satisfy the constraint of explanation, and a relatively small value otherwise. In the example in Figure 2-3, suppose Aga1 is down-regulated in Ste11Δ experiment. The potential function corresponding to this knock-out effect is a function of the edge presence, edge directions and edge signs along the pathway. It returns a high value when edge presence and directions are consistent with the pathway and the aggregated edge sign is consistent with the knock-out effect.

$$\psi(x_{e_1}, \cdots, x_{e_4}, d_{e_1}, \cdots, d_{e_4}, s_{e_1}, \cdots, s_{e_4}) = \begin{cases} 1.00 \text{ if } x_{e_1} = \cdots = d_{e_4} = +1, \prod_{i=1}^{4} s_{e_i} = +1. \\ \epsilon \text{ otherwise.} \end{cases} \tag{2.4}$$

Notice the returned values of a potential function do not need to sum to one. The normalization constant is immaterial when inferring the attribute configurations. Individual potential functions correspond to disjunctive constraints that need to be simultaneously satisfied. Therefore, we construct a joint likelihood function of the model by multiplying potential functions from each physical and knock-out interaction.

The goal of model inference is to find the model configurations which are consistent with the physical data measurements and functional data explanation as good as possible. This amounts to finding the optimal configurations which maximize the joint likelihood function. The optimal configuration restricted to each single variable is called the max-marginal probability and can be approximated by the max-product algorithm ([97]). By recursively applying these algorithms, we can either enumerate all optimal configurations or decompose the physical network model into submodels such that optimal configurations within each submodel are independent of the others. This decomposition provides an expressive power to represent all optimal

configurations without explicitly enumerating them. We will discuss the model inference procedures in Chapter Three.

2.5 Experiment design

The physical network model provides a consistent explanation for the physical and functional interactions in the existing datasets. However, existing data may not impose sufficient constraints to narrow down the space of configurations to a useful extent. The functional interactions may be caused by mechanisms other than molecular interactions. In addition, the physical network model may explain false positive interactions and draw inaccurate conclusions from the erroneous data. All these problems can only be resolved by acquiring data from new experiments. Although high-throughput technologies drastically improve the measurement efficiency of experiments, it is still expensive and time-consuming to do biological experiments. To make the best use of limited resources, it is critical to combine the modeling framework with the strategies of designing new experiments.

In this thesis, we focus on prioritizing the experiments of measuring expression profiles in single deletion mutants. This is because it takes little extra effort to incorporate new knock-out gene expression data in the model, and profiling mRNA expressions of yeast single deletion mutants is cheap and accessible compared to other assays. We use this type of experiments to discriminate degenerate configurations obtained from existing data. Model degeneracy in molecular cascades may arise from the freedom of assigning edge signs and edge directions. Eventually, we want to narrow down these likely configurations to a small number so that we can test these models in more details.

Given a probability distribution of a large number of configurations, how do we find a knock-out experiment which can best distinguish them? The capacity of a new perturbation experiment to discriminate candidate models relies on how diverse responses the perturbation can evoke according to model predictions. If the new perturbation yields distinct outputs for each different model, then we are able to identify

a single configuration from this experiment: just choose the model whose prediction is the closest to the observed output of the new experiment. It is also possible that the experiment outcomes contradict with all predictions and the true model is not in the candidate list. In most cases, a knock-out experiment is not able to identify a unique model because multiple configurations may predict the same outcomes in a knock-out experiment. To determine which experiment to take at the next step, we need to define a quantity to gauge the discriminative power of a knock-out experiment. In this thesis we use the Shannon entropy about network attributes to represent model uncertainty. The discriminative power of an experiment is the reduction of model uncertainty given the predicted outcomes of the experiment. This quantity is reduced to mutual information between the identity of model configurations M and the predicted response under experiment e Y^e:

$$H(M) - H(M|Y^e) = I(M; Y^e). \tag{2.5}$$

Because we only include significant knock-out effects (up or down regulations) in the physical network model but do not employ the information about insignificant effects (a gene does not change in a knock-out experiment), the mutual information score is revised so that it only incorporates significant predicted outputs in model discrimination. Discussions about revising the mutual information score will be covered in Chapter Five.

2.6 Combinatorial regulation of multiple transcription factors

Most gene promoters are bound by multiple transcription factors ([100]). These factors are likely to regulate genes in a coordinated fashion. Understanding the mechanisms of combinatorial control pertaining to multiple regulators at systems level is a challenging problem. It requires the information about various gene regulatory mechanisms in a genomic scale: protein-DNA bindings, complex formations, post-

translational modifications, protein abundance and localization, binding occlusions, and so on. Currently only part of those information are available.

Due to the lack of data which can reveal mechanisms, most computational works focus on inferring the functional relations of multiple transcription factors. A functional model establishes the relations between the "states" of regulators and regulated genes while leave the underlying mechanisms unspecified. For example, we may infer a transcription factor is a repressor but do not specify how it represses gene regulation. In reality, repression may be achieved by blocking the binding site of an activator, altering the conformation of an activator protein to disable its function, and so on. The states are typically mRNA or protein abundance of genes or other attributes which indicate the activities of genes. We choose mRNA levels of genes as their states due to the availability of data.

In the context of combinatorial control, a function specifying the relation between mRNA levels of genes can be formulated as a discrete function with noisy outputs. The number of such combinatorial functions grows super-exponentially as the input size. Therefore, an essential step of building a model for combinatorial control is to simplify these functions. A straightforward approach is to consider the effects of single regulators independently. This approach can retrieve the functions of single regulators but does not consider the combinatorial effect of multiple regulators. We extend the scope of independent regulator effects and consider the simple combinatorial effects that can be inferred from limited data. Geneticists often characterize the properties of single regulators in the context of combinatorial control. We decompose the properties of single factors into two dimensions: the function of a regulator as an activator or a repressor, and the direction of effectiveness of a regulator. The direction of effectiveness specifies in which direction the change of a regulator can lead to the change of regulated genes. A regulator is necessary if repressing it disrupts the normal function of the regulator. Conversely, a regulator is sufficient if increasing its activity enhances its normal function. A regulator can also be both necessary and sufficient or neither. At transcription level, we tell whether a regulator is necessary or sufficient by checking whether the mRNA change of the regulated gene is accompanied by the

mRNA change of the regulator in a specific direction (up or down regulation). For example, if a regulator is a sufficient activator, then it can explain the data that both the regulator and the regulated gene mRNAs are up regulated. One simple mechanistic explanation for necessary regulators is that they collaborate with other proteins in order to function. Similarly, a mechanistic explanation for sufficient regulators is that they are in redundant pathways which can independently function.

This characterization certainly does not capture all combinatorial effects. The direction of effectiveness of a regulator may depend on the presence or absence of other regulators. Moreover, this functional characterization covers only a specific mechanism of combinatorial regulation: regulators control transcription initiation by modulating their protein abundance (indirectly mRNA abundance). A transcription factor may regulate transcription by modulating the number of proteins bound to a specific promoter. The protein abundance localized on a promoter may not be proportional to the average mRNA abundance of the regulator captured by microarrays. Hence we may not be able to uncover the effect of this regulator from expression data alone. In spite of these limitations, this characterization reduces the number of possible functions from super-exponential to exponential in terms of input size. Therefore, it allows us to enumerate all possible functions for small input sizes. We will discuss the advantages and limitations of this characterization in Chapter Six.

We define the likelihood function of binding and expression data in order to fit a regulatory model to those data. For binding data, the likelihood function is translated into the constraint that all regulators bind to all regulated genes within a model. For expression data, the likelihood function pertains to the consistency of gene expression changes between regulators and regulated genes with respect to the combinatorial function. The actual binding and expression states are observed through noisy measurements. We marginalize the likelihood function over the hidden variable states consistent with the regulatory model.

Once the likelihood function is defined, we propose an algorithm which generates regulatory models that maximize likelihood scores. For a fixed set of regulators and a combinatorial function, the algorithm incrementally adds genes which maxi-

mize the likelihood score. It then selects the optimal and sub-optimal combinatorial functions according to their likelihood scores, and infers each regulator's direction of effectiveness from them. Finally it keeps the regulatory models which substantially fit the data. The algorithm and the analysis on high-throughput data are discussed in Chapter Six.

Chapter 3

Integrating Data in a Physical Network Model

We have described the motivation and the framework of the physical network models at a conceptual level. In this chapter, we will discuss in-depth the two core aspects of the physical network models – the association with the empirical data and the inference of the likely configurations of the model. By applying an independence assumption, we decompose each data into the evidence of pairwise interactions. This evidence includes protein-DNA and protein-protein interactions captured by various assays and the differential expression changes of genes in a single gene deletion mutant.

Both physical and knockout evidence impose constraints on the variables (edge presence, directions, signs, etc.) of the model. The goal of model inference is to find the configurations of variables which satisfy these constraints. To facilitate model inference, we express each constraint as a *potential function* of variables. Model inference then amounts to optimizing the joint likelihood function generated by the potential functions. This optimization can be efficiently approximated by various inference algorithms of graphical models. In this thesis we apply a special class of the message-passing algorithms – max-product and sum-product algorithms of factor graphs – in model inference. Furthermore, by recursively applying the max-product algorithm, we are able to decompose multiple optimal configurations into the product of the configurations of submodels.

3.1 Data sources

It is necessary to discuss the types of data sources we use before introducing data integration and model inference. In this thesis, we focus on the datasets which directly capture pairwise physical interactions and the datasets which measure the mRNA levels of deletion mutants in comparison with wild types. We name the first type *physical data* and the second type *functional data*. The distinction relies on whether the data probe the fundamental attributes in the model. The expression data under gene knockouts are functional because knockout interactions are composite effects related to multiple fundamental attributes such as directions and functions of physical interactions.

3.1.1 Protein-DNA interaction data

As described in Chapter One, the bindings of transcription factors on DNA promoters are necessary for transcription initiation. Currently there are several major methods of probing protein-DNA interactions in a large scale. Here we briefly introduce two of them. The least involved in terms of "wet" biological bench work is the sequence analysis of DNA promoters. The analysis is based on the postulation that the DNA sequence on the binding site of a specific transcription factor is closely related to its binding affinity. Thus by investigating the promoter DNA sequences, we are able to identify putative binding sites of transcription factors. A common approach of sequence analysis is to first select the putative promoter targets of a transcription factor (for example, by identifying the genes whose expression profiles are correlated), then find the *motifs* – statistically enriched sequence patterns – in putative targets. Many algorithms have been developed to identify motifs from promoter sequences; some examples include ([130, 162, 10]). Despite its usefulness, sequence analysis has two major limitations: the analysis does not directly observe protein-DNA bindings and the sequence information alone may not be sufficient to determine the bindings.

Other techniques such as chromatin immunoprecipitation can directly probe protein-DNA bindings. Chromatin immunoprecipitation – abbreviated as chromatin IP or

CHIP – purifies and amplifies the promoter segments bound by a specific transcription factor. Chromosomal DNAs which are bound by DNA-binding proteins are cleaved in vivo into small fragments. The target promoters of a transcription factor are purified by immunoprecipitation using the antibody specific to the DNA-binding protein. The purified promoter fragments are then amplified by polymerase chain reaction (PCR) and measured by Northern blot for single genes or DNA microarrays for high-throughput outcomes. The measurement outcome is compared to the background reading in control experiments where immunoprecipitation does not take place. This technology is also called CHIP-chip when DNA microarrays are used to measure the bindings of a large number of promoters. A technical introduction about chromatin IP can be found in [128].

Currently there are several large-scale datasets of CHIP-chip experiments available in budding yeasts. One of the most comprehensive datasets is the location analysis data published by Lee et al. ([100]). This dataset consists of the genome-wide binding profiles of 106 transcription factors in *S. cerevisiae*. Other similar datasets can be found in [78].

In this thesis, we incorporate the high-throughput CHIP-chip data published by Lee et al. in the physical network model. This choice is governed by the availability and coverage of data. The modeling framework, however, is capable of incorporating other types of data.

3.1.2 Protein-protein interaction data

In addition to protein-DNA interactions, physical interactions between proteins also play an important role in gene regulation. As described in Chapter Two, protein-protein interactions can influence transcription regulation via at least two mechanisms. A protein may chemically modify another protein and propagate the information of gene regulation by protein modification, or it may bind to other proteins to form a complex and carry out a specific function. Both mechanisms may also occur simultaneously on a protein-protein interaction. We focus on the signal pathway aspect of protein-protein interactions in this chapter.

Various techniques have been developed to detect pairwise interactions of proteins. Some methods include yeast two-hybrid systems ([160, 86]), co-immunoprecipitation ([146]) and mass spectrometry ([75, 62]). The basic idea of yeast two-hybrid systems is as follows. A eukaryote transcription factor protein consists of two essential subcomponents (*domains*): one is responsible for binding to DNA promoters and the other interacts with the RNA polymerase II holoenzyme to activate transcription initiation. To detect whether proteins A and B bind, the DNA sequence encoding transcription factor Gal4 DNA binding domain is inserted to the 5'-end of gene A, and the DNA sequence encoding Gal4 transcription activation domain is inserted at the 3'-end of gene B. Thus the expressed protein A contains Gal4 DNA binding domain, and the expressed protein B contains Gal4 activation domain. If proteins A and B bind together, then the complex (Gal4 DNA binding domain-A-B-Gal4 activation domain) acts as Gal4. We can probe the activity of this complex by measuring the expressions of Gal4-controlled genes such as Gal3.

Yeast two-hybrid experiments can be applied at either a small or large scale. There are already several high-throughput datasets of protein-protein interactions generated by yeast two-hybrid systems, such as [160] and [86]. The quality of those datasets is often questionable, for the false positive and false negative rates are reported to be high ([31]). For instance, the datasets generated by two different laboratories but under the same environmental condition and of the same model organism have less than 30% overlap ([160] and [86]). In contrast, the results generated from small-scale experiments are generally more reliable ([31]).

Co-immunoprecipitation is applied at a small scale to detect pairwise protein interactions. Suppose we want to test whether proteins A and B bind. The antibody specific to A is applied. The antibody-A complex is purified with immunoprecipitation. The purified complex contains B if B already binds to A. We can detect the presence of B by applying another antibody specific to B and performing a Western blot assay.

Mass spectrometry has become another primary technology of measuring protein-protein interactions. Mass spectrometry is typically employed to detect the presence

of specific molecules (in this case, protein complexes) in a specimen. The specimen is ionized and the ionized molecules move under an electric field. Similar to electrophoresis, ionized molecules are separated by their mass-to-charge ratios. Therefore, the presence of certain molecules can be detected by reading the mass spectra (the number of ions in each mass-to-charge ratio) of the specimen. Since the charge-to-mass ratio of a molecule is unique, it is possible to detect the presence of multiple types of molecules simultaneously from the aggregate spectra of the specimen. This property makes mass spectrometry intrinsically high-throughput. Therefore, it has been recently applied in many problems of systems biology such as proteomics, metabolic flux balance analysis and protein-protein interactions. A comprehensive review of current mass spectrometry technology in proteomics is given in [1].

In comparison to yeast two-hybrid systems and co-immunoprecipitation, mass spectrometry can detect protein complexes beyond pairwise interactions. However, current high-throughput datasets also suffer from high false positives (reported complexes which are artifacts) and false negatives (known complexes are not reported).

Currently, there are several on-line databases of pairwise protein-protein interactions reported from literature. Examples include the Database of Interacting Proteins (DIP) curated by UCLA [1] ([31]) and Biomolecular Interaction Network Database (BIND) maintained by University of Toronto [2] ([8]). Most databases do not explicitly annotate the experimental technology which reports those interactions, and none of the databases annotates the environmental conditions and assesses the confidence of reported interactions based on information contained in individual sources. Some databases such as DIP flag individual interactions according to the scale of experiments or whether they are reported from multiple sources. These properties (e.g., whether an interaction is reported in a large-scale assay, whether an interaction is reported from multiple assays) allow to create subcategories of protein-protein interactions. Interactions in certain subcategories are more reliable than other subcategories. For example, the interactions detected by small-scale experiments and

[1] http://dip.doe-mbi.ucla.edu/
[2] http://www.blueprint.org/bind/bind.php

confirmed by different types of technologies are more likely to occur than the interactions reported only by high-throughput yeast two-hybrid experiments. The confidence of interactions in each subcategory can be estimated by external experiments ([31]).

Like all other genomic databases, the databases of protein-protein interactions continue growing. Despite their large and growing sizes, all the known protein-protein interactions may constitute only a small fraction of all actual pairwise interactions ([31]). Many reported interactions are generated by few (two or three) high-throughput experiments, and most small-scale experiments focus on several small subsystems which are well characterized and studied. Therefore, the current knowledge about protein-protein interactions is both incomplete and biased.

In this thesis, we choose the protein-protein interaction data collected in the DIP database for it provides a systematic measure on the confidence of interactions. The details of the confidence evaluation will be discussed in the next section.

3.1.3 Gene expression data

Both protein-DNA and protein-protein interactions capture crude aspects of mechanisms but not consequences of gene regulation. To understand the functional aspect of gene regulation it is necessary to measure its "outputs" – mRNAs and proteins – under various conditions.

High-throughput gene expression analysis has become a standard tool in most biological laboratories. Those experiments quantitate the mRNA or protein levels of a large number of genes. There have been a rich collection of gene expression data under various conditions, and new datasets are generated in a fast pace. Here we give a very brief overview about gene expression data relevant to our work.

Gene expression denotes the synthesis of mRNAs and proteins. Thus gene expression data cover the measurements of both mRNA and protein abundance. Currently mRNA expression data are far more abundant than protein expression data. Many technologies of high-throughput mRNA measurements – such as Affimetrix gene chips ([50]) or two-channel DNA microarrays ([35]) – are based on DNA hybridization. DNA segments complementary to specific genes are implanted on different spots of a small

substrate called a chip. The mRNAs from the whole cell extract are converted to complementary DNAs (cDNAs) by reverse transcriptase and then hybridized with the probes on the chip. The hybridized chip is scanned, and the quantities of mRNA molecules captured on the chip can be measured by the fluorescence levels of spots in the scanned image. One can find brief yet informative overviews of these technologies in [70, 174] and the lecture note of MIT 2004 spring course "Computational functional genomics" [3].

We can also categorize gene expression data in terms of experimental conditions. Perturbation data probe gene expression under internal or external perturbations and compare the measurements to the data from the *control experiments* without perturbations. Relative changes with respect to the control experiments are reported. Internal perturbation denotes disturbing the internal machinery of cells. The most common internal perturbations are deleting single genes. A comprehensive data of yeast gene knock-out expression is the Rosetta Compendium data ([80]). It reports the genome-wide mRNA expressions of 300 experiments, including 271 single deletions, 5 double deletions and 24 drug response experiments. We use the subset of single knock-out experiments to constrain the physical network models. Other internal perturbations such as double knockouts ([157]) or over-expressions ([61]) have also been applied. External perturbations denote altering the external environment of the cellular culture. These perturbations expose cells under some abnormal (often stressful) conditions such as high (low) temperature, nutrient starvation, or addition of drugs. A comprehensive profiling of gene expressions under various stress conditions is reported by Gasch et al. [61].

The fundamental difference between internal and external perturbations in our work is that we can attribute the causal effects of internal perturbations to the changes along physical interaction pathways. External perturbations such as environmental stress often enter cells via receptors on the cellular membrane. We do not have information about the transduction from environmental signals to protein states, thus cannot reconstruct the pathways of their causal effects.

[3] http://psrg.lcs.mit.edu/6.874/lectures.html, lecture 7.

Observational data measure the expression of cells without perturbations. Without clearly defined control experiments, they often focus on the variations across tissue types ([66, 127, 2]), cell cycle stages ([143, 23, 11]), and both spatial and temporal variations in embryo development ([167]). Observational data provide snapshots of the internal states of the cell, but they do not directly capture the causal orders of gene regulatory events. It is possible, though, to reconstruct causal orders of genes from the temporal orders of their expressions.

3.1.4 Other types of data

Physical interaction and mRNA expression data certainly do not constitute all high-throughput datasets in systems biology. Many new technologies have been developed or are under active development. These technologies probe different aspects of the cellular processes. The growth of data in terms of quantity and variety is tremendous Here we briefly review some of the data types which we do not currently incorporate in the physical network model. We consider integrating some of them in the future work.

DNA sequence information is the earliest large-scale genomic data available. With the progress of automated sequencing technologies, the genomes of many organisms are being sequenced at a fast pace. Current focus on sequence analysis is on comparing the genomes of multiple organisms (for instance, [95]) and integrating sequence information with other data (for instance, [100]).

Many cellular functions are performed by complexes comprised of multiple proteins. The multi-way interactions of complex formation can be viewed as a generalization of pairwise protein-protein interactions. Curated databases about known protein complexes are available for a limited number of organisms [4]. In addition, several large-scale datasets from high-throughput experiments are already published ([75, 62]). Similar to the pairwise interactions, the quality of these datasets is also questionable.

Localization of proteins in different cellular compartments serves as an important

[4] http://mips.gsf.de/

mechanism for various cellular processes. A number of large-scale datasets about protein localization are available. Using fluorescence tagging, the presence of proteins in each cellular compartment can be visualized using microscopes ([81]). In addition, statistical analysis is also applied to infer the localization of proteins ([98]).

Protein expression (proteomics) is currently under active research. They are certainly important for understanding gene regulation since proteins are actual functioning units of cells. Currently protein expression data are not as common as mRNA expression data due to the high cost. Limited high-throughput data are available such as ([63]). With the progress of mass spectrometry, those data are expected to become more accessible.

In addition to molecular quantities such as mRNA or protein levels, researchers also measure cellular properties in a high-throughput fashion. For example, phenotype arrays measure the cellular phenotypes under different perturbation conditions ([64]). A special case of phenotype is cell death. There are also high-throughput experiments to detect lethal single gene knock-out experiments ([156]) and synthetic lethal double gene knock-out experiments ([157]).

3.2 Pros and cons of data integration

An increasing number of recent works in computational biology incorporate multiple sources of data. Aside from the convenient access to different types of data, there are several major arguments for favoring using multiple data sources in inferring gene regulation. First, different types of data contain overlaping information regarding the underlying system. These overlapped information can be used to reduce the ambiguities of the inferred models. In some cases, several different assays are designed to capture identical or similar properties. Data fusion in this scenario is statistical: a property is measured by independent experiments, and the noise after multiple observations is reduced. Many previous works pertaining to genomic data fusion follow this track. For instance, in [143] and [152], putatively co-regulated genes are reported by combining gene expression and promoter sequence motif data. On the other hand,

model uncertainty may also be reduced by different types of constraints from data. In our problem, physical data provide information about possible mechanisms for gene regulation, while knockout data reveal the causal order and functional effects of gene regulation. By putting two types of information together, we may be able to uniquely determine the attribute values in the physical network.

While the overlapped information between datasets helps reduce model uncertainty, the orthogonal information among them expand the scope of the model. Knowing both physical and functional aspects of gene regulation is certainly better than knowing each aspect separately. However, the information contained in multiple datasets are not simply the concatenation of information from individual datasets. This is because different datasets can be interrelated as described above. By applying proper assumptions we can uncover the information which are not contained in single datasets.

Despite its advantages, data integration also introduces new problems. The quality of each dataset becomes crucial if we want to extract information by synthesizing data. The errors in the overlapped part are more tolerable since this information is contained in multiple sources. In contrast, the errors in the orthogonal part are accumulated. For example, if both physical and knockout data are erroneous, then we may assign a wrong pathway to explain a wrong knockout effect.

3.3 Overview of the data association approach

Relating model variables to measurements is called *data association*. One typical data association problem in machine learning is to relate the measured data about a place (images, laser scans, sonar scans, and so on) with the position of this place in the internal map stored in the robot ([112]). Because of the uncertainty about the robot position and the inaccuracy of the map, associating a captured image to a corner in the map poses a non-trivial computational problem. In our problem, many variables in the model are not directly observed, and many measurements capture the aggregate processes involved with multiple variables. Thus it is essential to establish

the rules of relating measurement data to model variables.

Before establishing the rules of data association, we need to demarcate the basic units of the data. All the high-throughput data are quantitative descriptions of gene relations. The fundamental properties of these relations are the same as the descriptions in classical assays. Therefore, we can decompose high-throughput data into the following basic units which can be understood in the context of classical assays. Protein-DNA interaction data are essentially a collection of pairwise relations. We express a pairwise relation as an ordered pair (f, g), denoting that transcription factor f binds to the promoter of gene g. We can either assign confidence measures (or p-values) to these pairs or treat them as discrete events. Pairwise protein-protein interactions data can be expressed as an unordered pair: (g_1, g_2) denotes that proteins g_1 and g_2 bind together. Knock-out or over-expression data of gene expression can be treated as relations between perturbed genes and affected genes. For single knock-out data, a signed and ordered pair $(g_1, g_2, +/-)$ denotes that deleting g_1 up or down regulates g_2. For simplicity we only consider qualitative changes (up or down regulation or no change) of knock-out effects.

Protein-DNA, protein-protein interactions and knock-out gene expression data can be decomposed into these basic units of gene relations without ambiguity. Each decomposed relation from these three datasets (protein-DNA, protein-protein and knock-out interactions) imposes a constraint on the physical network model. Evidence from physical data is directly linked to a variable of physical interactions: it informs us whether a physical interaction exists or the confidence about the observation. Evidence from knock-out data contains two layers. The first layer is the existence or the confidence about an actual knock-out effect: whether gene 2 is up/down regulated or unchanged by deleting gene 1. The second layer is the interpretation of this actual knock-out effect according to the physical network model: what are the constraints on the directions and signs of the physical interactions along the pathways connecting the cause and effect genes.

The problem of data association amounts to finding a mathematical representation for these constraints. One common representation is *potential functions* of discrete or

continuous variables. It maps each state of variables into a non-negative real number reflecting the confidence pertaining to the constraint. To be precise, let $x_1, \cdots, x_n$ be n variables and D_i is the domain of variable x_i. A potential function of $x_1, \cdots, x_n$ is defined as

$$\phi : D_1 \times \cdots \times D_n \rightarrow R^+. \tag{3.1}$$

One can immediately see a hard constraint such as a Boolean function is a special case of a potential function: ϕ returns 1 if the constraint is satisfied and 0 otherwise. In fact, potential functions are used to represent the hard constraints of complex satisfiability problems. For example, in decoding complex error correction codes such as turbo codes ([15]) or Gallager codes ([107]), parity check functions of received bits are expressed as potential functions ([55, 173]). Moreover, potential functions can also express (unnormalized) probability functions of random variables. For instance, exponential families such as Bayesian networks and Markov random fields have equivalent forms as the product of potential functions ([173]).

We decompose each dataset into simple constraints pertaining to pairwise relations. Each constraint yields one potential function term. The construction of potential functions pertaining to each type of data will be introduced in subsequent sections. Potential functions are joined by multiplication because we assume constraints are independently imposed and require all constraints need to be (ideally) simultaneously satisfied. These premises may not hold in general. For instance, errors of adjacent spots on DNA microarrays may be correlated, or some constraints are linked in an OR fashion (that the satisfaction of any one constraint suffices). We leave the first problem to subtle error models in the future and discuss the treatment of the second scenario in the next section.

Notice we should not confuse the independence of constraints with the independence of data. Multiple datasets capture different (or even identical) aspects of the same biological system. Hence these data are indeed highly dependent. This dependency, however, does not prevent us from decomposing data into independent constraints. More precisely, individual datasets are conditionally independent given

the underlying process.

3.4 Constructing potential function terms

In this section we will discuss the construction of potential function terms for three datasets: protein-DNA interactions, protein-protein interactions and knock-out gene expression data. We focus on a specific instance for each type of data: yeast genome-wide chromatin IP for protein-DNA interactions ([100]), protein-protein interaction database DIP ([31]), and the Rosetta Compendium dataset for yeast knock-outs ([80]). The procedure of data association can be extended to other datasets of the same types if their measurement error models are provided.

3.4.1 Location analysis data

The raw data of CHIP-chip experiments are the images of two-channel microarrays of promoter nucleotides. One channel reflects the population of DNA promoter fragments bound by the target transcription factor and purified by immunoprecipitation, and the other channel measures the background population without immunoprecipitation. The image files were pre-processed (mapping spot intensities into real numbers, error correction, normalization, and so on) and converted to real-valued matrices as other gene expression data. The ratio (or the log ratio) of the readings on two channels reflects the enrichment of a transcription factor-bound promoter. We define the ratio as $\frac{\text{purified}}{\text{background}}$. The larger the ratio is, the more likely the promoter fragment is enriched after the purification. However, the ratio is also affected by the variations of the readings on each spot. To take the spot-specific noise into account, a number of control experiments (background versus background in both channels) were undertaken. A null model was constructed from readings of control experiments, and the p-values were computed according to this null model. The error model in location analysis data was adapted from the error model in the Rosetta Compendium data. Detailed discussions about this error model can be found in the supplementary webpage of [80].

We threshold on the reported p-values and only consider the factor-gene pairs whose p-values are below the threshold. For each candidate pair, we construct a potential function that incorporates its p-value information. To do this we introduce the following notations.

- $E^{pd} = \{e_i = (f, g)\}$ is the collection of all factor-gene pairs which pass the p-value threshold on location data.
- $X_{E^{pd}} = \{x_{e_i} : e_i \in E^{pd}\}$ denotes the indicator variables whether these protein-DNA pairs interact or not. They are observed through noisy measurements.
- $Y_{E^{pd}} = \{y_{e_i} : e_i \in E^{pd}\}$ denotes the measurements about the protein-DNA bindings. They are directly reported and their domains are real numbers. We can interpret y_{e_i} as the log ratio of the two channel readings or the p-value derived from the log ratio.

The potential function $\phi_{e_i}(x_{e_i}; y_{e_i})$ pertaining to the location analysis evidence about a protein-DNA interaction e_i is proportional to the ratio of the conditional probabilities derived from the error model:

$$\phi_{e_i}(x_{e_i}; y_{e_i}) \quad = \quad \left[\frac{P(y_{e_i}|x_{e_i}=1)}{P(y_{e_i}|x_{e_i}=0)}\right]^{x_{e_i}}. \tag{3.2}$$

$\phi_{e_i}(x_{e_i}; y_{e_i})$ is a function of x_{e_i} only since the value of y_{e_i} is given by the data. We simplify the notation by stating that $P(y_{e_i}|x_{e_i} = 1)$ denotes the conditional probability of observing "y_{e_i} = the empirical value" given $x_{e_i} = 1$. ϕ_{e_i} returns the likelihood ratio $\left[\frac{P(y_{e_i}|x_{e_i}=1)}{P(y_{e_i}|x_{e_i}=0)}\right]$ if $x_{e_i} = 1$, and 1 if $x_{e_i} = 0$. The problem becomes evaluating conditional probabilities $P(y_{e_i}|x_{e_i} = 1)$ and $P(y_{e_i}|x_{e_i} = 0)$.

Suppose the error models of both false positives and false negatives were provided, then we could directly apply those models to evaluate $P(y_{e_i}|x_{e_i} = 1)$ and $P(y_{e_i}|x_{e_i} = 0)$. In reality, a complete characterization of measurement errors is not yet available. In the CHIP-chip assay we use, the p-values associated with each protein-DNA interaction are provided. These p-values are heuristically defined instead of formally defined from known distributions.

How do we estimate $P(y_{e_i}|x_{e_i} = 1)$ and $P(y_{e_i}|x_{e_i} = 0)$ from empirical p-values? The ambiguity of the p-value definition allows multiple possible relations between conditional probabilities and p-values. If we treat y_{e_i} as the binding affinity (log ratio between two channels), and the measurement p-value as the p-value for y_{e_i}, then $P(y_{e_i} \geq \hat{y}_{e_i}|x_{e_i} = 0) = \hat{p}$ by definition ($\hat{y}_{e_i}$ and $\hat{p}$ are empirical values of affinity and p-value). If we view y_{e_i} as the reported p-value itself, then $P(y_{e_i}|x_{e_i} = 0) = 1$ due to the definition of p-values. Suppose there is a test statistic T and the p-value $p = \hat{p}$ of observing $T = \hat{T}$ is defined as $\hat{p} \equiv Pr(T \geq \hat{T}|H_0)$ under the null hypothesis H_0. There is a monotonic relation between T and p. Treating p as a random variable, the cumulative distribution of p then becomes

$$F(\hat{p}) \equiv Pr(p \leq \hat{p}|H_0) = Pr(T \geq \hat{T}|H_0) = \hat{p}. \tag{3.3}$$

Thus p has a uniform distribution in $[0, 1]$ and the probability density function is $P(\hat{p} \leq p \leq \hat{p} + dp|H_0) = 1 \cdot dp$. The condition $x_{e_i} = 0$ corresponds to the null model since the control experiments are done by excluding protein-DNA bindings. Therefore, $P(y_{e_i}|x_{e_i} = 0) = 1$.

In both cases, however, computing $P(y_{e_i}|x_{e_i} = 1)$ is more problematic. It requires us to build an alternative model of measurements: what is the distribution of log ratios or p-values when there is a protein-DNA interaction? Since we do not know whether an interaction exists a priori, it is difficult to evaluate this probability from empirical data. Moreover, even if we can evaluate the probability from some known interactions, the results can not necessarily be extrapolated to other interactions. One remedy is to substitute $P(y_{e_i}|x_{e_i} = 1)$ with an ad-hoc distribution. If y_{e_i} is the p-value, then we expect the computed p-value is small if the interaction occurs. Hence $P(y_{e_i}|x_{e_i} = 1)$ can be modeled as a decreasing function like an exponential distribution ([135]). However, the choice of the distribution is arbitrary and does not have a solid basis.

We approach this problem by interpreting the empirical p-value as the p-value of the log likelihood ratio appeared in equation 3.2. Under certain regularity conditions

about the likelihood functions, the log likelihood ratio under the null hypothesis has an asymptotic χ^2 distribution. Therefore, we can convert the empirical p-value into the log likelihood ratio.

We model the null hypothesis H_0 and the alternative hypothesis H_1 as parametric families of probability distributions of random variables z with parameters θ: $H_0 : \theta \in \Omega_0, H_1 : \theta \in \Omega_1$. We assume the null hypothesis is a restricted subclass of the alternative hypothesis: $\Omega_0 \subseteq \Omega_1$. In our case, the non-binding model is subsumed to the binding model since it can be conceived as a special case of very weak bindings. The binding model has one extra degree of freedom compared to the non-binding model, namely the affinity or strength of binding. By applying the asymptotic theory of model selection, we are able to evaluate the asymptotic likelihood ratio without specifying the parameter model classes Ω_0 and Ω_1.

The standard procedure of testing H_1 against H_0 is to evaluate the maximum log likelihood ratio. Let $\hat{\theta}_0$ and $\hat{\theta}_1$ be the maximum likelihood estimates in Ω_0 and Ω_1. The test statistic is the maximum log likelihood ratio:

$$\mathcal{L} = 2 \log \left(\frac{P(z; \hat{\theta_1})}{P(z; \hat{\theta_0})} \right). \tag{3.4}$$

The larger $\mathcal{L}$ is, the more likely that the data is generated by H_1. We are interested in the p-value of the test:

$$p(\hat{\mathcal{L}}) = Pr(\mathcal{L} \geq \hat{\mathcal{L}} | H_0). \tag{3.5}$$

where $\hat{\mathcal{L}}$ is the empirical maximum log likelihood ratio. The p-value is difficult to evaluate since the true model classes Ω_1 and Ω_0 are often unknown. However, as the number of samples increases, $\mathcal{L}$ asymptotically approximates the χ^2 distribution regardless of the underlying distributions ([27]):

$$p(\mathcal{L}) \approx 1 - F_{\chi^2}(\mathcal{L}). \tag{3.6}$$

where $F_{\chi^2}(.)$ is the χ^2 cumulative distribution with degree of freedom = 1. We denote $\mathcal{L}$ as the empirical maximum likelihood ratio and p as the empirical p-value for the

economy of notations. Equation 3.6 establishes a one-to-one mapping between the maximum log likelihood ratio and the p-value. In the location analysis data, the p-values are reported. Hence can invert equation 3.6 and obtain the log likelihood ratio

$$\mathcal{L} = F_{\chi^2}^{-1}(1-p). \tag{3.7}$$

where $F_{\chi^2}^{-1}(.)$ is the inverse χ^2 cumulative distribution. This value is biased toward H_1 since $\Omega_1 \supseteq \Omega_0$. In other words, since we can always find a $\theta_1 \in \Omega_1$ which performs at least as well as $\hat{\theta}_0$, $\mathcal{L}$ is always ≥ 0. To remove this bias we need to take the complexity of model classes into account. Instead of the maximum log likelihood ratio in 3.4, we are interested in the marginal log likelihood ratio

$$\mathcal{L}' = 2 \log \left(\frac{P(z|H_1)}{P(z|H_0)} \right). \tag{3.8}$$

where $P(z|H_i)$ is the marginal likelihood over the parametric class Ω_i with a prior $P(\theta_i)$:

$$P(z|H_i) = \int_{\Omega_i} P(z;\theta_i) P(\theta_i) d\theta_i. \tag{3.9}$$

The marginalization is often computationally demanding. However, by applying the asymptotic theory again, this task is remarkably simplified. As the number of samples increases, the marginal likelihood approximates the maximum likelihood with a penalty term for model complexity:

$$\log P(z|H_i) \approx \log P(z;\hat{\theta}_i) - \frac{d_i}{2} \log n. \tag{3.10}$$

where d_i is the degree of freedom in Ω_i and n is the number of samples. The asymptotic approximation is independent of the prior distribution. Equation 3.10 is called Bayesian Information Criteria (BIC) developed by Schwartz ([134]).

As mentioned, Ω_1 has one more degree of freedom than Ω_0. By substituting equation 3.10 in the marginal log likelihood ratio, we obtain

$$\frac{P(z|H_1)}{P(z|H_0)} \approx e^{\frac{1}{2}\mathcal{L} - \frac{1}{2}\log n}. \tag{3.11}$$

$\mathcal{L}$ can be computed from the p-value p in equation 3.7. Thus

$$\frac{P(z|H_1)}{P(z|H_0)} \approx e^{\frac{1}{2}F^{-1}_{\chi^2}(1-p)-\frac{1}{2}\log n}. \tag{3.12}$$

This is the likelihood ratio $\frac{P(y_{e_i}|x_{e_i}=1)}{P(y_{e_i}|x_{e_i}=0)}$ in equation 3.2. Hence we are able to transform a p-value in the location analysis data into a potential function term.

The major limitation of this approach is the assumption of large sample size. In fact, the sample size of all high-throughput genomic data is very small in the statistical sense. In the location analysis data, most results have only three replicate experiments. Thus both χ^2 approximation of p-values and BIC approximation of the marginal likelihood ratio may make the estimated value substantially deviant from the real value.

3.4.2 Protein-protein interaction data

Unlike location analysis data, protein-protein interaction data in DIP are obtained from heterogeneous sources. These sources have different error properties in their experiments. The database does not annotate the confidence of each reported interaction. In fact, many articles that report these interactions contain only qualitative results. Even if error models or error estimations are provided, it will be very tedious to excavate these information embedded in thousands of papers.

Despite this deficiency, DIP annotates the number of publications which report each interaction and the types of assays (high-throughput or small-scale classical assays) in these publications. These annotations provide useful information to estimate the confidence of interactions ([31]). Intuitively, the interactions reported by classical assays are more reliable than the interactions which are only reported in a high-throughput experiment. Hence the authors in [31] specified a subset of interactions in DIP that satisfied these criteria and viewed this subset dominated by true interactions. They also generated a subset of random protein pairs and speculated that it comprised predominantly false interactions. Once the positive and negative reference sets were constructed, they devised two statistical tests on the confidence of

an arbitrary subset of putative protein-protein interactions. The Expression Profile Reliability Index (EPR) is based on an observation that interacting proteins tend to be co-regulated. They evaluated the distributions of the Euclidean distances of normalized expression profiles between pairs within each reference set. The distribution in the positive set $\rho_i(d^2)$ is tilted to small values compared to the distribution in the negative set $\rho_n(d^2)$. They defined the EPR index α_{EPR} of a subset of putative protein-protein interactions as the mixture coefficient of the two distributions:

$$\rho_{exp}(d^2) = \alpha_{EPR}\rho_i(d^2) + (1 - \alpha_{EPR})\rho_n(d^2). \tag{3.13}$$

where $\rho_{exp}(d^2)$ is the expression profile Euclidean distance distribution in the target set. This quantity was used to estimate the false discovery rate in a given set:

$$Pr((g_1, g_2) \text{ does not interact}|(g_1, g_2) \text{ appears in the subset}) \approx 1 - \alpha_{EPR}. \tag{3.14}$$

In addition to expression data, they also evaluated the quality of individual interactions by phylogenetic data. If a pair of proteins bind together and they have paralog proteins, then their paralog proteins are also likely to interact. This is based on the assumption that evolution tends to preserve non-random protein-protein interactions. They developed the Paralogs Verification Method (PVM) according to this hypothesis. For a pair of proteins, this test simply checks if there exist interactions between their paralog proteins.

EPR and PVM assess the quality (reliability) of subsets of putative protein-protein interactions categorized by different criteria. The α_{EPR} coefficient on the entire DIP dataset is about 0.5, suggesting that the false discovery rate in this set is about 50%. In contrast, $\alpha_{EPR} \approx 0.85$ on the subset of interactions confirmed by at least two studies. Thus the false discovery rate reduces to 15% in this restricted subset. PVM is a property of individual interactions instead of the whole dataset. Thus we can evaluate the false positive and false negative rates of the subset confirmed by PVM. It turns out PVM is very selective (false positive rate $\leq 5\%$) but also very insensitive (false negative rate $\approx 50\%$). Among the interactions in the positive reference set, only

half of them are confirmed by PVM. Thus the positive evidence of PVM is a strong indicator of true interactions, while the negative evidence is uninformative about true interactions. The detailed description about EPR and PVM methods can be found in [31].

Based on the results of EPR and PVM, we construct the potential function terms of protein-protein interaction data. First we define the following notations similar to protein-DNA interactions.

- $E^{pp} = \{e_i = (g_1, g_2)\}$ is the collection of protein pairs that appear in DIP database.
- $X_{E^{pp}} = \{x_{e_i} : e_i \in E^{pp}\}$ denotes the indicator variables whether these protein-protein pairs interact or not. They are observed through noisy measurements.
- $Y_{E^{pp}} = \{y_{e_i} : e_i \in E^{pp}\}$ denotes the measurements about the protein-protein bindings. They are directly reported.

The values in $Y_{E^{pp}}$ are yet to be defined. As mentioned earlier, we can divide the entire DIP dataset into subsets according to several different criteria. For each protein-protein pair e_i, the observed value y_{e_i} denotes its membership according to these categorizations. We introduce four categorizations according to early discussions.

1. $\gamma_1(e_i) = 1$ if e_i appears in the DIP database, and $\gamma_1(e_i) = 0$ otherwise. Naturally $\gamma_1(e_i) = 1$ for all protein-protein edges in the skeleton graph.
2. $\gamma_2(e_i) = 1$ if e_i is reported from multiple sources, and $\gamma_2(e_i) = 0$ otherwise.
3. $\gamma_3(e_i) = 1$ if e_i is validated in PVM, i.e., there exists protein-protein interactions in the paralogs of their end proteins.
4. $\gamma_4(e_i) = 1$ if e_i is reported in small-scale experiments, and $\gamma_4(e_i) = 0$ if it appears only in high-throughput experiments.

And we define $y_{e_i} = (\gamma_1(e_i), \gamma_2(e_i), \gamma_3(e_i), \gamma_4(e_i))$ as the vector of labels according to these categorizations. These labels are given in accompany of protein pairs in DIP database.

Our goal is to compute the likelihood ratio

$$\frac{P(y_{e_i}|x_{e_i}=1)}{P(y_{e_i}|x_{e_i}=0)} \equiv \frac{P((\gamma_1(e_i), \gamma_2(e_i), \gamma_3(e_i), \gamma_4(e_i))|x_{e_i}=1)}{P((\gamma_1(e_i), \gamma_2(e_i), \gamma_3(e_i), \gamma_4(e_i))|x_{e_i}=0)} \tag{3.15}$$

[31] reports the empirical values of the false rates in each categorization. The EPR analysis shows the false discovery rates in the entire DIP dataset and the subset of multiple confirmations are 0.5 and 0.85. Thus $P(x_{e_i} = 1|\gamma_1(y_{e_i}) = 1) = 0.5$ and $P(x_{e_i} = 1|\gamma_2(y_{e_i}) = 1) = 0.85$. The interactions verified in classical assays are treated as true interactions in their analysis, thus $P(x_{e_i} = 1|\gamma_4(y_{e_i}) = 1) = 1.0$. To avoid assigning zero values to potential functions, we set $P(x_{e_i} = 1|\gamma_4(y_{e_i}) = 1) = 1.0 - \epsilon$, where ϵ is a very small number. $P(x_{e_i} = 1|\gamma_1(y_{e_i}) = 0)$, $P(x_{e_i} = 1|\gamma_2(y_{e_i}) = 0)$ and $P(x_{e_i} = 1|\gamma_3(y_{e_i}) = 0)$ are not designated in their paper. Since knowing that a protein pair is not contained in a given dataset does not provide any information about whether it is a true interaction, we assign $x_{e_i} = 1$ and $x_{e_i} = 0$ equal probability conditioned on $\gamma_1(y_{e_i}) = 0, \gamma_2(y_{e_i}) = 0, \gamma_3(y_{e_i}) = 0$. Moreover, since we do not have prior knowledge about $P(x_{e_i} = 1)$, we assign an uninformative prior $P(x_{e_i} = 1) = P(x_{e_i} = 0) = 0.5$.

The empirical analysis of PVM gives an error estimate of γ_3. In the negative reference set, only 5% of protein pairs pass the PVM test, thus the false positive rate $P(\gamma_3(y_{e_i}) = 1|x_{e_i} = 0) = 0.05$. In the positive set, however, 50% of interactions do not have paralog interactions. Thus $P(\gamma_3(y_{e_i}) = 1|x_{e_i} = 1) = 0.5$.

The evidence from $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are combined by multiplication assuming they are independent.

$$P((\gamma_1(e_i), \gamma_2(e_i), \gamma_3(e_i), \gamma_4(e_i))|x_{e_i}) = \prod_{j=1}^{4} P(\gamma_j(e_i)|x_{e_i}). \tag{3.16}$$

By applying the Bayes law in equation 3.15,

$$\begin{aligned}\frac{P((\gamma_1(e_i),\gamma_2(e_i),\gamma_3(e_i),\gamma_4(e_i))|x_{e_i}=1)}{P((\gamma_1(e_i),\gamma_2(e_i),\gamma_3(e_i),\gamma_4(e_i))|x_{e_i}=0)} &= \frac{P(\gamma_1(e_i)|x_{e_i}=1)P(\gamma_2(e_i)|x_{e_i}=1)P(\gamma_3(e_i)|x_{e_i}=1)P(\gamma_4(e_i)|x_{e_i}=1)}{P(\gamma_1(e_i)|x_{e_i}=0)P(\gamma_2(e_i)|x_{e_i}=0)P(\gamma_3(e_i)|x_{e_i}=0)P(\gamma_4(e_i)|x_{e_i}=0)} \\ &= \frac{P^3(x_{e_i}=0)P(x_{e_i}=1|\gamma_1(e_i))P(x_{e_i}=1|\gamma_2(e_i))P(\gamma_3(e_i)|x_{e_i}=1)P(x_{e_i}=1|\gamma_4(e_i))}{P^3(x_{e_i}=1)P(x_{e_i}=0|\gamma_1(e_i))P(x_{e_i}=0|\gamma_2(e_i))P(\gamma_3(e_i)|x_{e_i}=0)P(x_{e_i}=0|\gamma_4(e_i))}.\end{aligned} \tag{3.17}$$

We substitute the empirical values of $P(x_{e_i}=1|\gamma_1(y_{e_i})=1)$, $P(x_{e_i}=1|\gamma_2(y_{e_i})=1)$, $P(x_{e_i}=1|\gamma_4(y_{e_i})=1)$, $P(\gamma_3(y_{e_i})=1|x_{e_i}=0)$, $P(\gamma_3(y_{e_i})=1|x_{e_i}=1)$ and $P(x_{e_i}=1)$ into equation 3.17, and ignore the terms $P(x_{e_i}=1|\gamma_1(y_{e_i})=0)$, $P(x_{e_i}=1|\gamma_2(y_{e_i})=0)$ $P(x_{e_i}=1|\gamma_4(y_{e_i})=0)$. The potential function term is constructed from the likelihood ratio as the protein-DNA data:

$$\phi_{e_i}(x_{e_i};y_{e_i}) = \left[\frac{P(y_{e_i}|x_{e_i}=1)}{P(y_{e_i}|x_{e_i}=0)}\right]^{x_{e_i}}. \tag{3.18}$$

3.4.3 Knock-out gene expression data

The association with knock-out expression data contains two layers. In the first layer, the expression data are tied to the *actual* knock-out effects which are unobserved. In the second layer, the actual knock-out effects are explained in terms of pathways in the physical network.

For simplicity we hypothesize that a gene deletion can generate three effects on other genes: it can up-regulate, down-regulate or have no effect on other genes. These actual effects are unobserved; instead, the log ratios of the noisy expression measurements between the mutant and the wild type are measured. Hence we need to convert the log ratio measurements into the three-state actual effects. We first define the following notations.

- $\mathcal{K}_p = \{(i,j)\}$ is the index set of significant knock-out effects in the data whose p-values are below threshold p. An index is represented as a pair (i,j) for a pairwise knock-out effect.

- $K = \{k_{ij} : (i,j) \in \mathcal{K}_p\}$ is a collection of the discrete variables of pairwise single knock-out effects whose domains are $\{-1,0,+1\}$. k_{ij} denotes the effect

of deleting g_i on gene g_j. $k_{ij} = -1$ if g_j is down-regulated, $+1$ if g_j is up-regulated, and 0 if g_j is unaffected by the knock-out.

- $\mathcal{E} = \{\mathcal{E}_{ij} : (i,j) \in \mathcal{K}_p\}$ denotes the log ratios (or ratios) of mutant versus wild type gene expression levels in two-channel microarray experiments. $\mathcal{E}_{ij}$ denotes the log ratio of g_j in $g_i\Delta$ (deleting g_i) experiment.

We need to link an actual knock-out effect k_{ij} with its measurement $\mathcal{E}_{ij}$ by a potential function term. The relation between a measurement and an actual knock-out effect is specified by an error model, and the p-values of measurements according to the error model are reported ([80]). The translation from a measurement p-value into a potential function is analogous to equation 3.12 for protein-DNA interaction data. The potential function term is proportional to the likelihood ratio

$$\phi_{ij}(k_{ij}; \mathcal{E}_{ij}) = \left[\frac{P(\mathcal{E}_{ij}|k_{ij})}{P(\mathcal{E}_{ij}|k_{ij}=0)} \right]. \tag{3.19}$$

$\phi_{ij}(k_{ij} = 0; \mathcal{E}_{ij}) = 1$ by definition. If $k_{ij} \neq 0$ and has the same sign as $\mathcal{E}_{ij}$, then the likelihood ratio can be calculated from its p-value p_{ij} as in Section 3.3.1:

$$\frac{P(\mathcal{E}_{ij}|k_{ij})}{P(\mathcal{E}_{ij}|k_{ij}=0)} \approx e^{\frac{1}{2}F^{-1}_{\chi^2}(1-\hat{p}_{ij})-\frac{1}{2}\log n}. \tag{3.20}$$

What is the likelihood ratio when k_{ij} and $\mathcal{E}_{ij}$ have opposite signs? It cannot be derived from the p-value since the alternative model in the p-value calculation assumes that the actual knock-out effect and the measurement change are in the same direction. Given a stringent threshold on the p-value of knock-out observations and our interest in significant knock-out effects only, it is very unlikely that the actual effect and the measurement have opposite signs. Thus we assign $\frac{P(\mathcal{E}_{ij}|k_{ij})}{P(\mathcal{E}_{ij}|k_{ij}=0)}$ a very small constant ϵ when k_{ij} and $\mathcal{E}_{ij}$ have opposite signs. To sum up,

$$\phi_{ij}(k_{ij}; \mathcal{E}_{ij}) = \begin{cases} e^{\frac{1}{2}F^{-1}_{\chi^2}(1-\hat{p}_{ij})-\frac{1}{2}\log n} & \text{if } k_{ij} \cdot \mathcal{E}_{ij} > 0, \\ 1 & \text{if } k_{ij} = 0, \\ \epsilon & \text{otherwise.} \end{cases} \tag{3.21}$$

The potential function terms we have introduced so far are pertaining to the evidence of single variables. "Data fusion" does not occur since this evidence are independent of each other (unless there are several data probing the same set of variables). To link these data (protein-DNA and protein-protein interactions, knock-out expression) together, we need to build potential function terms which associate the knock-out effects with attributes in the physical network model. These associations reflect the constraints of the physical network model attributes in explaining knock-out effects.

Before constructing the potential function terms for explaining knock-out data, we need to clarify what knock-out effects we plan to explain. In this chapter, we focus on the effects of single deletion mutants for the simplicity of encoding their potential functions. The cause and effect of a single knock-out effect are clear, and we can explain this effect through the pathways of molecular interactions connecting the terminal genes. Furthermore, the potential functions explaining single knockout effects are easier to implement compared to double or multiple knockout experiments. This is because we assume the perturbation along a single pathway suffices to affect the downstream gene, hence do not consider the combinatorial effects of multiple pathways. Among the single knock-out effects, we focus on significant interactions (up or down regulations) whose end genes are connected via pathways in the physical network model. We pre-select the knock-out effects by thresholding on the p-values of knock-out data and checking whether their end genes are connected by candidate paths. This restriction is for reducing the unnecessary complexity of the model. Insignificant knock-out effects which are not connected are trivially explained by the disconnection of subsystems. This explanation, however, provides little information about the underlying system. Significant knock-out effects which do not have connecting pathways of physical interactions cannot be explained by cascades of molecular interactions alone. Insignificant knock-out effects which are connected may have been suppressed for many reasons. They can be potentially explained by the physical network model.

The explanation of knock-out effects imposes constraints on the physical network model. For a path π in the skeleton graph G to to qualify in explaining k_{ij}, it must

satisfy the following conditions:

1. The end nodes of π are g_i and g_j.
2. The length of π is less than or equal to a pre-defined upper bound.
3. If intermediate genes along π are deleted, they also exhibit a knock-out effect on g_j.
4. The last edge in π is a protein-DNA interaction.
5. All edges along π are present.
6. The path has a uniform direction from g_i to g_j; in other words, there are no convergent nodes ($\rightarrow \cdot \leftarrow$) or divergent nodes ($\leftarrow \cdot \rightarrow$) along the path.
7. The aggregate sign along π – namely the product of edge signs along π – is the opposite of the sign of the knock-out effect.

The first condition manifests the assumption of using a cascade of physical interactions to explain gene regulation. The second condition excludes using unreasonably long cascades to explain knock-out effects. The optimal upper bound can be found by verifying the performance of the model with different maximum path length. The third condition requires that each interaction along π is a necessary component for regulating g_j. Thus perturbing any member along π should also affect g_j. This condition does not address the change of g_j when the deletion experiment of an intermediate gene is not available. The fourth condition reflects the current biological model of gene regulation: the last step of transcription regulation is always a protein-DNA binding from a transcription factor to a DNA promoter. The fifth condition seems trivial since edges along a path are already present in the graph. However, since the skeleton graph includes candidate interactions and the confidence of edge presence is gauged from physical data, this condition prevents us from using unlikely paths to explain knock-out interactions. The sixth condition ensures that the path has a causal interpretation. Because the path has a uniform direction, the effect of

deleting the most upstream gene can be propagated to the most downstream gene. The seventh condition ensures the actual knock-out effect is consistent with the predicted functional direction along the path. Since the deletion effect of a function is reversed (deleting an activator yields down-regulation and vice versa), the aggregate sign along π needs to be the opposite of the knock-out effect.

A knock-out effect k_{ij} is explained by the physical network model if there exists at least one path which satisfies all seven conditions. Conditions 1-4 can be verified without knowing the model attribute values, thus we can identify the connecting paths which satisfy conditions 1-4 before constructing potential function terms and carrying out inference . They are the candidate paths which can possibly explain knock-out effects. For the convenience of explanation we define the following notations. Suppose we want to explain a knock-out effect k_{ij},

- $\Pi_{ij} = \{\pi_1, \cdots, \pi_n\}$ denotes a collection of paths connecting g_i and g_j which satisfy conditions 1-4. We call them candidate paths or valid paths.

- a denotes the index of a candidate path and $\pi_a \in \Pi_{ij}$ denotes a candidate path for explaining k_{ij}.

- $E_a = \{e \in \pi_a\} = E_a^{pd} \cup E_a^{pp}$ denotes the physical interactions along π_a, where E_a^{pd} is their protein-DNA interaction edges and E_a^{pp} protein-protein interaction edges.

- $X_a = \{x_e : e \in E_a\}$ denotes the presence variables of edges along π_a.

- $S_a = \{s_e : e \in E_a\}$ denotes the sign variables of edges along π_a.

- $D_a = \{d_e : e \in E_a^{pp}\}$ denotes the direction variables of protein-protein edges along π_a.

- $\hat{D}_a = \{\hat{d}_e : e \in E_a^{pp}\}$ denotes the fixed values of protein-protein edge directions along π_a. $\hat{d}_e$ follows the path direction from g_i to g_j. For example, if π_a contains one protein-protein edge $e_1 = (g_i, g_k)$ and one protein-DNA edge $e_2 = (g_k, g_j)$,

then we set the fixed direction value of e_1 is $\hat{d}_{e_1} = +1$, indicating the edge direction is (g_i, g_k).

Conditions 5-7 are translated into the hard constraints pertaining to variables in X_a, S_a and D_a:

- $\forall e \in E_a, x_e = 1$.
- $\forall e \in E_a^{pp}, d_e = \hat{d}_e$.
- $\prod_{e \in E_a} s_e = -k_{ij}$.

The potential function encoding these conditions is expressed as

$$\psi^0_{ija}(X_a, D_a, S_a, k_{ij}) = \prod_{e \in E_a} I(x_e = 1) \cdot I(\prod_{e \in E_a} s_e = -k_{ij}) \cdot \prod_{e \in E_a^{pp}} I(d_e = \hat{d}_e). \quad (3.22)$$

where $I(.)$ is the indicator function. Notice directions of protein-DNA edges are fixed and need to satisfy condition 6 in order to be a candidate path. ψ^0_{ija} returns 1 if the conditions are satisfied and 0 otherwise. We relax the hard constraints by constructing the potential function term as follows:

$$\psi_{ija}(X_a, D_a, S_a, k_{ij}) = (1 - \epsilon)\psi^0_{ija}(X_a, D_a, S_a, k_{ij}) + \epsilon. \quad (3.23)$$

where ϵ is small number whose value is externally set. ψ_{ija} returns 1 if all the hard constraints are satisfied and returns ϵ otherwise. ϵ can be understood as the relative weight of explaining the knock-out effect k_{ij} with causes other than path π_a. Since we do not specify alternative causes, ϵ reflects our subjective belief and is a free parameter. We will show in Chapter Four that the prediction outcomes are robust against a wide range of ϵ values.

When there are multiple candidate paths connecting g_i and g_j, we require that the conditions along at least one of the paths suffice to explain k_{ij}. Translating into logical phrases, the constraints corresponding to different candidate paths are joined

by logical OR. For example, suppose two candidate paths π_a and π_b connect g_i and g_j. The potential function term of explaining k_{ij} becomes

$$\psi_{ijab}(X_a, D_a, S_a, X_b, D_b, S_b, k_{ij}) = \begin{cases} 1 & \text{if } I(\psi^0_{ija}(X_a, D_a, S_a, k_{ij}) = 1) \vee I(\psi^0_{ijb}(X_b, D_b, S_b, k_{ij}) = 1), \\ \epsilon & \text{otherwise.} \end{cases} \tag{3.24}$$

where $\bigvee$ denotes logical OR. We introduce auxiliary path selection variables in order to represent the potential function of multiple pathways with a compact form. Define the path selection variables as

- $\Sigma = \{\sigma_{ija} : k_{ij} \in K, \pi_a \in \Pi\}$ is a collection of binary (0/1) path selection variables, where Π is the set of all candidate paths in G. σ_{ija} denotes whether path π_a is an active causal explanation of the knock-out effect k_{ij}.
- $\Sigma_{ij} \subset \Sigma$ denotes the selection variables of candidate paths for explaining k_{ij}.

Physically, σ_{ija} represents whether the pathway π_a plays a regulatory role and is perturbed in $g_i\Delta$ experiment (gene i is deleted). The potential function term corresponding to a single path hence is augmented with the condition that the path is selected.

$$\psi_{ija}(X_a, S_a, k_{ij}, \sigma_{ija}) = (1-\epsilon_2)\psi^0_{ija}(X_a, D_a, S_a, k_{ij}) \cdot I(\sigma_{ija} = 1) + (\epsilon_1 - \epsilon_2) I(\sigma_{ija} = 0) + \epsilon_2. \tag{3.25}$$

It returns 1 when the path is selected and explanatory conditions are satisfied, ϵ_1 when the path is not selected, and ϵ_2 when the path is selected and the conditions are violated. We require that $1 > \epsilon_1 \gg \epsilon_2$ so that selecting a path that explains the knock-out pair is the most desirable outcome. Not selecting π_a is inferior to selecting π_a and explaining k_{ij}, but is still better than selecting π_a but not being able to explain k_{ij} with π_a. The value of ϵ_2 is immaterial so long as it is sufficiently small. ϵ_1 pertains to the *a priori* probability that a valid path π_a should be active (explain the knock-out effect).

We then construct a potential function term ψ^{OR}_{ij} to specify the condition that at least one candidate path is selected to explain k_{ij} if k_{ij} is explained. Similar to other

potential functions, ψ_{ij}^{OR} is a "soft" or "noisy" version of logical OR:

$$\psi_{ij}^{OR}(\sigma_{ij1}, \cdots, \sigma_{ij|\Pi_{ij}|}) = \epsilon + (1-\epsilon)(1 - \prod_a I(\sigma_{ija} = 0)). \tag{3.26}$$

Now we construct the potential function of a knock-out effect as the product of the potential terms corresponding to individual paths and the noisy OR term. Recall Π_{ij} is the collection of candidate paths for explaining k_{ij}. Denote $E_{ij} = \cup_{\pi_a \in \Pi_{ij}} E_a$, $X_{ij} = \cup_{\pi_a \in \Pi_{ij}} X_a$, $S_{ij} = \cup_{\pi_a \in \Pi_{ij}} S_a$, $D_{ij} = \cup_{\pi_a \in \Pi_{ij}} D_d$, and $\Sigma_{ij} = \{\sigma_{ija} : \pi_a \in \Pi_{ij}\}$. Then

$$\psi_{ij}^0(X_{ij}, S_{ij}, D_{ij}, \Sigma_{ij}, k_{ij}) = \psi^{OR}(\sigma_{ij1}, \cdots, \sigma_{ij|\Pi_{ij}|}) \cdot \prod_a \psi_{ija}(X_a, S_a, D_a, \sigma_{ija}, k_{ij}). \tag{3.27}$$

$\psi_{ij}^0(.)$ returns a relatively high value if at least one path is selected and explains the knock-out effect. It rewards the configurations where more paths are selected and the knock-out effect is explained. However, the difference between the cases when multiple paths explain k_{ij} and a single path explains k_{ij} is not large, provided that ϵ_1 in equation 3.25 is not very small. In contrast, $\psi_{ij}^0(.)$ is severely penalized when no paths are selected or selected paths cannot explain knock-out effects. This is because ϵ_2 in equation 3.25 and ϵ in equation 3.26 are close to 0.

Since we currently explain significant knock-out effects (i.e., excluding unaffected genes), we modify the potential function slightly to incorporate this choice *a priori*:

$$\psi_{ij}(X_{ij}, D_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}) \quad = \quad I(k_{ij} \neq 0)\psi_{ij}^0(X_{ij}, D_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}) + I(k_{ij} = 0). \tag{3.28}$$

$\psi_{ij}(.)$ returns a relatively high value if either there is a significant knock-out effect between g_i and g_j and the model explains this knock-out effect, or there is no significant knock-out effect between g_i and g_j.

3.5 Inference of model attributes

Each potential function term represents one constraint obtained from one or multiple sources of data. Ideally, a model configuration which conforms with the underlying biological mechanisms should be able to satisfy all these constraints simultaneously. Therefore, potential functions are combined by multiplication to form a joint likelihood function. Combining equations 3.2, 3.15, 3.19 and 3.28:

$$P(X_E, S_E, D_{E^{pp}}, K, \Sigma; Y_E, \mathcal{E}_K) \propto$$
$$\prod_{\vec{e}_i \in E^{pd}} \phi_{\vec{e}_i}(x_{\vec{e}_i}; y_{\vec{e}_i}) \cdot \prod_{\bar{e}_j \in E^{pp}} \phi_{\bar{e}_j}(x_{\bar{e}_j}; y_{\bar{e}_j}) \cdot \prod_{k_{ij} \in K} \phi_{ij}(k_{ij}; \mathcal{E}_{ij}) \cdot \prod_{k_{ij} \in K} \psi_{ij}(X_{ij}, D_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}). \tag{3.29}$$

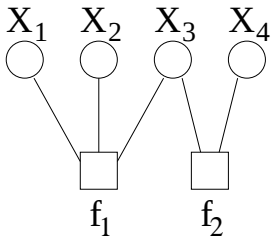
The goal of model inference is to find the configurations of variables $(X_E, S_E, D_E^{pp}, K, \Sigma)$ which maximize the joint likelihood function. There are multiple optimal configurations when the the data do not provide sufficient constraints on the model. On the other hand, the data may over-constrain part of the model so that not all constraints can be satisfied simultaneously. The violation of hard constraints is allowed because the potential functions return non-zero values for all input configurations.

In this section, we will describe the inference algorithm which obtains the optimal configurations and represents them in a concise, decomposed form. We will start with the introduction of factor graph models and two message-passing algorithms – max-product and sum-product, then describe a recursive algorithm of decomposing the model variables and finding the optimal sub-configurations.

3.5.1 Factor graph models

The joint likelihood function in equation 3.29 can be viewed as an unnormalized probability distribution of a graphical model. A probabilistic model qualifies as a graphical model if its probability density/mass function can be factorized into the terms pertaining to subsets of the random variables it models. The model is *graphical* in the sense that we can associate the factorization structure of the model with a graph. By this definition every probabilistic model is trivially a graphical model of a single term, though this expression is certainly of no interest. A graphical model

Figure 3-1: A toy example of a factor graph



is computationally useful only when there are multiple terms and the number of variables involved in each term is small.

In this thesis, we represent the likelihood function in equation 3.29 with a class of graphical models – factor graphs ([97, 173]). Like Bayesian networks or Markov random fields, factor graphs model the probability functions which can be factorized into potential function terms. For visualization and for structuring inference calculations, a factor graph can be represented as an undirected, bi-partite graph with two types of nodes: variable nodes and factor nodes. A variable node corresponds to a variable in the model, and a factor node corresponds to a potential function term. Only edges between variable and factor nodes are allowed. A variable node is adjacent to a factor node if the corresponding variable appears as an argument of the corresponding potential function. An example of a factor graph is shown in Figure 3-1. Circles stand for variables and squares stand for factors. The corresponding probability function is proportional to the product of potential functions:

$$P(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3) f_2(x_3, x_4). \tag{3.30}$$

Notice the potential function does not need to sum up to one since the normalization constant does not affect the inference results.

Factor graphs are widely used in decoding complicated error-correcting codes such as Gallager codes and turbo codes ([55, 173]). In the channel coding problem, messages are encoded to codewords according to deterministic functions (for example, parity check functions). On the other hand, codewords transmitted into a channel

are corrupted by noise thus mapped to received bits through probabilistic functions. Both deterministic and probabilistic functions can be formulated as potential functions. The problem of retrieving the transmitted messages which conform with both coding functions and channel noise becomes inferring the maximum likelihood configuration of a factor graph.

3.5.2 Max-product and sum-product algorithms

Model inference denotes evaluating the probability of some variables conditioned on the evidence of other variables. We are interested in two types of probabilities. The *marginal* probability (or the conditional marginal probability) is computed by summing over all configurations of other variables. For example,

$$P(x_1, x_2|x_3 = 1, x_4 = 0) = \sum_{(x_5,\cdots,x_n)} \frac{P(x_1, x_2, x_3 = 1, x_4 = 0, x_5, \cdots, x_n)}{P(x_3 = 1, x_4 = 0)}. \quad (3.31)$$

The *max marginal* probability (or the conditional max marginal probability) is computed by maximizing over all configurations of other variables. For example,

$$P^{max}(x_1, x_2|x_3 = 1, x_4 = 0) = \max_{(x_5,\cdots,x_n)} \frac{P(x_1, x_2, x_3 = 1, x_4 = 0, x_5, \cdots, x_n)}{P(x_3 = 1, x_4 = 0)}. \quad (3.32)$$

Marginal or max-marginal probabilities of single variables are in general easier to compute and also useful. Formally, the marginal probability of a single variable can be written as

$$P(x) = \sum_{\sim\{x\}} P(x, U\backslash\{x\}). \quad (3.33)$$

where $\sim \{x\}$ denotes the summation is taken over the configurations of all variables excluding the target variable x. Similarly, the max-marginal probability of a single variable is

$$P(x) = \max_{\sim\{x\}} P(x, U\backslash\{x\}). \quad (3.34)$$

The time complexity for model inference seems to be exponential for there are an exponential number of configurations. However, the marginal or max marginal prob-

abilities can be approximated efficiently in graphical models due to their factorized structure. Here we consider the marginal probability of a single variable. If the the graph structure corresponding to the likelihood function does not have loops, then we can order the variables in the expression of the likelihood function and carry out the summation in a sequence. The time complexity becomes polynomial instead of exponential. For example, consider the model in equation 3.30.

$$P(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3) f_2(x_3, x_4) \tag{3.35}$$

and evaluating the marginal probability $P(x_4)$. We operate summations in the following order:

$$\begin{aligned} &P(x_4) \propto \textstyle\sum_{x_1,x_2,x_3} P(x_1, x_2, x_3, x_4) = \sum_{x_1,x_2,x_3} f_1(x_1, x_2, x_3) f_2(x_3, x_4) \\ &= \textstyle\sum_{\sim\{x_4\}} [f_2(x_3, x_4) \sum_{\sim\{x_3\}} f_1(x_1, x_2, x_3)]. \end{aligned} \tag{3.36}$$

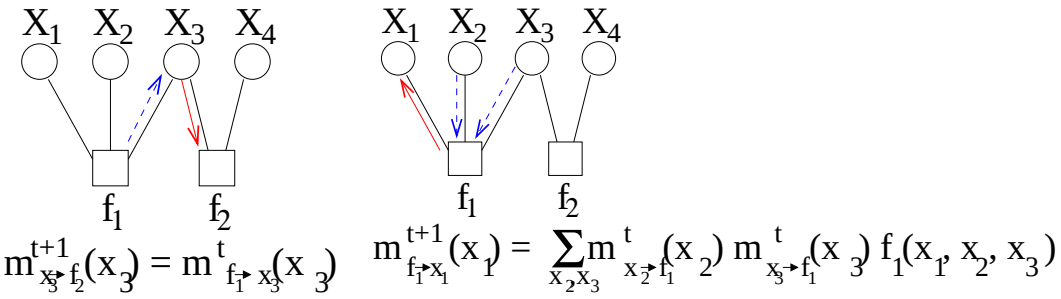
where $\sim \{x_i\}$ denotes the collection of all variables except x_i. The summation in the inner term (over $\sim \{x_3\}$) is carried out first. The result $\sum_{\sim\{x_3\}} f_1(x_1, x_2, x_3)$ is a function of x_3. This function multiplies with f_2 and is summed over $\sim \{x_4\}$. Hence the summations are carried out over (x_1, x_2) and x_3 instead over (x_1, x_2, x_3).

This property forms the basis of the well-known message passing algorithms for graphical models. There are several variants of these algorithms on different types of graphical models. We implement the *sum-product* and *max-product* algorithms for factor graph models ([97]). We briefly describe the sum-product and max-product algorithms as follows. A detailed discussion of the algorithm can be found in [97].

Define a *message* as a function associated with an edge in the factor graph. It takes the variable node of the corresponding edge as the argument. For instance, $m_{\phi_1 \to x_1}(x_1)$ is a message from ϕ_1 to x_1 and is a function of x_1. Notice messages are directed, hence $m_{\phi_1 \to x_1}(x_1) \neq m_{x_1 \to \phi_1}(x_1)$. A message must be either from a variable to a factor or vice versa due to the bi-partite property of the factor graph.

Intuitively, an initial message reflects the local property (a potential function) of a factor graph model. To generate a globally consistent solution, each local message

Figure 3-2: Message updates in a toy factor graph



must be propagated throughout the graph. The propagation of messages can be achieved by updating all messages simultaneously. At each step, messages are updated according to messages emanating from adjacent nodes at previous step. There are two rules of updating messages. An updated message from a variable x to a factor f is simply the product of all messages incident to x except from f:

$$m_{x \to f}(x) = \prod_{f_i \in N(x) \backslash \{f\}} m_{f_i \to x}(x). \tag{3.37}$$

where $N(x)$ denotes the neighboring factor nodes of x. The update of a variable $\to$ factor message is illustrated in Figure 3-2.

An updated message from a factor f to a variable x, in contrast, is the product of the factor function and incident messages marginalized over $N(f) \backslash \{x\}$:

$$m_{f \to x}(x) = \sum_{N(f) \backslash \{x\}} f(x, N(f) \backslash \{x\}) \prod_{x_i \in N(f) \backslash \{x\}} m_{x_i \to f}(x_i). \tag{3.38}$$

where $N(f)$ denotes the neighboring variable nodes of f, i.e., arguments in f. The update of a factor $\to$ variable message is also illustrated in Figure 3-2.

We consider a connected factor graph where every node is reachable from every other node. When the graph has no cycles (namely a tree), then the message passing algorithm yields exact estimations of marginal probabilities. The sum-product algorithm initializes messages emanating from leaf nodes and iteratively updates messages. It stops when all initial messages emanating from leaf nodes have been

Figure 3-3: Sum-product algorithm

1. Initialize messages emanating from leaf nodes. An initial message from a variable node x is $m^0_{x\to f}(x) = 1$ if x is not conditioned to a fixed value, and $m^0_{x\to f}(x) = I(x = c)$ if x is fixed to value c. An initial message from a factor node f is $m^0_{f\to x}(x) = \sum_{\sim\{x\}} f(x, \sim\{x\})$.

2. Iteratively pass messages to their neighbors (except the nodes where they come from) and update messages according to equations 3.37 and 3.38.

3. Terminate when all messages converge to fixed functions. This is equivalent to the condition that all initial messages have reached every node if the graph is a tree.

4. The belief function of a variable x is the product of its incident messages:

$$b(x) = \prod_{f_i \in N(x)} m_{f_i\to x}(x). \tag{3.39}$$

propagated to every other node in the graph. The marginal probability – the belief function – of a variable is proportional to the product of messages incident to the variable. The procedures of the sum-product are described in Figure 3-3.

Max-product algorithm evaluates the max-marginal probabilities of single variables. It is identical to sum-product algorithm except summation in the message update rule in equation 3.40 is replaced by maximization:

$$m_{f\to x}(x) = \max_{N(f)\backslash\{x\}} f(x, N(f)\backslash\{x\}) \prod_{x_i \in N(f)\backslash\{x\}} m_{x_i\to f}(x_i). \tag{3.40}$$

Sum-product and max-product algorithms can efficiently estimate marginal and max marginal probabilities. The number of message updates at each iteration is proportional to the number of edges in the factor graph, which is linear in the number of variables for a tree. The number of iterations is also proportional to the number of edges for a tree. The computational bottleneck is the sum or maximization in equations 3.38 and 3.40. In the worst case, the running time is exponential in the maximum number of variables in potential functions. Hence the time complexity is $O(nm2^C)$ for each iteration, where n is the number of variables, m is the number of

potential functions, and C is the maximum size of variables in potential functions. Similarly, the space complexity is $O(m2^C)$ since each factor takes at most $O(2^C)$ space if it is saved as a lookup table. The number of iterations for convergence on a loopy graph is difficult to estimate. The algorithm may not converge in the worst case.

The running time and space required in our problem are much smaller than these upper bounds because the potential functions have simple structures. The potential functions of knock-out explanation are relaxed logical expressions and need not be saved as lookup tables. The sum/maximization step in equations 3.38 can be evaluated by considering only a few scenarios without visiting all configurations. The potential functions of physical or functional data observation are lookup tables, but each potential function has only one variable. Thus max and sum product algorithms can be efficiently implemented. The simplification of the sum/maximization step in these algorithms is discussed in the Appendix.

The primary limitation for sum-product and max-product algorithms is the requirement for the loopless graph structure. Both algorithms yield exact evaluations of marginal or max-marginal probabilities if the factor graph is a tree ([97]). If a factor graph contains cycles, then messages may circulate around some loops, and the information of one potential function may be utilized multiple times. The converged solution is no longer the exact estimate of marginal or max marginal probabilities. In the worst case, messages may oscillate between several functions and the algorithm does not terminate. One can easily see that the factor graph becomes "loopy" when two potential functions share more than one variable. Therefore, the factor graph constructed from the joint likelihood function (equation 3.29) is guaranteed to contain loops.

Resolving the inference problems on loopy graphs has attracted broad attention in the machine learning community. We will not give a comprehensive overview in this thesis but only introduce two approaches. Generalized belief propagation ([173]) is an extension of the standard belief propagation algorithm. The algorithm divides the graph into (possibly overlapped) clusters and further divides each cluster into smaller units. One can construct a meta-graph specifying the relations between clusters and

their subunits. Each node corresponds to a cluster or a subunit and each edge denotes a subsume relation. This decomposition yields a hierarchical structure. Messages are passed between the meta-nodes in this extended graph. Since the extended graph is a tree, the belief propagation is guaranteed to converge. The generalized belief propagation yields a higher order approximation than the standard belief propagation. The crux of this algorithm relies on a proper decomposition of the original graph into clusters. Currently there are no systematic ways of performing this task.

The other approach estimates the exact max marginal (max a posteriori) probability by applying tree decomposition of an arbitrary graph ([164]). The probability function of a graphical model with fixed parameters can be represented as a convex combination of the probability functions of all spanning trees for the graph. The goal is to find a variable configuration which maximizes the likelihood function with the fixed parameters. This has been shown equivalent to finding the parameters for each tree and the weighting vector of trees, such that the convex combination of tree distributions is equal to the given distribution, and the intersection of optimal configurations for each tree distribution is non-empty. The authors expressed a tree distribution as the product of max marginal functions of nodes and edges, and operated the optimization problem on the max marginal functions. The transformed problem can be solved by tree reweighted max-product algorithm.

Although these algorithms give better approximations or exact solutions under certain conditions, their computations can be cumbersome, especially for large-scale problems. In practice, the standard belief propagation algorithms are often applied to large loopy graphs for computational efficiency. Empirical studies report good performance in various problems such as decoding complex error-correcting codes ([55]). In this thesis, we apply standard max-product and sum-product algorithms on the joint likelihood function equation 3.29. The inference results indicate that they satisfy the constraints from data and reflect meaningful biological subnetworks.

3.5.3 Recursive algorithms of inferring optimal configurations

Max marginal probabilities of single variables immediately yield the optimal solution if there is a unique optimal configuration. Suppose $(x_1, \cdots, x_n) = (\hat{x}_1, \cdots, \hat{x}_n)$ is the unique MAP (Maximum a Posteriori) configuration. It is clear that

$$\arg\max_{x_1} P^{max}(x_1) = \arg\max_{x_1} \max_{(x_2,\cdots,x_n)} P(x_1, x_2, \cdots, x_n) = \hat{x}_1 \tag{3.41}$$

and the equality holds for every variable. Thus the MAP configuration is simply the arg max of the max marginal probability of each variable. Equation 3.41 also holds when there are multiple MAP configurations but they all have the same value on x_1. Consequently, if the max marginal probability of a variable has a unique arg max value, then all the MAP configurations have the same value arg max value on this variable.

Problems arise when there are multiple arg max values of max marginal probabilities. Suppose there are two MAP configurations $\hat{\mathbf{x}}^a$ and $\hat{\mathbf{x}}^b$ whose values on x_1 differ: $\hat{x}_1^a = 0$ and $\hat{x}_1^b = 1$. Then

$$\begin{aligned} &P^{max}(x_1 = 0) = \max_{(x_2,\cdots,x_n)} P(x_1 = 0, x_2, \cdots, x_n) = P(\mathbf{x} = \hat{\mathbf{x}}^a) \\ &= P(\mathbf{x} = \hat{\mathbf{x}}^b) = \max_{(x_2,\cdots,x_n)} P(x_1 = 1, x_2, \cdots, x_n) = P^{max}(x_1 = 1). \end{aligned} \tag{3.42}$$

The first and last equalities are the definition of max marginal probabilities. The second and fourth equalities state that maximizations at $x_1 = 0$ and $x_1 = 1$ occur at $\hat{\mathbf{x}}^a$ and $\hat{\mathbf{x}}^b$ respectively. The third equality holds because $\hat{\mathbf{x}}^a$ and $\hat{\mathbf{x}}^b$ are degenerate MAP configurations. Consequently, degeneracy occurs on a variable whenever its max marginal probability has multiple optimal values.

Although degenerate max marginals indicate the degeneracy of MAP configurations on these variables, they do not suffice to retrieve these configurations. Consider a simple example that $(x_1, x_2) = (0, 1), (1, 0)$ are the degenerate MAP configurations. Degeneracy occurs at both x_1 and x_2, but not all the four possible configurations are

MAP configurations.

Since max-product is an approximation algorithm for loopy graphs, the inferred belief functions may not be identical to the true max marginal probabilities. However, the inferred belief functions are also likely to have degenerate max arguments. Therefore, the degeneracy problem also exists in approximated max marginal probabilities.

Degeneracy calls for a recursive algorithm to identify all optimal configurations. We select a degenerate variable and branch out the procedure. In each branch we fix the selected variable to one of its optimal values. Conditioned on the extra evidence of the newly fixed variable, we run the max-product algorithm again to compute the conditional max marginal probabilities. The degenerate variables which are tightly coupled with the newly fixed variable should be subsequently fixed. We then choose another variable which remains degenerate and repeat the same procedure recursively. The algorithm terminates when all variables in all branches are fixed. We describe the procedures of the algorithm in Figure 3-4.

The execution of the algorithm is illustrated by a simple example in Figure 3-5. The aggregate signs from g_1 to g_4, g_5, g_6 are positive, while individual edge signs are not fixed. The recursive procedure branches out as a decision tree. At the first level the sign of the first edge (s_1) is set, and at the second level the sign of the second edge (s_2) is set. The signs of the remaining edges are fixed once these two edge signs are set. A leaf node is reached when all variables are fixed according to values specified in the path from root to this leaf.

The choice of externally fixing a degenerate variable in each step may affect the resulting configurations. In the example stated in Figure 3-5, if we choose to first fix $s_3 = +1$ and then fix $s_4 = -1$, then the remaining edge signs cannot explain both knock-out effects of $(g_1, g_4, -)$ and $(g_1, g_5, -)$ simultaneously. Thus not all configurations along each branch of the tree are optimal. We employ a heuristic of choosing the degenerate variable which connects to the greatest number of undetermined variables via factor nodes. In the same toy example, s_1 and s_2 have higher priorities than s_3, s_4, s_5 in the beginning because s_1 and s_2 have 3 one-step neighbors in the factor

Figure 3-4: Recursive algorithm for obtaining all MAP configurations

1. Run max-product algorithm and fix the variables which yield unique optimal values. Let X_F denote the fixed variables and V_F denote the configuration of the fixed variables.
2. Generate a root node with configuration V_F on variables X_F and all other variables undetermined.
3. Recurse on the following steps until all branches of the tree terminate.
   (a) Select a variable x_i which is not fixed according to the configuration designated by the path from the root to the current node. Denote this configuration as V_C and the corresponding fixed variables as X_C.
   (b) Find the optimal values of x by running the max-product algorithm. Denote the optimal values as $v_1, \cdots, v_m$.
   (c) For each optimal value v_j,
      i. Branch out a child node. Set $x_i = v_j$.
      ii. Run max-product algorithm to estimate $P^{max}(x|x_i = v_j, X_C = V_C)$.
      iii. Identify the variables which have unique max marginal probabilities. Set these variables to their arg max values.
      iv. If all variables are fixed, then terminate this branch. Otherwise recurse on step 3.

graph as opposed to s_3, s_4, s_5 with 2 one-step neighbors.

Complete enumeration of all branches in the decision tree suffers from the drawback of exponential explosion. This is the case for the large-scale data we use: there are thousands of genes but only a few measurements. Consequently, we need to revise the algorithm to avoid direct enumeration of all MAP configurations.

One approach to avoid exponential explosion is to traverse along one specific path of the decision tree instead of branching out all degenerate values in each step. However, this approach does not retrieve the degeneracy information of the model. The other approach is to exploit the modular nature of the physical network models. We decompose the entire physical network model into submodels according to the decoupling relations of constraints. The variables which have unique max marginal probabilities constitute a subset. The variable subconfigurations on this subset are

Figure 3-5: A toy example of recursively fixing variables

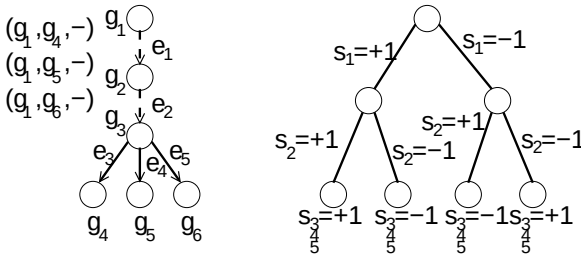
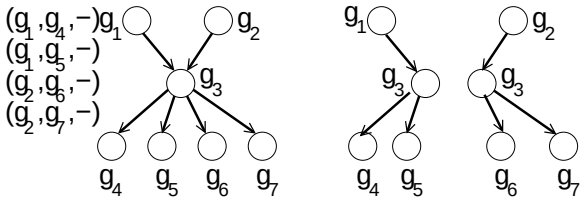


Figure 3-6: A toy example of decomposed subnetworks



invariant across all MAP configurations. The remaining variables can be further divided into subsets. Variable degeneracies across different subsets are decoupled, hence we can fix variables in one subset without affecting others. The overall MAP configurations can be expressed as the product of subconfigurations in these subsets. Each subset is small enough to enumerate all optimal subconfigurations. Therefore, we have a concise representation for the exponential number of MAP configurations.

Figure 3-6 illustrates the concept of decomposed subnetworks with a toy example. Knock-out effects $(g_1, g_4, -), (g_1, g_5, -), (g_2, g_6, +), (g_2, g_7, +)$ decompose the physical network on the left into two subnetworks on the right. Edge signs in one subnetwork are free to adjust without affecting edge signs of the other. Notice the variables contained in each subnetwork are disjoint, but these subnetworks share a vertex g_3. In other words, the variable sets associated with subnetworks are a partition of model variables, but the subnetworks are not a disjoint partition of the physical network.

How do we identify these variable subsets? We can extract the dependencies of degenerate variables when running the recursive max-product algorithm, and reconstruct the subsystems from these dependency relations. We define independent

Figure 3-7: Recursive algorithm for decomposing MAP configurations

1. Find variables $X_I = \{x_i : i \in I\}$ such that $P^{max}(x_i)$ has a unique max value at $\hat{x}_i$ for each x_i. V_I is the invariant subconfiguration of the model.
2. Recurse on the following steps until all variables are fixed.
   (a) Select an x_i which is not yet fixed.
   (b) Find one optimal value of x_i denoted by v_i. Set $x_i = v_i$.
   (c) Run max-product algorithm to compute $P^{max}(x|x_i = v_i)$.
   (d) $\forall x_k$ which is not yet fixed and $v = \arg\max_{x_k} P^{max}(x_k|x_i = v_i)$ is unique, establish a relation $x_i \succ x_k$.
   (e) $\forall x_k$ whose value is determined at current iteration, identify the independent variables x_j which have been externally set at previous steps, and which participate in the potential functions that contribute to determine the value of x_k. Establish the dependency relations $x_j \succ x_k$.
   (f) Recurse to 2.
3. Construct a graph G_d of degenerate variables according to binary relation $\succ$.
4. Identify connected components in G_d. The subsets $G_1, \cdots, G_k$ are vertex sets of connected components.

variables as the ones which are externally fixed during the recursion, and variables affected by an independent variable as the ones whose values are determined by the (approximate) max marginal probabilities conditioned on the evidence of fixing the independent variable. Clearly, an independent variable and all its affected variables belong to the same subsystem, for their degenerate MAP configuration values are coupled. In addition, the unique determination of affected variables may also depend on the independent variables fixed at previous steps. This can be checked by whether the independent variable appears in the potential functions which contribute to determine the affected variable. Back to the toy example in Figure 3-5, s_3 depends on both s_1 and s_2 because they appear in the same potential function explaining the knock-out effect $(g_1, g_4, -)$. After all dependency relations are established, we assign the variables which have dependency relations in the same subsystem. The algorithm of decomposing subsystems is described in Figure 3-7.

$x \succ y$ is an asymmetric relation denoting variable x affects variable when running the recursive algorithm to fix their values. The output of algorithm 3-7 is a partition of variables in the physical network model. The subnetworks corresponding to the partition of variables can be easily constructed. We can apply algorithm 3-4 on each subset to enumerate all optimal subconfigurations. Notice the subnetworks induced by different subsets may share nodes but not edges in the physical network, for variables are associated with edges or paths.

There are several issues associated with the algorithm described in 3-7. The order of selecting externally fixed variables may affect the quality of the solution: one order of fixing variables may explain more knock-out effects than others. On the other hand, the dependency relations described above may depend the values of an externally fixed variable. This property will make the decision tree induced by dependency relations contain different sets of variables at different branches. Therefore, it would be insufficient to construct the decision tree by traversing a specific branch as described in 3-7. This property may not hold when externally fixed variables are edge directions. For example, two opposite directions of a protein-protein interaction may explain two different sets of knockout effects. Therefore, the variables associated with each set of knockout effects are different. Practically, this problem seems to be a minor issue for degeneracy occurs predominantly on edge signs.

3.6 Comparison with Bayesian network models

The formulation of physical network models resembles to a large extent Bayesian networks for gene expression analysis. In fact, we can transform a factor graph into a Bayesian network that represents the same joint probability distribution over the random variables in the model. One way of transforming a factor graph into a Bayesian network is as follows. Augment the set of variables in a factor graph with a set of binary variables corresponding to factors in the model. For example, if the factor graph has a joint likelihood function $P(x_1, x_2, x_3) \propto \phi_1(x_1, x_2)\phi_2(x_2, x_3)$, then define evidence variables v_1 and v_2 corresponding to factors ϕ_1 and ϕ_2. Set a uniform and

independent prior of the variables in the factor graph. The conditional probability over original variables is proportional to the potential function of the factor. For example, $P(v_1 = 1|x_1, x_2) \propto \phi_1(x_1, x_2)$. The joint distribution $P(x_1, x_2, x_3, v_1, v_2) = P(x_1)P(x_2)P(x_3)P(v_1|x_1, x_2)P(v_2|x_2, x_3)$ defines a Bayesian network on the augmented variable set. Finding the MAP configuration of the factor graph is equivalent to finding the MAP configuration of the likelihood function of the Bayesian network with augmented variables fixed to 1: $P(x_1, x_2, x_3, v_1 = 1, v_2 = 1)$.

Despite the similarity between factor graphs and Bayesian networks, the physical network models differ significantly from current approaches of using Bayesian networks to model gene expression data, for instance, [71, 136, 123]. A primary difference is the meaning of variables in the model. Most current works of using Bayesian network models on gene regulatory systems choose variables which are directly linked to data, for example, gene expression levels and the presence or absence of a protein-binding promoter. The structural and functional properties of the network have to be extracted from the learned models. For example, the *causal order* between two genes can be revealed by checking whether models with the two genes in a specific causal order yield better scores than other models. In contrast, physical network models directly model these properties as variables such as the presence and direction of an edge. Notice that this difference does not apply to Bayesian network models in general, since we can construct a Bayesian network model over edge directions, edge signs and other attributes as stated in the previous paragraph. It only applies to a typical use of Bayesian networks for modeling the dependencies of gene expression levels (possibly with other measurement related variables such as motif presence).

The difference in model representation also leads to different computational treatments. Model selection is required in order to learn the Bayesian network structure. Random sampling or greedy search is often needed since it is infeasible to exhaust all possible models of a moderate size. In contrast, since the structure of physical network models is pre-determined by the skeleton network, learning the model involves inferring the configurations of variables which in turn determine the molecular interaction networks.

Furthermore, since physical network models encode specific hypothesis about the physical interaction aspects of gene regulation, the modeling results are easy to interpret. In contrast, current Bayesian network models for gene expression capture the statistical dependence of gene expression data. Each approach has its pros and cons. Physical network models have a clear interpretation and are linked to the physical processes of gene regulation, but their power of discovering novel information hidden in the data is limited since they are built on specific hypothesis. Bayesian network models can capture more general relations by learning the statistical dependencies between variables, yet the learned properties may arise from complex underlying physical processes thus lack clear interpretations.

Finally, although it is possible to express the constraints from the physical and functional data as priors in Bayesian networks (for example, [72]), it is non-trivial to find a concise prior representation of various constraints on model structure. For example, to impose the constraint on the direction of a pathway, we would need to assign higher probabilities to all the network structures whose directions along this pathway are consistent with the constraint. Therefore, expressing these constraints directly as potential functions is mathematically more convenient.

Chapter 4

Empirical Analysis of Physical Network Models

We discussed in Chapter Three the method of integrating the data of physical interactions and knock-out expression in the physical network model. In this chapter, we will apply the modeling framework to three high-throughput datasets and analyze the inference results. We choose *S. cerevisiae* (budding yeasts) as the model organism due to the rich datasets and knowledge about them. Three datasets are employed in the model: high-throughput chromatin-IP data of protein-DNA interactions ([100]), protein-protein interactions pulled out from the literature (DIP database), and the mRNA expression of knock-out experiments ([80]).

The skeleton graph of likely physical interactions is constructed from protein-DNA and protein-protein interaction datasets. We threshold on the p-values of the CHIP-chip data and extract the protein-DNA interactions whose p-values are below the threshold. To reduce false positives we set a stringent threshold (0.001) on the CHIP-chip data. Later we will relax the threshold and study the robustness of inferred results. The protein-protein interaction data is already a list of protein pairs thus can be directly incorporated in the skeleton graph. We also extract significant pairwise knock-out effects by thresholding their p-values in the knock-out gene expression data. The threshold of knock-out data is set to be 0.02. The robustness of inferred results by varying the knock-out data threshold will also be discussed.

We first focus on a small subnetwork involved in the yeast mating pathway as a specific example of our modeling framework. We choose this subnetwork for three reasons. First, the size of the subnetwork is small enough such that we can examine the inference results with more involved validation methods such as cross validations on the predicted knock-out interactions. Second, the yeast mating pathway is well constrained by the given data: about one third of genes in this subnetwork have been deleted in the Rosetta data. Third, we can verify the biological significance of inference results by refering to the rich knowledge about the mating pathway.

We run the inference algorithms described in Chapter Three to find the optimal annotations according to the data. We find the MAP (maximum a posteriori) models can explain a very high fraction of knockout effects in the data. This result, similar to the training errors in supervised learning tasks, indicates the physical network models have sufficient expressive power to fit the data. Nevertheless, fitting the data with low training errors does not necessarily validate the model since overfitting is possible. We validate the physical network models from the following perspectives. First, by applying cross validation tests, we show that the physical model can not only explain the existing data, but also predict the direction of changes of a knock-out effect. Second, we demonstrate that the predictive ability of the models is robust against various choices of values for the parameters and the addition of random edges to the skeleton graph. Third, we compare the inferred models to the knowledge about the mating pathway and verify that the inferred properties reflect the functions of genes along those pathways.

After analyzing this subnetwork, we apply the modeling framework to the genome-wide datasets. Quantitatively evaluating the predictive power of models by cross validation and robustness tests is too expensive for models at genomic scale. Therefore, we focus our validation on comparing the inferred subnetworks with the knowledge of yeast biology. We apply the recursive algorithm introduced in Section 3.4.3 to extract decomposed subnetworks whose configurations vary independently. We then validate the subnetworks by examining whether they contain known pathways according to previous studies and whether they are enriched with genes belonging to

specific functional categories.

4.1 Mating response pathways

4.1.1 Mechanisms of mating response pathways

We give a coarse introduction about the yeast mating response pathway in this section. The content in this section is mainly excerpted from [133] and [176]. More detailed and comprehensive reviews are covered in the texts of these references.

The life history of yeasts contains the stages with single chromosomes (haploids) and chromosome pairs (diploids). The transformation from diploids to haploids is achieved by meiosis, and the conversion from haploids to diploids is carried out by the mating process. There are two mating types in haploid cells – MATa and MATα. A haploid cell fuses itself with its complementary mating type during the mating process. The mating process is triggered when a haploid cell detects pheromone molecules secreted by its complementary mating type in the surrounding. Various physiological processes start upon the pheromone detection, for instance, cytokinesis against the pheromone gradient, cell polarization, cell cycle arrest, and so on. Therefore, pheromone response plays a critical role in yeast mating processes.

Pheromone response is carried out by a signal transduction pathway from the cellular membrane to the nucleus. Pheromones are bound by receptors encoded by Ste2 in MATa and Ste3 in MATα embedded on the cellular membrane. The receptor interacts with a G-protein complex of three components: $G_\alpha, G_\beta, G_\gamma$. They are encoded by genes Gpa1, Ste4 and Ste18 respectively. Pheromone binding on the receptor results in the exchange of GDP for GTP and dissociation of G_α from $G_{\beta\gamma}$, which triggers the cascade of phosphorylations. The G-protein physically interacts with a scaffold protein Ste5, in which a cascade of phosphorylations take place. Ste20 encodes a protein kinase which phosphorylates Ste11. The cascade Ste11 $\rightarrow$ Ste7 $\rightarrow$ Fus3 is the classic mitogen-activated protein kinase (MAPK) pathway. Each protein on the pathway phosphorylates the protein at the next step, which passes the signal

along the pathway. Ste7 also phosphorylates another MAP kinase Kss1. Both Fus3 and Kss1 phosphorylate Ste12, a transcription factor. Ste12 regulates genes involved in mating response and filamentous growth under stress conditions. The phosphorylation by Fus3 and Kss1 enables Ste12 to activate these two types of genes respectively. However, Fus3 and Kss1 also act in a redundant fashion. Genetic studies showed deleting Fus3 or Kss1 does not significantly affect the activation of mating response genes.

4.1.2 Quantitative analysis

We selected 46 genes involved in Ste12 related pathways. These genes were selected from two sources: 32 genes were chosen by Hartemink et al. ([72]) in their study of learning Bayesian networks by combining gene expression and location data, 14 genes were bound by Ste12 in the location data (p-value ≤ 0.001) and demonstrated significant changes in Ste12Δ experiment of the Rosetta Compendium data (p-value ≤ 0.02). Table 4.1 enlists the selected genes and their annotated functions. Notice this subset does not include all genes in the yeast mating pathway. Dig1 and Dig2, for example, are the repressors of Ste12 function but are not on the list.

By setting the p-value threshold = 0.001, 37 protein-DNA interactions are extracted from the location analysis data on this subset of genes. Instead of directly utilizing the protein-protein interaction data from DIP, we manually pulled out 30 interactions from the Yeast Knowledge Database (YPD) [1]. This is because DIP does not include transient interactions such as the binding of Fus3 and Ste12. These interactions, however, are responsible for protein modifications such as phosphorylation in signal transduction pathways. In this example, we want to demonstrate that the model inference obtains meaningful results if sufficient data are provided. Thus it is sensible to resort some external knowledge to reduce false negative interactions. Figure 4-1 demonstrates the subnetwork of physical interactions. Solid lines denote protein-DNA interactions and dash lines denote protein-protein interactions. The directions of protein-protein interactions are unspecified.

[1] https://www.incyte.com/tools/proteome/databases.jsp

Table 4.1: Selected genes in yeast mating response pathway

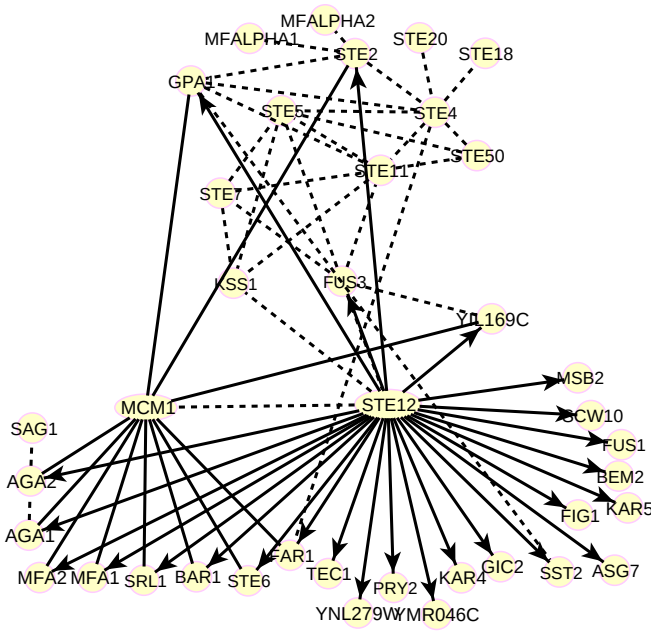
Gene	Function	Gene	Function
Ste2	α-factor G protein-coupled receptor	Ste3	a-factor G protein-coupled receptor
Gpa1	G protein subunit	Ste4	G protein subunit
Ste18	G protein subunit	Fus3	MAP kinase for mating response
Ste7	MAP kinase kinase	Ste11	MAP kinase kinase kinase
Ste5	scaffolding protein	Ste12	transcription factor
Kss1	MAP kinase for filamentous growth	Ste20	MAP kinase kinase kinase kinase
Ste50	signal transduction feedback control	Mfa1	mating a-factor
Mfa2	mating a-factor	Mfalpha1	mating α-factor
Mfalpha2	mating α-factor	Ste6	membrane transporter
Far1	cyclin-dependent kinase inhibitor	Fus1	mating cell fusion
Aga1	mating cell adhesion	Aga2	mating cell adhesion
Sag1	mating cell adhesion	Bar1	α-factor degradation
Sst2	G protein regulator	Kar3	microtube motor
Tec1	transcription activator	Mcm1	transcription activator
Sin3	transcription regulator	Tup1	repressor of RNA pol II
Snf2	global transcription activator component	Swi1	global transcription activator component
Kar4	transcription factor	Msb2	osmosensor protein
Pry2	starvation response	Fig1	calcium channel regulator
Gic2	bud emergence	Bem2	GTPase-activating protein
YGR296W	telomere maintenance	YIL169C	unknown
Asg7	inhibition of Ste4p localization	YMR046C	unknown
Kar5	homotypic nuclear fusion	Scw10	glucosidase activity
YNL279W	cell fusion	Srl1	unknown

Inferring the annotations from the mating response pathway is advantageous because it is tightly constrained by the knock-out data. 13 genes on the list of Table 4.1 are deleted in the Rosetta Compendium Dataset: Ste2, Ste4, Ste18, Fus3, Ste7, Ste11, Ste5, Ste12, Kss1, Ste20, Sst2, Sin3, Tup1. There are 149 pairwise knock-out interactions generated from these experiments on the mating pathway subset. The list of physical interactions and knock-out effects are included in the Appendix.

The existence of candidate paths between deleted and affected genes is a prerequisite for explaining a knock-out interaction. The conditions specifying candidate paths have been discussed in Section 3.3.3. Naturally, the longer path length is allowed, the more knock-out interactions are connected by candidate paths. Figure 4-2 shows the number of knock-out interactions connected by candidate paths as a function of the maximum path length. The number of connected knock-out pairs steadily grows as the maximum path length increases. The number stabilizes as the maximum path length exceeds 5, indicating that all the explainable knock-out pairs in this set are explained by paths ≤ 5 edges. Consequently, we restrict the path length ≤ 5.

There are 1291 candidate paths satisfying conditions 1-4 in Section 3.3.3. We relaxed the conditions on the paths containing Fus3 or Kss1 by not requiring signifi-

Figure 4-1: Yeast mating response subnetwork



solid: protein-DNA interaction, dash: protein-protein interaction

cant effects of deleting Fus3 or Kss1 on downstream genes. This is because we know a priori that both Fus3 and Kss1 can independently phosphorylate Ste12. Deleting either gene only affects few genes regulated by Ste12. Potential functions pertaining to physical interactions and explaining knock-out effects were constructed as described in Section 3.3. Table 4.2 summarizes the properties extracted from the joint likelihood function. We report the 103 connected knock-out pairs in the Appendix.

We applied the recursive max-product algorithm described in Section 3.4.3 to infer the optimal configurations from the joint likelihood function. Because this network is small and relatively well constrained, there are only 4 optimal configurations. The optimal configurations are shown in Figure 4-5.

We first measured the flexibility of the physical modeling approach by verifying how many of the knock-out effects can be explained. By explaining a knock-out effect we mean the following conditions are satisfied. For each MAP configuration we identified all the paths which connected the knock-out pair and were active according

Figure 4-2: Number of connected knock-out pairs

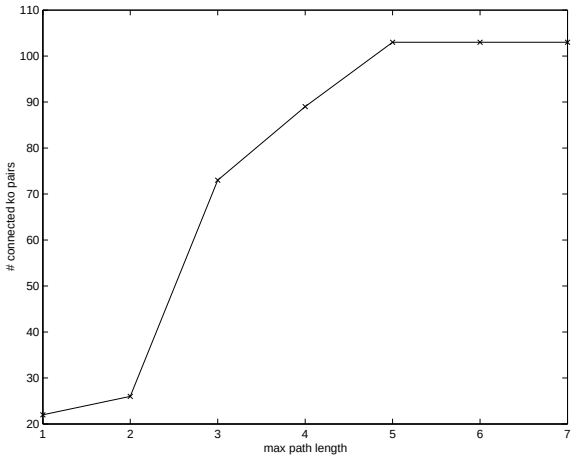


Table 4.2: Properties of the inferred physical network model

max path length	5
# candidate paths	1291
# variables	556
# constraints	576
# connected knock-out pairs	103

to the configuration ($\sigma_a = 1$). The variables along each active path (edge presence, edge directions, edge signs and the knock-out effect) must satisfy the path explanation conditions specified in Section 3.3.3. If no connecting path is active or any active path violates the explanation constraints, then this configuration does not explain the knock-out effect. We required all MAP configurations to explain the knock-out effect in order to make the knock-out effect qualified as explained. According to this stringent criterion, we compared the number of knock-out effects which were explained to the number of knock-out effects which could possibly be explained by physical interactions (namely, the number of knock-out effects connected by candidate paths). Table 4.3 shows the comparison result as a function of the maximum path length. Clearly, all possibly explainable knock-out effects are explained regardless of the upper bound of the path length. This suggests that the knock-out effects do not contradict each other, so that we can find multiple configurations of edge directions and signs

Table 4.3: Training accuracy of knock-out prediction

max path length	# connected ko pairs	# explained ko pairs
1	22	22
2	26	26
3	73	73
4	89	89
5	103	103

consistent with all of them. Moreover, the model is able to capture the flexibility of the path regulation hypothesis by demonstrating all configurations consistent with the observed data. Finally, only 21 knock-out interactions are trivially explained by paths of length one, namely the direct bindings of Ste12 on promoters. This shows the importance of incorporating pathway constraints in elucidating knock-out effects. Since the direct effects of protein-DNA interactions comprise only a small fraction of knock-out interactions, consistent explanations of other knock-out interactions rely on the inferred properties along the paths.

The results in Table 4.3 demonstrate the capability of our model inference algorithm to find the configurations fitting the constraints from the data. The high percentage of explained knock-out effects is not surprising given the fact that the model is under-constrained. The capability of fitting existing data is nevertheless not a proper gauge for a model, since we can easily construct an over-complicated model to fit the data. A better way of validating model quality is to check whether the model is capable of predicting the behavior of a new data.

We performed leave-n-out cross validation tests to evaluate the predictive accuracy of the model. We randomly held out a fixed number of knock-out pairs when constructing the joint likelihood function and running the inference algorithm. For each leave-out pair, we then examined whether all the connecting candidate paths in all MAP models predicted the same sign change consistent with its knock-out effect. This procedure was repeated many times and the average fraction of wrong predictions among the held-out examples was evaluated. For leave-one-out cross validation, the number of trials is the size of the training data (the number of connected knock-

Table 4.4: Cross validation on knock-out pairs

# held-outs	# trials	% error
1	106	2.83 %
5	200	3.5 %
20	200	5.9 %

out pairs): we held one example each time and use the remaining data to train the model. For leave-n-out tests, we performed a fixed number (200) of random trials instead of exhausting all possible n-sets as held-out sets. Table 4.4 shows the results of leave-n-out cross validation test error rates, where n equals to 1, 5 and 20. For $n > 1$, the error rate is the total number of mistakenly predicted held-out pairs divided by the total number of held-out pairs in all trials. For instance, in the leave-20-out experiments over 200 trials, there are $20 \times 200 = 4000$ held-out pairs (many of them are repeated) and 236 pairs are inaccurately predicted over the 200 trials. The reported error rate is thus $236/4000 = 5.9\%$. The knock-out pairs considered in the cross validation experiments are connected via valid paths. The low error rate in each experiment indicates that the algorithm can predict the knock-out effects with high accuracy. This is expected since there are sufficient number of knock-out experiments perturbing a small network, and the information about a knock-out interaction is distributed among multiple interactions along pathways. We may also ask whether such a high accuracy rate can be achieved by a trivial predictor, namely always predicting +1 or -1 regardless of the knock-out pairs. By checking the balance of positive and negative changes of knock-out effects in the training set, we found negative effects substantially outnumbered positive effects (103 versus 11). Therefore, by applying a -1 predictor on knock-out effects the error rate is about 10%. In spite of unbalanced ratios of positive and negative effects, our prediction results are still substantially better than the trivial prediction.

In addition to randomly holding out knock-out pairs and predicting sign changes, the predictive power of the physical network models were further elucidated by two additional tests. We first performed the leave-one-out cross validation test in terms of knock-out experiments: held out all knock-out pairs generated by one deletion

experiment in each trial. Among the 13 experiments whose deleted genes are in the subnetwork, only 9 experiments contain pairs connected via candidate paths: Ste4, Ste18, Fus3, Ste7, Ste11, Ste5, Ste12, Kss1, Sst2. Unlike the previous test results, the error rates of the hold-experiment-out tests are 100% in all trials. Notice all errors are caused by uncertainty of model configurations rather than wrong predictions. In each trial, there are multiple optimal configurations which predict knock-out effects with opposite directions. The big difference between the two cross validation outcomes is not due to the number of held-out examples, since the number of held-out knock-out pairs in each experiment is roughly identical to the leave-20-out test for random held-outs. Instead, it is due to the pathway topology of the physical network. A relatively small number of proteins (Ste2, Ste4, Ste20, Ste5, Ste7, Ste11, Fus3, Kss1, Ste12) form pathways of protein-protein interactions which end at Ste12. Ste12 in turn controls a relatively large number of genes by protein-DNA bindings. All knock-out effects are between the genes along this pathway and the downstream genes bound by Ste12. Directions and signs of edges along the main pathway are constrained by a large number of knock-out interactions on downstream genes. If some knock-out interactions are randomly removed, the remaining knock-out interactions are still likely to provide sufficient information to constrain those protein-protein edges. In contrast, if all knock-out pairs from one experiments are removed, then some edges or subpaths of these protein-protein pathways are under-constrained. Hence we can no longer predict the held-out knock-out effects.

We then evaluated the predictive power of the model by checking whether it predicted the occurrence or absence of a knock-out interactions. Ideally, a model should be able to predict not only the sign change of a knock-out effect, but also whether a knock-out interaction is significant or not. Hence the predictive accuracy should also be measured by false positive (fraction of knock-out effects which are expected to occur but do not) and false negative (fraction of knock-out effects which are not expected to occur but do) rates. Strictly speaking, our model does not predict the occurrence of knock-out effects because it only specifies necessary but not sufficient conditions for knock-out effects and it ignores the negative evidence of

knock-out interactions (a knock-out effect does not occur). We nevertheless discuss briefly the degree of match/mismatch in this sense. In a simple extension of the physical network models, we define that the model predicts the occurrence of a knock-out interaction if end genes are connected by valid paths and all edges along these paths are utilized in explaining known knock-out effects. This extension only predicts the occurrence but not the absence of a knock-out interaction. Therefore, we only discuss the false positive rate in this setting. According to this definition, if a gene is bound by Ste12 with a protein-DNA interaction, and its expression level is perturbed in any one of the 9 experiments (Ste2Δ, Ste4Δ, Ste18Δ, Fus3Δ, Ste7Δ, Ste11Δ, Ste5Δ, Ste12Δ, Kss1Δ), then it is expected to change in the experiments that delete genes downstream of the target experiment. For example, if Aga1 is down-regulated in Ste11Δ experiment, then it is expected to experience a significant change in Ste12Δ experiment. This is because we assume every intermediate gene along an explanatory pathway is a necessary component for activating/repressing the downstream gene. Hence deleting any one of them will lead to a significant change (condition 3 in Section 3.3.3). According to this criterion, 216 knock-out interactions are expected to occur among 24 genes. Since there are only 106 knock-out interactions in this set, about half of the predicted changes are missing. Some of these false positive predictions are due to the threshold of knock-out data (for instance, if we relax the threshold to 0.05, then there are 121 knock-out interactions). Some other false positive predictions can be understood from the exception of the explanatory conditions stated in Section 3.3.3 based on the knowledge of the pheromone response pathway (for instance, Fus3 and Kss1 phosphorylate Ste12 in parallel, thus a single deletion of either one does not create a significant response). There are still some false positive interactions which cannot be attributed to these simple explanations.

A more systematic way of utilizing negative evidence is to constrain network attributes (thus build potential functions) according to the lack of each pairwise knock-out effect. If genes A and B are connected by valid paths but B is unaffected in AΔ, then we may explain A's response by assuming either some edges along the paths are false positives or there are multiple redundant paths. We leave the incorporation

of negative evidence for the future work.

The randomly held-out cross-validation outcome suggests the knock-out effects are predictable with sufficient constraints in the data. This is quite encouraging for it validates the hypothesis that pathways of molecular interactions are responsible for gene regulation. However, we still need to ensure that the results are not artifacts of a particular setting of the model parameters. We have externally set the following free parameters when constructing the model and performing the inference: the thresholds on p-values for including protein-DNA and knock-out pairs, the returned values of potential functions for knock-out explanation (ϵ_1 and ϵ_2 in equation 3.25) and noisy OR (ϵ in equation 3.26), and the upper bound of the path length. The default setting of these parameters is as follows:

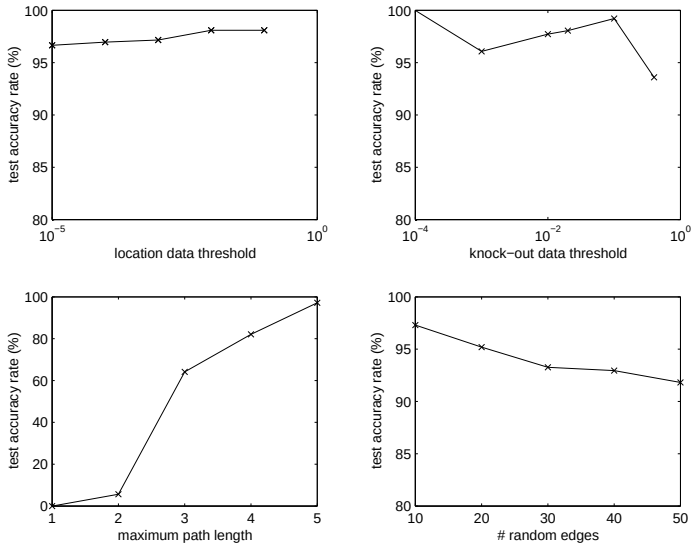
$$p^{loc}_{thre} = 0.001, p^{ko}_{thre} = 0.02, \epsilon_1 = 0.4271, \epsilon_2 = 0.0014, \epsilon = 0.001, l_{\max} = 5. \qquad (4.1)$$

A standard procedure to verify that the inference results do not vary with different parameter settings is the robustness (sensitivity) test: measuring the variation of the results by changing the values of one parameter each time. If the variation is high, then the outcome is sensitive with respect to a parameter and is less reliable.

In addition to the robustness tests against parameter settings, it is also important to test whether the predictive accuracy varies as noise are introduced in the datasets. We introduced noise by adding random protein-DNA and protein-protein interactions to the skeleton graph. The confidence of the randomly added edges is assigned to be the strongest confidence value according to the existing protein-DNA and protein-protein interaction data.

We consider the following adjustable parameters for robustness tests: the maximum length of candidate paths, thresholds on p-values of selecting candidate protein-DNA and knock-out pairs, the error probabilities used as soft constraints in the potential functions (ϵ_1 in the definition of ψ_{ija}), and the number of random edges added to the skeleton graph. Figure 4-3 shows the leave-one-out test accuracy rates across a wide range of these parameters. The test accuracy here is normalized by the number

Figure 4-3: Sensitivity analysis on test accuracy

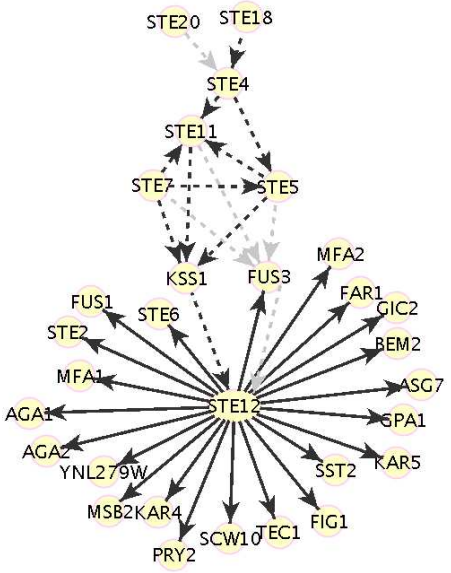


of knock-out effects that the inferred model can in principle explain (i.e., the number of connected knockout pairs). Clearly, test errors are very robust against location and knock-out p-value thresholds and as well as the potential function values. The test accuracy is above 90% across wide ranges of location and knock-out thresholds (10^{-5} to 0.1 and 10^{-4} to 0.1 respectively), and remains at 95% across the potential function value from 0.1 to 0.45. The result of the potential function value is not drawn for the test accuracy remains as constant. In contrast, test errors are very sensitive to the path length upper bound. The models constructed from short paths (length < 3) can hardly predict knock-out effects for short paths can receive very few (or no) constraints from other knock-out pairs. The test errors are also very robust against the addition of random edges: it increases only from 2% to 7% even when the number of random edges added to the skeleton graph is approximately equal to the original size of the skeleton graph.

4.1.3 Qualitative verification

Quantitative tests results suggest the inferred models can both fit existing data and predict new knock-out effects according to related constraints. To further validate the

Figure 4-4: Invariant part of yeast mating response network



solid: protein-DNA interaction, dash : protein-protein interaction.
black: positive edge, gray: negative edge.

results, we directly compared the inferred models with the knowledge about the yeast mating pathway. We first applied the max-product algorithm once and identified the variables whose values were uniquely determined by the max-marginal probabilities. Figure 4-4 shows the physical subnetwork annotated with these attributes. It is visualized using *Cytoscape*[2], a free software for visualizing gene networks. Solid lines correspond to protein-DNA and dash lines represent protein-protein interactions. The directions of protein-DNA arrows are given in the data, while the arrows (and the existence) of protein-protein edges are inferred from the model. Edge signs are color-coded with black (positive) and light grey (negative).

Comparison shows the inference results are highly overlapped with previous studies. First, all protein-DNA edges emanating from Ste12 have positive signs. This result confirms the activating role of Ste12, yet it can be directly obtained from location and knock-out data without doing the inference. Second, the inferred directions of most protein-protein interactions – including (Ste18,Ste4), (Ste4,Ste5), (Ste5,Ste11),

[2]http://www.cytoscape.org.

(Ste7,Fus3), (Ste7,Kss1), (Fus3,Ste12) and (Kss1,Ste12) – are consistent with the directions of the signal transduction pathway. Interestingly, the directions of protein-protein interactions have a variety of interpretations beyond phosphorylations. Although the binding (Ste18,Ste4) is part of a complex and cannot indeed be viewed as a pathway, the interaction with the scaffolding protein Ste5 is via the contact of Ste4. Thus we can view Ste18 → Ste4 → Ste5 as a pathway of information flow from the G-protein to the MAP kinase phosphorylation. The interaction (Ste5,Ste11) denotes the binding of MAP kinase kinase kinase Ste11 on the scaffolding protein Ste5. Its direction can be treated again as the information flow from the G-protein to the MAP kinases if we view it together with the path Ste18 → Ste4 → Ste5. The information flow follows this direction because the G-protein does not directly contact the MAP kinase kinase kinase Ste11 but via the scaffolding protein. The directions of (Ste7,Fus3), (Ste7,Kss1), (Fus3,Ste12) and (Kss1,Ste12) clearly denote the directions of MAP kinase phosphorylations. These directions are obtained because they yield consistent explanations for the depressions of the Ste12-regulated genes in deletion experiments Ste5Δ, Ste11Δ and Ste7Δ. For example, Ste4 is on all valid pathways from Ste18 to Ste12, hence only the direction from Ste18 to Ste4 is feasible.

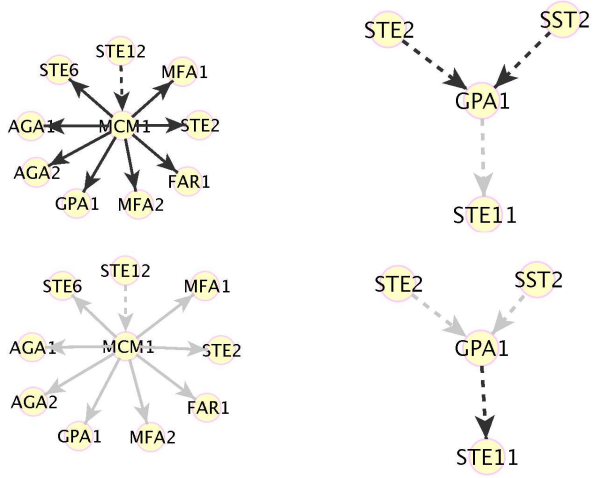
However, some inferred attribute values contradict with current biological knowledge. The inferred direction of (Ste7,Ste11) is the opposite of the MAP kinase phosphorylation from Ste11 to Ste7. The model infers the wrong direction of (Ste11,Ste7) because Ste11 also binds to Kss1 and Fus3. These bindings provide shortcuts to bypass Ste7 when generating paths to explain knock-out effects of Ste11. As the shortcuts are established, the edge (Ste11,Ste7) becomes non-critical for explaining the knock-out effects of Ste11. Furthermore, its direction may be reverted to explain other knock-out effects. For example, the effects of deleting Ste7 can be explained by the path Ste7 → Ste11 → Kss1/Fus3 → Ste12 → downstream genes. The interaction (Ste20,Ste4) does not carry a role in the signal transduction pathway, but its inferred direction is from Ste20 to Ste4 for it allows us to explain the knock-out effects on deleting Ste20. Finally, the phosphorylation of Ste11 by Ste20 is not captured in the skeleton graph, thus we cannot infer its direction.

In addition to the directions of some protein-protein edges, the inferred signs of edges (Fus3,Ste12), (Ste7,Fus3), (Ste5,Fus3) and (Ste11,Fus3) are also opposite to the actual functions of these MAP kinases. These inferred signs are negative and contradict with their functions as activations in the MAP kinase phosphorylations. The reason for the sign flip of these edges is the up-regulations of genes Pry2, Msb2, Srl1, and YMR046C in Fus3Δ experiment and down-regulations in Ste12Δ and Ste7Δ experiments. The path Ste7 $\rightarrow$ Fus3 $\rightarrow$ Ste12 $\rightarrow$ downstream genes is utilized to explain these knock-out effects. The only viable explanation is to make (Ste7,Fus3), (Fus3,Ste12) negative and (Ste12,downstream genes) positive. In fact, most down-regulated genes in Ste7Δ and Ste12Δ experiments are not affected by Fus3 deletion. The up-regulation of the small number of genes in Fus3Δ experiment is probably caused by some feedback regulation outside the scope of this subnetwork. The negative signs of these edges, however, are the best explanations to fit the given data under the simplified hypothesis about gene regulation. The negative signs of (Ste11,Fus3) and (Ste5,Fus3) are obtained for the same reason.

Although these discrepancies reduce the explanatory power of the computational model, they also provide good opportunities to verify the existing biological model and discover new biological mechanisms. The discrepancy of the edge direction (Ste11,Ste7) leads us to challenge the regulatory functions of protein-protein interactions (Ste11,Fus3) (Ste11,Kss1). These interactions are not part of the signal transduction cascade. However, their functions cannot be falsified before further validation. The discrepancy of the edge signs (Fus3,Ste12), (Ste7,Fus3), (Ste5,Fus3) and (Ste11,Fus3) suggests the existence of another pathway connecting Fus3 to the genes up-regulated in Fus3Δ experiment, or a different function of Fus3 as a repressor for the activity of Ste12. In fact, Fus3 is shown to play a inhibitory role on Ste12 ([176]). Although the double function of Fus3 is not necessarily related to the up regulations of few genes in Fus3Δ experiment, complexity of this type should be considered when interpreting the results.

We then applied the recursive max-product algorithm to decompose all MAP configurations into the product of subconfigurations in subnetworks. Since the net-

Figure 4-5: Variant part of yeast mating response network



solid: protein-DNA interaction, dash : protein-protein interaction.
black: positive edge, gray: negative edge.

work was small and well-constrained by knock-out experiments, we found only four MAP configurations which were decomposed into two subnetworks. Each subnetwork has two optimal subconfigurations. The decomposed MAP configurations of the variant part of the models are shown in Figure 4-5. All degeneracies occur at edge signs. Subnetwork 1 reflects the ambiguity of the sign of protein-protein interaction (Ste12,Mcm1). Some genes which are down-regulated in Ste12Δ experiment are jointly bound by Ste12 and Mcm1 with protein-DNA interactions. Thus their knock-out effects in Ste12Δ can be explained either by the direct protein-DNA bindings of Ste12 or the paths mediated by Mcm1. Since Mcm1Δ experiment is unavailable (in fact deleting Mcm1 is lethal for yeast cells), we speculate that both paths Ste12 $\rightarrow$ downstream genes and Ste12 $\rightarrow$ Mcm1 $\rightarrow$ downstream genes are active. The product of signs of (Ste12,Mcm1) and (Mcm1,downstream gene) is fixed while individual signs are not. Hence both $(+,+)$ and $(-,-)$ are viable. Subnetwork 2 reflects the ambiguity of the sign of protein-protein interaction (Gpa1,Ste11). This edge is essential for explaining the up-regulation of some genes in Sst2Δ and Ste2Δ experiments. The same group of genes are also down-regulated in Ste11Δ experiment, and these down

regulations can be explained by paths from Ste11 to the affected genes. Therefore, the aggregate signs along the paths from Ste11 to these genes are positive, and the aggregate signs along the paths Sst2 → Gpa1 → Ste11 and Ste2 → Gpa1 → Ste11 are negative.

4.2 Genome-wide analysis

4.2.1 Summary statistics

The analysis of the pheromone response pathways demonstrates the capability of physical network models on a small, well-constrained system with rich knowledge about the underlying biology. The primary goal of our modeling approach, however, is to uncover and annotate regulatory mechanisms on a sparsely constrained, large-scale system. Therefore, we applied the same modeling framework to the high-throughput datasets of the entire genome and attempted to interpret the biological meanings of the inference results. The analysis work is still preliminary since we have not done a comprehensive literature survey on inferred subnetworks. All the biological knowledge cited in this section is pulled out from the Yeast Protein Database (YPD [3]).

The location analysis data covers the binding profiles of 106 transcription factors on 6135 genes ([100]). By choosing the p-value threshold = 0.001, we extracted 5485 protein-DNA interactions from the data. There are 14876 protein-protein interactions of yeast proteins reported in the DIP database snapshot in May 2003. The Rosetta Compendium data contains 271 single deletion experiments (we discarded the remaining 29 experiments of double deletions and drug response) on 6295 genes. By choosing the p-value threshold = 0.02, we extracted 23766 pairwise knock-out effects.

Compared to the mating response network, the size of the entire physical network is about 300 times larger (in terms of the number of interactions), and the number of knock-out effects grows by about 150 times. However, a much smaller fraction

[3] https://www.incyte.com/proteome/Retriever/index.html

of knock-out effects can possibly be explained by cascades of molecular interactions. Table 4.5 compares the numbers and fractions of knock-out interactions connected by valid paths of different upper bounds on path length. While $> 60\%$ knock-out interactions are connected by paths ≤ 3 in the mating pathway, only about one twentieth (1091 out of 23766) are connected in the entire physical network. This small fraction may be due to the limitation of the model (a longer path length is required, or mechanisms beyond molecular interactions dominate the knock-out effects) or the limitation of the data (physical interaction data are incomplete, knock-out data are not reliable). In this thesis, we restrict path length ≤ 3 for computational efficiency purpose and focus on the knock-out pairs which can be explained by short paths.

We applied the same modeling framework to construct the joint likelihood function. The returned values of potential functions (ϵ_1 and ϵ_2 in equation 3.25 and ϵ in equation 3.26) are identical to the default settings in mating pathway analysis (equation 4.1). Table 4.5 summarizes the statistics of the genome-wide network including the number of candidate paths, the number of knock-out pairs connected by these paths, the numbers of variables and potential function terms. Compared to the model of the mating response pathways, the number of variables increases by 103 folds but the constraint number is scaled up only by 41, indicating that the entire physical network is much less constrained and more optimal configurations are expected.

We investigated the flexibility of the model to explain knock-out effects. Similar to the analysis of the mating pathway, we first applied the max-product algorithm once and checked the knock-out interactions which could be explained by the uniquely determined part of the network. The resulting model induces a much smaller network: it contains 128 genes and 142 physical interactions, and explains 194 knock-out interactions. Many other knock-out interactions may still be explained by the variant part of the model. For example, if a knock-out pair is connected by a valid path of length 2, it can be explained by all optimal edge sign configurations along the path. Since there are a large number of optimal configurations, we can no longer enumerate all optimal configurations and identify the knock-out effects explained by all of them. Instead, we randomly selected 100 optimal configurations and identified the over-

Table 4.5: Summary statistics of the large-scale network

# genes	6135
# protein-DNA interactions	5485
# protein-protein interactions	14876
# knock-out effects	23766
# valid paths	4836
# connected knock-out effects	1091
# connected knock-out effects via path length 1	125
# connected knock-out effects via path length 2	127
# connected knock-out effects via path length 3	839
# connected but isolated knock-out effects	534
# variables in the model	57484
# constraints in the model	23771
# knock-out effects explained by the invariant configuration	194
# explained knock-out effects via path length 1	121
# explained knock-out effects via path length 2	20
# explained knock-out effects via path length 3	53
# utilized pd edges in the invariant configuration	209
# utilized pp edges in the invariant configuration	55
# knock-out effects explained by one MAP configuration	987
# explained knock-out effects via path length 1	118
# explained knock-out effects via path length 2	118
# explained knock-out effects via path length 3	751
# utilized pd edges in one MAP configuration	861
# utilized pp edges in one MAP configuration	281
# genes appeared in subnetworks	773

lapped set of their explained knock-out effects. Random optimal configurations were generated by running the max-product algorithm recursively as described in Section 3.4.3. When we had to externally fix a variable to one of the degenerate values, we randomly chose one degenerate value and continue running the max-product. It turns out a large fraction of connected knock-out pairs remained explained in all sampled configurations: 984 out of 1091 knock-out interactions connected via valid paths are explained by 100 randomly selected MAP configurations. Notice many knock-out interactions are compatible with the optimal configurations but are not connected in the uniquely determined subnetwork.

We then instantiated one MAP configuration and examined the properties of its explained knock-out pairs. This configuration explains 986 knock-out interactions. The explanatory paths of 792 knock-out pairs are not constrained by any other knock-out effects. This happens when every edge along the explanatory path is utilized only once to explain knock-out effects. In this case, we have the freedom to adjust the sign configurations to fit a knock-out effect, and the inference result is less reliable.

There are 105 knock-out pairs connected by valid paths but are not explained by the MAP configuration. This is because the edge signs of explaining one knock-out effect contradict with the edge signs of explaining the other knock-out effect. For instance, in Figure 4-6 the sign of edge (Gln3,Gcn4) is known to be negative from the invariant part of the network. Since Gln3 $\rightarrow$ Gcn4 $\rightarrow$ Met4 $\rightarrow$ Met14 is the only pathway connecting Gln3 and Gcn4 to Met14, the knock-out effects (Gln3,Met14,-) and (Gcn4,Met14,-) cannot be simultaneously explained. This type of contradiction suggests one (or multiple) of the following scenarios may occur: the physical interaction data are incomplete, some knock-out gene expression data are spurious, or the knock-out effects are not caused by perturbations along molecular cascades.

As mentioned in Chapters Two and Three, the quality of protein-protein interaction data reported from high-throughput assays is often questioned. We want to know whether the differential quality of protein-protein interaction data is also reflected when combining with knock-out gene expression data in model inference. We summarize the types of protein-protein interactions involved in explaining knock-out

Figure 4-6: Contradictory knock-out effects

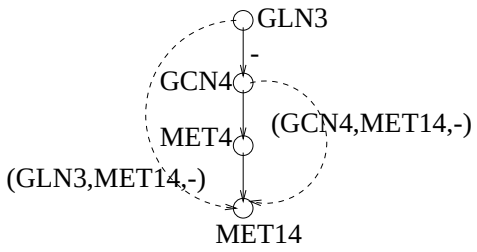


Table 4.6: Protein-protein interactions used in explaining knock-out effects

Type	Not used	Used ≤ 10 times	Used > 10 times
High-throughput	11650	140	23
Small-scale	2647	93	19

effects in Table 4.6. There are roughly similar number of active protein-protein interactions from small-scale experiments and from high-throughput data. However, since there are far more interactions reported in high-throughput data, the equal proportion is strongly biased toward the interactions from small-scale experiments. The p-value of the hyper-geometric test is 4.8×10^{-17}.

We did not perform cross-validation tests on predicting the knock-out effects as in the previous analysis for computational efficiency. This task is not necessary provided we already understand the conditions for the model predictability.

4.2.2 General properties of inferred subnetworks

We then applied the recursive algorithm described in Section 3.4.3 to decompose the network of physical interactions. The inference results are a collection of 48 subnetworks covering 671 genes and 1111 physical interactions and one of their optimal annotations. One subnetwork contains attributes (edge presence, directions and signs) which are uniquely determined. In other words, they are sufficiently constrained by the Rosetta data. Other subnetworks contain multiple optimal annotations. We will discuss in details about the properties and biological knowledge associated with each subnetwork in the next section. In this section we discuss general properties of the

inference results.

The subnetwork whose annotations are uniquely determined contains several well-studied regulatory pathways in yeast genetics. This is not surprising due to the selection bias of both knock-out and physical interaction data. Knock-out experiments in Rosetta data were selected to fulfill two distinct purposes. Genes belonging to several well-studied biological systems were chosen for the confirmation of computational methods (clustering in their case). Genes with unknown functions were chosen for the interest of understanding their functions by comparing their responses to known gene deletions. Experiments of the first category are likely to constrain the subnetwork tightly since they are targeted to a relatively small network. For example, almost all genes along the pheromone response pathway are perturbed. In contrast, experiments of the second category are likely to scatter throughout the physical network, thus provide sparse constraints on the entire network. Moreover, because both protein-DNA and protein-protein interaction data have strong bias toward the interactions of genes with known functions, effects of deleting unknown ORFs are unlikely to be explained by the current physical interaction network.

The power of the constraint-based modeling framework relies on the property of indirect inference: we can infer the value of an attribute from the constraints which indirectly relate to this attribute. Direct observations of this property are not necessary when there are sufficient indirect constraints. We have seen in the mating pathway example that directions and signs of protein-protein interactions are not directly observed but indirectly inferred from the data. Indirect inference also holds for edge signs of protein-DNA interactions. Because many transcription factors require only a small number of molecules in order to activate/repress genes, the control between transcription factors may not be revealed by checking the expression level changes of transcription factors. For example, the sign of protein-DNA interaction (Gln3,Gcn4) in Figure 4-7 is inferred as negative because a subset of genes involved in amino acid metabolism are down regulated in Gcn4Δ and up regulated in Gln3Δ. Although Gcn4 is also up regulated in Gln3Δ, the change is weak (p-value ≤ 0.039) thus does not qualify as a significant knock-out effect in our model.

Another important property about inferred subnetworks is the interpretation of protein-protein interaction directions. Previously we interpret the direction of a protein-protein interaction as the direction of information flow in regulatory control. This meaning can be understood as the direction of in a signal transduction pathway, like the MAP kinase cascade in the mating response. However, because protein-protein interactions can also foster other regulatory mechanisms, signal transduction pathways may not be the only interpretation for edge directions. An interesting example occurs on the pathway Tup1 $\rightarrow$ Ssn6 $\rightarrow$ Nrg1 in Figure 4-8.3. The complex Tup1-Ssn6 is needed in order to activate the function of Nrg1. Here the pathway does not denote a signal transduction cascade but a causal relation similar to Markov properties in a Bayesian network. Tup1 affects Nrg1 via Ssn6 because it forms a complex with Ssn6 and Ssn6 directly binds to Nrg1. This relation provides another interpretation on pathways of protein-protein interactions. Detailed discussions about this subnetwork will be covered in the next section.

We mentioned at the end of Chapter Three about the potential problems of the recurive algorithm for network decomposition (Figure 3-7). A major concern is that the decomposed subnetworks generated by traversing along different branches of the decision tree may have different structures. To show this problem is empirically minor, we applied the decomposition algorithm 500 times by randomly varying the values of fixed variables and checked whether the structures of decomposed subnetworks varied across the sample. The results are encouraging. Among the 47 decomposed subnetworks 46 of them are identical across the 500 random trials. Only one edge from one subnetwork varies across random trials. The results indicate that the networks reported in this chapter are not the artifacts of a specific MAP configuration.

4.2.3 Descriptions of inferred subnetworks

The inferred subnetworks and their optimal annotations are visualized in Figures 4-7 to 4-13. The graph semantics of edges is identical to the subnetworks of pheromone response in Figure 4-4. Edge types are represented by line types (solid – protein-DNA interaction, dash – protein-protein interaction) and edge signs are color coded (black

– positive, gray – negative). For conciseness we only draw one optimal annotation in these figures. Among the 47 subnetworks we only report 34 of them which can explain a substantial number knockout interactions relative to the network sizes.

To demonstrate the inferred subnetworks reflect the gene regulation mechanisms, we applied two types of internal validations. First, we extracted pathways from all subnetworks and searched online databases (PubMed [4] and YPD [5]) to check whether these pathways are reported in previous studies. 44% (15 out of 34) of the subnetworks contain pathways at least partially identified in previous works. Table 4.7 enlists the known pathways and their functions appeared in the subnetworks. Second, we evaluated the enrichment of functional categories of genes in the same subnetwork. We calculated the hyper-geometric p-values of enrichment according to Munich Information Center for Protein Sequences (MIPS) functional categories [6] and reported the p-values on Table 4.8. 11 of 34 subnetworks are significantly enriched with genes belonging to the same functional category (p-value ≤ 0.07 after Bonferroni correction of multiple hypothesis).

We have discussed in Chapter Three about the unreliable quality of high-throughput protein-protein interaction data from yeast two hybrid systems. By inspecting the subnetworks which contain neither known pathways nor genes enriched with certain functions, we found most of them contained protein-protein interactions reported from high-throughput experiments, primarily yeast two-hybrid systems ([160, 86]) or complex detection ([75]). We suspected most of these subnetworks were spurious due to the unreliable quality of their protein-protein interactions. As a sanity check we excluded the protein-protein interactions reported from high-throughput experiments and performed the same tests on restricted subnetworks. The results (Tables 4.9, 4.10) indicate a moderate improvement in both known pathway fraction and the functional enrichment. There are 24 subnetworks inferred from the restricted protein-protein interactions. Among them 13 subnetworks contain verified pathways and 9 subnetworks are enriched with genes belonging to the same functional categories. These

[4]http://www.ncbi.nlm.nih.gov/PubMed/
[5]https://www.incyte.com/proteome/index.html
[6]http://mips.gsf.de/

Table 4.7: Verified pathways in subnetworks with high-throughput pp interactions

subnetwork	pathway	function	reference
invariant	Kss1→Ste12, Fus3→Ste12	mating response, filamentous growth	[46, 108]
variant1	Sok2→Msn4	yeast PKA pathway	[141]
variant2	Tup1→Hhf1	histone regulation	[30]
variant3	Tup1/Ssn6→Nrg1	glucose metabolism	[121]
variant4	Tup1/Ssn6→ α2/Mcm1	repression of a-specific genes	[163]
variant5	Rpd3→Abf1	histone modification	[33]
variant6	Swi4→Ndd1→Ace2	cell cycle control	[139]
variant8	Slt2→Rlm1, Slt2→Swi4	PKC pathway	[91, 9]
variant10	Med2→Gal4, Med2→Gcn4	general transcription mediator	[120]
variant15	Cmd1→Cna1, Cna1→Skn7	calcium signaling	[51, 168]
variant29	Yap1→Cad1	metal response	[12]
variant30	Med2→Gal4, Med2→Srb6	general transcription mediator	[120, 77]
variant32	Med2→Gal11→Gal4	general transcription mediator	[120]
variant33	Med2→Srb5, Med2→Gal4	general transcription mediator	[77, 120]
variant34	Ste12→Mcm1	mating response	[47]

Table 4.8: Functional enrichment of subnetworks with high-throughput pp interactions

subnetwork	# genes	functional category	p-value
invariant	109	cell fate, homeostasis, metabolism	$1.48 \times 10^{-7}, 0.0067$
variant2	63	protein synthesis	7.13×10^{-8}
variant3	44	transport, metabolism, energy	$1.05 \times 10^{-5}, 5.41 \times 10^{-4}, 0.0766$
variant4	58	cell fate	1.12×10^{-5}
variant7	26	cell cycle	0.035
variant19	9	cell defense	6.33×10^{-6}
variant23	17	metabolism, energy	1.49×10^{-6}, 0.04
variant26	8	cell defense	9.62×10^{-5}
variant34	9	cell fate, homeostasis, cell signal	$4.55 \times 10^{-8}, 0.0012, 0.0345$
variant36	7	metabolism	0.0258
variant40	5	metabolism	0.0017

Table 4.9: Verified pathways in subnetworks without high-throughput pp interactions

subnetwork	pathway	function	reference
invariant	Kss1→Ste12, Fus3→Ste12	mating response, filamentous growth	[46, 108]
variant1	Sok2→Msn4	yeast PKA pathway	[141]
variant2	Tup1/Ssn6→Nrg1	glucose metabolism	[121]
variant3	Tup1/Ssn6→α2/Mcm1	repression of a-specific genes	[163]
variant4	Swi4→Ndd1→Ace2	cell cycle control	[139]
variant6	Slt2→Rlm1, Slt2→Swi4	PKC pathway	[91, 9]
variant7	Cmd1→Cna1, Cna1→Skn7	calcium signaling	[51, 168]
variant10	Med2→Gal11→Gal4	general transcription mediator	[120]
variant14	Gcn4→Met4	methionine synthesis	[113]
variant16	Ste12→Mcm1	mating response	[47]
variant17	Cka2→Abf1	casein kinase pathway	[161]
variant22	Cka1→Abf1	casein kinase pathway	[161]
variant25	Cka1→Abf1	casein kinase pathway	[161]

test results suggest the quality of physical interaction data is crucial for capturing relevant properties of gene regulation. However, we cannot disclaim the usefulness of high-throughput protein-protein interaction data because they do not match the prior knowledge. The better match between the inferred models from the interactions of small-scale assays and prior knowledge may also merely reflect the bias of previous studies concentrated on a small number of well-known systems. Further experiments are required in order to verify the inferred models which are not confirmed by prior knowledge.

The invariant subnetwork can be further divided into 5 components. Each component reflects known biological processes in yeast gene regulation. It is expected since the high-throughput we use are biased toward known biological pathways. Details about each component of the invariant subnetwork are as follows.

- Part of the pheromone response pathway is retrieved by the algorithm as demonstrated in Section 4.1. Because we restrict the maximum path length ≤ 3, this subnetwork can be viewed as a hierarchy of four levels. The bottom level contains mating response genes such as Aga1 and Mfa1. They are all controlled the transcription factor Ste12 at the second level. Ste12 is connected by another

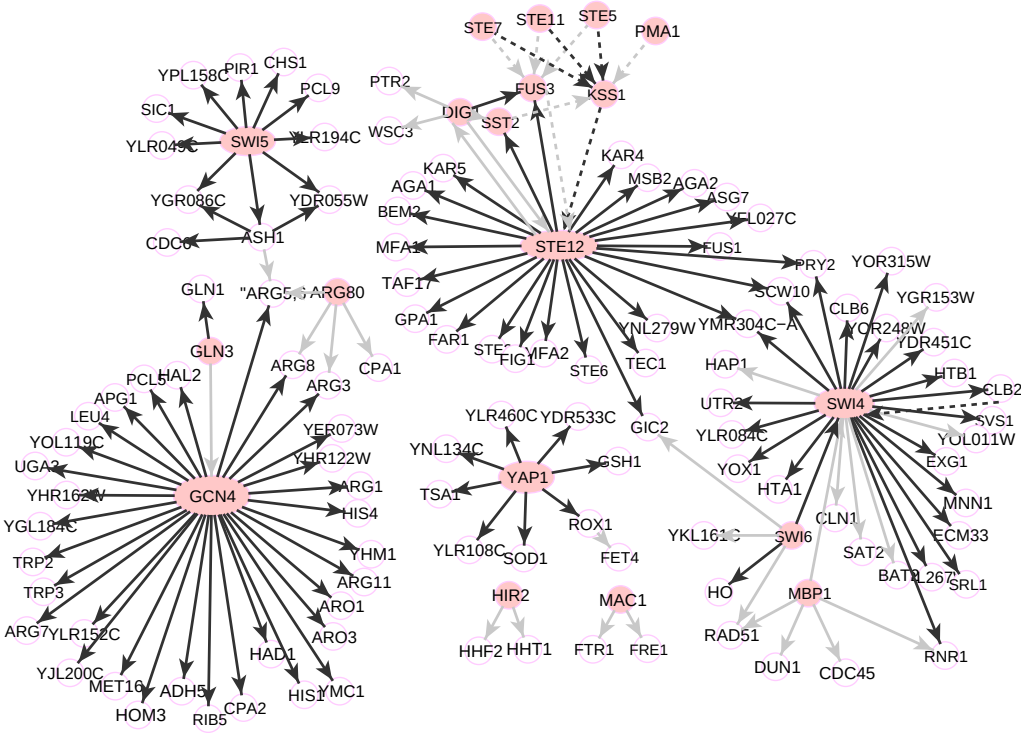
Table 4.10: Functional enrichment of subnetworks without high-throughput pp interactions

subnetwork	# genes	functional category	p-value
invariant	109	cell fate, homeostasis, metabolism	$1.48 \times 10^{-7}, 0.0067$
variant2	44	transport, metabolism, energy	$1.05 \times 10^{-5}, 5.41 \times 10^{-4}, 0.0766$
variant3	58	cell fate	1.12×10^{-5}
variant8	9	cell defense	6.34×10^{-6}
variant12	7	metabolism	0.0258
variant13	17	metabolism, energy	$1.49 \times 10^{-6}, 0.0406$
variant14	6	metabolism	2.88×10^{-4}
variant16	9	cell fate, homeostasis, cell signal	$4.55 \times 10^{-8}, 0.0012, 0.0345$
variant20	5	metabolism	0.0017

transcription factor (Dig1) and MAP kinases (Kss1, Fus3) via protein-protein interactions at the third level. The top level contains genes involved in signal transduction cascade (Ste11, Ste7, Ste5) and some other genes (Pma1, Sst2). The edge signs emanating from Ste12 and edge directions of most protein-protein interactions are consistent with prior knowledge as stated in Section 4.1. The negative sign of protein-DNA interaction (Dig1,Ste12) is also verified in previous study ([153]). There are also some discrepancies with previous studies. For instance, edge signs connecting to Fus3 and the spurious interaction from Pma1.

- A subnetwork centered at Gcn4 is involved with amino acid metabolism. Gcn4 directly controls 32 genes by protein-DNA interactions and most of the interactions are positive. Gln3 indirectly controls some of these genes via the protein-DNA interaction to Gcn4. The sign of (Gln3,Gcn4) edge is negative because these downstream genes are down-regulated by Gcn4 and up-regulated by Gln3. In addition, Arg3, Arg5,6, and Arg8 are also negatively regulated by Arg80. Gcn4 is known to be a master regulator for amino acid synthesis in response to amino acid or purine starvation ([115]). The knock-out data suggests it also regulates these genes under normal condition. Gln3 is a transcription factor for nitrogen regulation ([171]). Its negative effects on several amino acid

Figure 4-7: Physical model uniquely determined from Rosetta data



metabolic genes are explained by pathways via Gcn4. Arg80 is a member of a regulatory complex for arginine synthesis. Arg3, Arg5,6, Arg8 and Cpa1 are all known to be regulated by this complex ([111]).

- The cell cycle related genes are distributed at two components. One component is centered at Swi5 with positive protein-DNA interactions. Swi5 is known to activate genes at M/G1 phase ([139]), and the genes controlled by Swi5 in this component are all active at M/G1 or G1 phase. The activative function of Swi5 is consistent with the positive signs on protein-DNA edges emanating from Swi5. The other component contains cell cycle regulators Swi4, Swi6, and Mbp1 for G1/S phase ([139]). Most genes controlled by these factors occur at G1/S phase, which is consistent with their functional roles. However, some protein-

DNA edges are annotated with negative signs, contradicting with the knowledge about the activating roles of Swi4 and Swi6. The components of cell cycle related genes are smaller than the cell cycle regulatory networks proposed in [139]. This is because our results are based on the static knock-out measurements rather than time-course measurements of synchronized cell culture. Moreover, some of these factors play complementary roles (such as Mbp1 and Swi6). Single deletions will not manifest significant changes on their regulated genes.

- A small component is centered around Yap1 and controlled by protein-DNA interactions. Yap1 is known to be a regulator for oxidative stress response ([24]). All the controlled genes in this component except YLR460C are involving in oxidative stress response. The knock-out data suggests that Yap1 also has a regulatory role under normal condition.

- Hir2 regulates histones Hhf1 and Hhf2 ([142]). Mac1 regulates Fre1 and Ftr1 involved in iron utilization ([92]). These regulatory relations have been identified in previous studies.

We then investigated the biological relevance of variant subnetworks. Some subnetworks are unlikely to bear biological meanings because they contain many edges but only explain few knock-out effects. We discarded the subnetworks whose ratios (# explained knock-out pairs)/ (# edges) is less than 0.5. Each variant subnetwork contains subconfigurations which differ only on edge signs. We illustrate these subnetworks in Figures 4-8, 4-9, 4-10, 4-11, 4-12 and 4-13.

- Subnetwork 1 constitutes hubs Swi4, Sok2, Cup9, Hap4, Msn4 and Yap6 connected by protein-DNA interactions. Swi4 connects to hubs Cup9, Hap4, Msn4 and Yap6 via Sok2. Each hub is a transcription factor and directly controls a number of genes by protein-DNA interactions. This structure explains many knock-out effects in Swi4Δ experiment. The explanations are carried out along paths Swi4 $\rightarrow$ Sok2 $\rightarrow$ (Cup9, Hap4, Msn4, Yap6) $\rightarrow$ downstream genes. Degeneracy arises in the freedom of varying edge sign configurations. There are

5 edges connecting hubs and two hub genes have significant changes in Swi4Δ, thus there are 8 degenerate edge sign configurations. Sok2 is known to repress Msn4 according to previous studies ([141]): Sok2 is at the end of the PKA pathway and Msn4 is repressed by PKA pathway. Further discussions and validations about this subnetworks will be covered in Chapter Five.

- Subnetwork 2 constitutes the protein-protein pathways Tup1 → Hhf1 → Fhl1, Tup1 → Hhf1 → Fkh1, and Tup1 → Hhf1 → Rfx1. The terminal proteins Fhl1, Fkh1 and Rfx1 are transcription factors and each in turn binds to a number of genes. These pathways explain the knock-out effects in Tup1Δ on their downstream genes. The path lengths are three while the aggregate signs are fixed by the knock-out constraints. Thus there are 4 degenerate configurations. Only part of these pathways is verified in previous studies. Tup1 is known to mediate the repression of many genes via interacting with histones including Hhf1 ([30]). This interaction may affect transcription factors including Fhl1, Fkh1 or Rfx1. Notice a number of ribosome genes are down regulated in Tup1Δ. These knock-out effects are explained via the pathway Tup1 → Hhf1 → Fhl1.

- Subnetwork 3 constitutes a mixed pathway Tup1 → Ssn6 → Nrg1 → Yap6. Both Nrg1 and Yap6 are transcription factors and bind to a number of genes. (Nrg1,Yap6) is a protein-DNA binding and other edges along the path are protein-protein bindings. It explains the knock-out effects in Tup1Δ and Ssn6Δ experiments. These pathways contain 3 edges and are constrained by 2 knock-out experiments. Thus there are 2 degenerate configurations. These pathways match the biological knowledge about glucose repression response. Nrg1 is a transcription repressor regulating glucose metabolic genes such as Sta1. Its repression activity is known to be enabled by recruiting the Ssn6-Tup1 complex ([121]). There is a freedom of assigning the edge sign of (Ssn6,Nrg1). This degeneracy can be removed by performing Nrg1Δ experiment.

- Subnetwork 4 constitutes intermediate hubs A2 and Mcm1. It explains the knock-out effects in Tup1Δ by the path Tup1 → A2 → Mcm1 → downstream

genes. It has 4 degenerate configurations. Downstream genes are enriched with mating response genes. According to previous studies, the homeodomian protein $\alpha 2$ together with Mcm1, recruits general transcription repressors Ssn6 and Tup1 to the promoters of a-specific genes ([163]). A2 is similar to $\alpha 2$. These interactions are exactly captured in this subnetwork.

- Subnetwork 5 constitutes the protein-protein pathway Rpd3 → Ckb2 → Abf1. Abf1 is a DNA-binding protein that controls a number of genes. It explains the knock-out effects in Rpd3Δ and Ckb2Δ experiments. There are 2 degenerate configurations for pathway length is 3 and is constrained by two knock-out experiments. Ckb2 is a casein kinase which phosphorylates Abf1 ([161]), and Rpd3 is a histone deacetylase required for the regulation of many genes. The transcription activation function of Abf1 is weakly repressed by Rpd3 ([33]). This evidence supports using the pathway Rpd3 → Ckb2 → Abf1 to explain knock-out effects.

- Subnetwork 6 constitutes cell cycle related genes mediated by cell cycle transcription factors Ndd1, Ace2 and Swi5. It explains mostly the knock-out effects in Swi4Δ experiment. The pathways mediated by Ndd1 contain 2 edges while the ones mediated by Ndd1 and Ace2 contain 3 edges. Thus there are totally 3 degenerate configurations. These transcription factors control cell cycle genes at different phases: Swi4 – S/G1, Ndd1 – G2/M, Ace2 – M/G1 ([139]). The data of ΔSwi4 experiment indicates deleting Swi4 affects not only genes occurring at S/G1 phases but also G2/M and M/G1 phases as well. Some of these affected genes are bound by Ndd1 and Ace2. A causal explanation is difficult because an intervention of cell cycles can lead to drastic changes. Nevertheless, the pathway Swi4 → Ndd1 → Ace2 provides one possible explanation.

- Subnetwork 7 constitutes the protein-protein pathway Cdc42 → Adh2 → Fkh2, and Fkh2 controls a number of genes. It explains the knock-out effects in Cdc42Δ experiment. There are 4 degenerate configurations. Downstream genes are enriched with cell cycle genes. We have not found the support of this

pathway from literature survey. The binding of (Cdc42,Adh2) belongs to a complex detected by systematic mass spectrometry ([75]). Thus it might be generated from artifacts.

- Subnetwork 8 constitutes hubs Slt2, Rlm1, and Swi4 connected by protein-protein interactions. Rlm1 is a transcription factor that controls a number of genes. The knock-out effects in experiments YOR080WΔ, Pma1Δ, Qcr2Δ and Swi6Δ are explained by paths mediated via (Slt2,Rlm1) or (Slt2,Swi4). There are 4 degenerate configurations. Rlm1 regulates the genes involved in cell wall integrity. It is enabled by the MAP kinase Slt2 ([91, 9]). Interestingly, most knock-out effects explained by (Slt2,Rlm1) are on the genes involved in cell wall structure (Exg1, Scw10, Mnn1, YGR153W, Hap1, Pry2, Sat2, YDR451C, Svs1). This fact supports the validity of our explanation.

- Subnetwork 10 constitutes hubs Med2, Srb4, Gcn4, Gal4, Mcm1 and Yap6. The knock-out effects of deleting Med2 are explained by pathways Med2 $\rightarrow$ Srb4 $\rightarrow$ Gal4, Med2 $\rightarrow$ Srb4 $\rightarrow$ Gcn4, Med2 $\rightarrow$ Srb4 $\rightarrow$ Mcm1, or Med2 $\rightarrow$ Srb4 $\rightarrow$ Yap6. Med2 and Srb4 belong to the general transcription apparatus which are known to interact with Gcn4 and Gal4 ([120]).

- Subnetwork 11 constitutes the protein-protein pathway Rpd3 $\rightarrow$ Vps1 $\rightarrow$ Reb1, and Reb1 is a transcription factor. It explains the knock-out effects in Rpd3Δ experiment and has 4 degenerate configurations. Rpd3 and Vps1 belong to the same complex detected by systematic mass spectrometry, and Vps1 and Reb1 belong to another complex detected by the same technology.

- Subnetwork 13 contains intermediate hubs of cell cycle regulators Fkh2, Ace2 and Mcm1 which relay the effects of deleting Sin3 to downstream genes. The protein-protein interaction (Sin3,Fkh2) is reported from the high-throughput experiment of protein complexes ([62]). Therefore, the results are less reliable.

- Subnetwork 14 constitutes the protein-protein pathway Sin3 $\rightarrow$ Ckb1 $\rightarrow$ Abf1 and explains the knock-out effects in Sin3Δ. The interaction (Sin3,Ckb1) is

again reported from high-throughput experiment of protein complexes ([75]).

- Subnetwork 15 constitutes the protein-protein pathway Cmd1 → Cna1 → Skn7, and Skn7 is a transcription factor. It explains the knock-out effects in Cmd1Δ experiment and has 4 degenerate configurations. Cmd1 and Cna1 are on the calcineurin pathway ([51]). Skn7 is involved in oxidative and osmotic stress response and cell cycle control, but is also required for the calcineurin pathway ([168]).

- Subnetwork 16 contains the pathways of explaining the knock-out effects in Vma8Δ experiment. The explanation is carried out by paths Vma8 → Gcn3 → Fhl1, and Vma8 → Gcn3 → Mac1. There are 8 optimal configurations along the two pathways. (Vma8,Gcn3) and (Gcn3,Fhl1) belong to separate complexes detected by systematic mass spectrometry ([75]), and (Gcn3,Mac1) is detected by two-hybrid experiment ([86]).

- Subnetwork 19 constitutes the mixed pathway Sir4 → Rap1 → Msn4, and Msn4 binds to the DNA promoters of stress response genes. (Sir4,Rap1) is a protein-protein interaction and (Rap1,Msn4) is a protein-DNA interaction. It explains the knock-out effects in Sir4Δ experiment. Downstream genes are enriched with stress response genes. There are 4 degenerate configurations along the pathway. Rap1 is known to be a silencer of telomere sites, and its activity is enabled by binding to Sir4 ([105]).

- Subnetwork 20 constitutes the protein-protein pathway Rrp6 → Rif2 → Rap1. It explains some knock-out effects in Rrp6Δ experiment. Downstream genes are enriched with ribosome genes. The interaction (Rrp6,Rif2) is reported from a high-throughput experiment ([86]) thus is less reliable.

- Subnetwork 21 constitutes the protein-protein pathway Cmd1 → Hap5 → Gal4. It explains some knock-out effects in Cmd1Δ. The interaction (Cmd1,Hap5) is reported from a high-throughput experiment ([177]).

- Subnetwork 23 explains several knock-out effects in Gcn4Δ and Gln3Δ experiments by protein-DNA pathway Gln3 → Gcn4 → Rtg3. There are 2 degenerate configurations. Downstream genes are enriched with genes involved in amino acid synthesis. Most affected genes are also directly connected to Gcn4, for instance, Yhm1, Arg5,6, Arg1, YHR162W, Arg11, Uga3, Cpa2, Aro1. The pathway via Rtg3 may not be necessary to explain these knock-out interactions. However, evidence shows Rtg3 is involved in amino acid synthesis as Gcn4 and Gln3 ([88]). Thus this alternative pathway may still be valid.

- Subnetwork 24 contains protein-protein pathway Swi4 → Spk1 → Cbf1. It explains a few knock-out effects in Swi4Δ. Downstream genes are enriched with genes involved in metabolism. The interaction (Swi4,Spk1) is reported from a high-throughput experiment ([75]).

- Subnetwork 25 contains protein-protein pathway Erg2 → Soh1 → Yap6. It explains some knock-out effects in Erg2Δ. Both protein-protein interactions are reported from yeast two-hybrid high-throughput experiments ([86]).

- Subnetwork 26 contains protein-protein pathways YER083C → YDL100C → Msn4 and YER083C → YDL100C → Yap6. Downstream genes are enriched with stress response genes. These protein-protein interactions are reported from high-throughput experiments ([86]).

- Subnetwork 27 contains protein-protein pathways Kim4 → Sua7 → Gal4, Kim4 → Sua7 → Pho4, and Kim4 → Sua7 → Ace2. It explains some knock-out effects in Kim4Δ experiment. The protein-protein interactions in this subnetwork are again reported from high-throughput experiments ([86]).

- Subnetwork 28 contains pathway Sin3 → Stb1 → Ndd1. (Sin3,Stb1) is a protein-protein interaction and (Stb1,Ndd1) a protein-protein interaction. It explains a few knock-out effects in Sin3Δ experiment. Stb1 and Sin3 are known to form a complex which affect histone acetylation of many genes ([93]). However, the effect of Sin3-complex on Ndd1 is not reported.

- Subnetwork 29 contains protein-protein pathways Yap1 → Nup116 → Cad1 and Kim4 → Nup116 → Cad1. It explains some knock-out effects in Yap1Δ and Kim4Δ experiments. The interactions (Yap1,Nup116) and (Kim4,Nup116) are reported from high-throughput experiments ([86]).

- Subnetwork 30 contains protein-protein pathway Med2 → Srb6 → Gal4. It explains some knock-out effects in Med2Δ experiment. Med2 and Srb6 belong the the general transcription apparatus and affect the function of Gal4 ([120]).

- Subnetwork 31 contains protein-protein pathway YER044C → Lys14 → Yap6. It explains some knock-out effects in YER044CΔ experiment. Both protein-protein interactions are reported in high-throughput assays ([86]).

- Subnetwork 32 constitutes protein-protein pathway Med2 → Gal11 → Gal4. It explains some knock-out effects in Med2Δ. Med2 and Gal11 belong to the general transcription apparatus and it is known to affect the function of Gal4 ([120]).

- Subnetwork 33 constitutes protein-protein pathway Med2 → Srb5 → Gal4. It explains some knock-out effects in Med2Δ. Med2 and Srb5 belong to the general transcription apparatus and it is known to affect the function of Gal4 ([120]).

- Subnetwork 34 constitutes protein-protein pathway Ste12 → α1 → Mcm1. It explains some knock-out effects in Ste12Δ. Downstream genes are enriched with mating response genes. However, most of these interactions can be explained by direct binding from Ste12.

- Subnetwork 36 contains pathway Swi4 → Tye7 → Ino4. (Swi4,Tye7) is a protein-DNA interaction and (Tye7,Ino4) a protein-protein interaction. It explains a few knock-out effects in Swi4Δ experiment. Downstream genes are enriched with genes involved in metabolism. The protein-protein interaction (Ino4,Tye7) is reported in previous study ([129]), and Ino4 is known to regulate some of the downstream genes ([67]). Thus this pathway is possibly active.

- Subnetwork 39 contains protein-protein pathway Gln3 → Gts1 → Yap6. It explains knock-out effects in Gln3Δ. Both interactions are reported in high-throughput assays ([160, 86]).

- Subnetwork 40 explains the knock-out effects in Rtg1Δ experiment on Ade genes by the protein-DNA edges (Rtg1,Bas1). There are 2 degenerate configurations. ADE genes are responsible for amino acid synthesis and Bas1 is known to activate these genes ([34]). (Rtg1,Bas1) interaction is reported in a high-throughput experiment ([86]). Although its function is not supported by previous studies according to literature search, this small subnetwork is worth of further verification.

- Subnetwork 42 contains protein-protein pathway Clb2 → Nap1 → Yap6. It explains some knock-out effects in Clb2Δ experiment. Nap1 is known to be required for the function of Clb2 ([3]), but the interaction (Nap1,Yap6) is reported from high-throughput assays ([86]). Thus it is uncertain whether the entire pathway is active.

From the descriptions above, we can see some subnetworks correspond to known gene regulatory networks, while others do not have clear biological interpretations. The pathways which are not verified in previous studies are generated by high-throughput experiments such as coIP, two-hybrid systems, and systematic mass spectrometry. These data are subject to high false positive rate, thus the resulting pathways may be artifacts. However, if a subnetwork is able to explain the knock-out effects from multiple experiments, then it is less likely to be generated by random chance. In any case, the only method to verify/falsify these explanations is to perform further experiments – for example, deleting other genes along the pathway.

Figure 4-8: Decomposed subnetworks 1-6

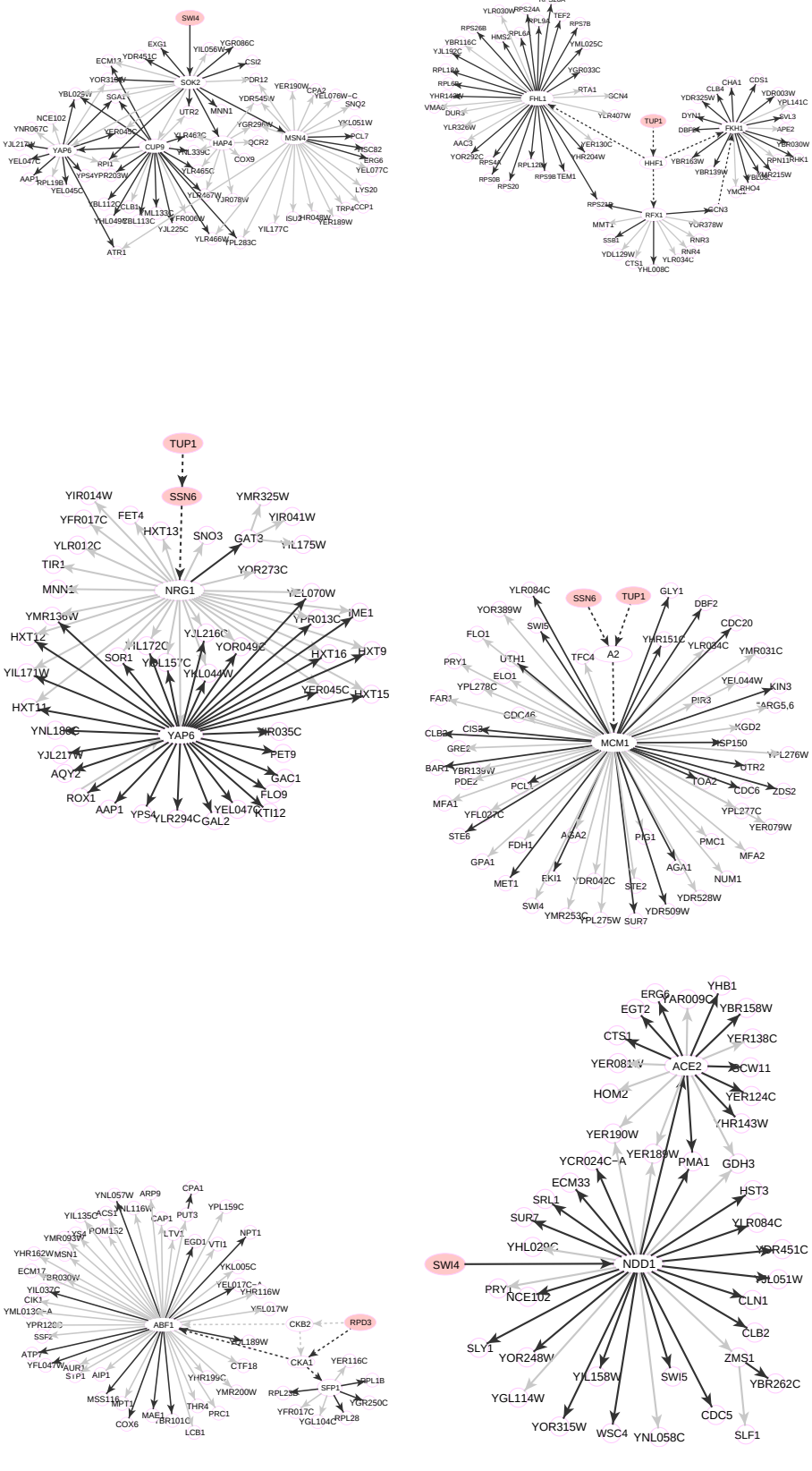


Figure 4-9: Decomposed subnetworks 7-14

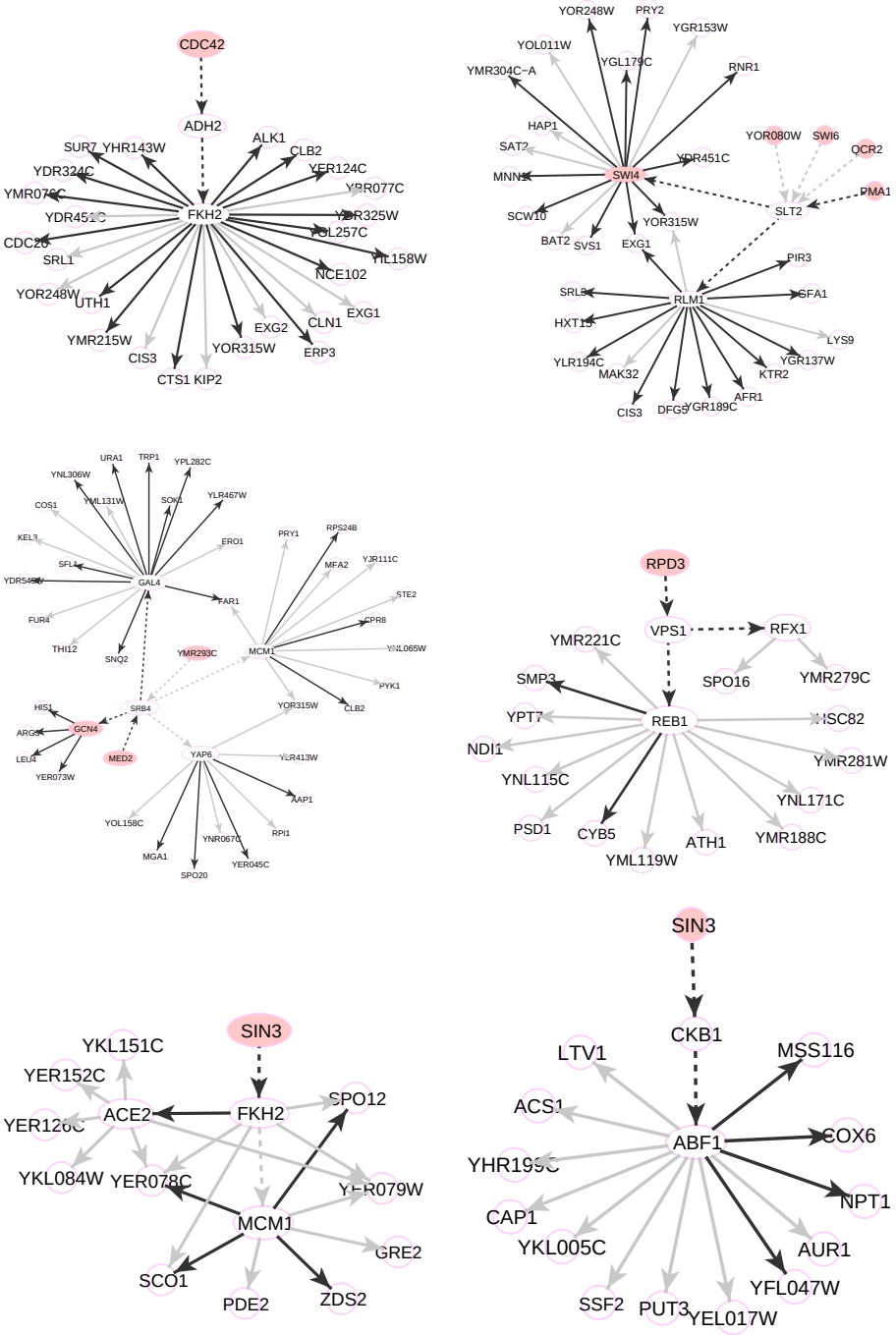


Figure 4-10: Decomposed subnetworks 15-23

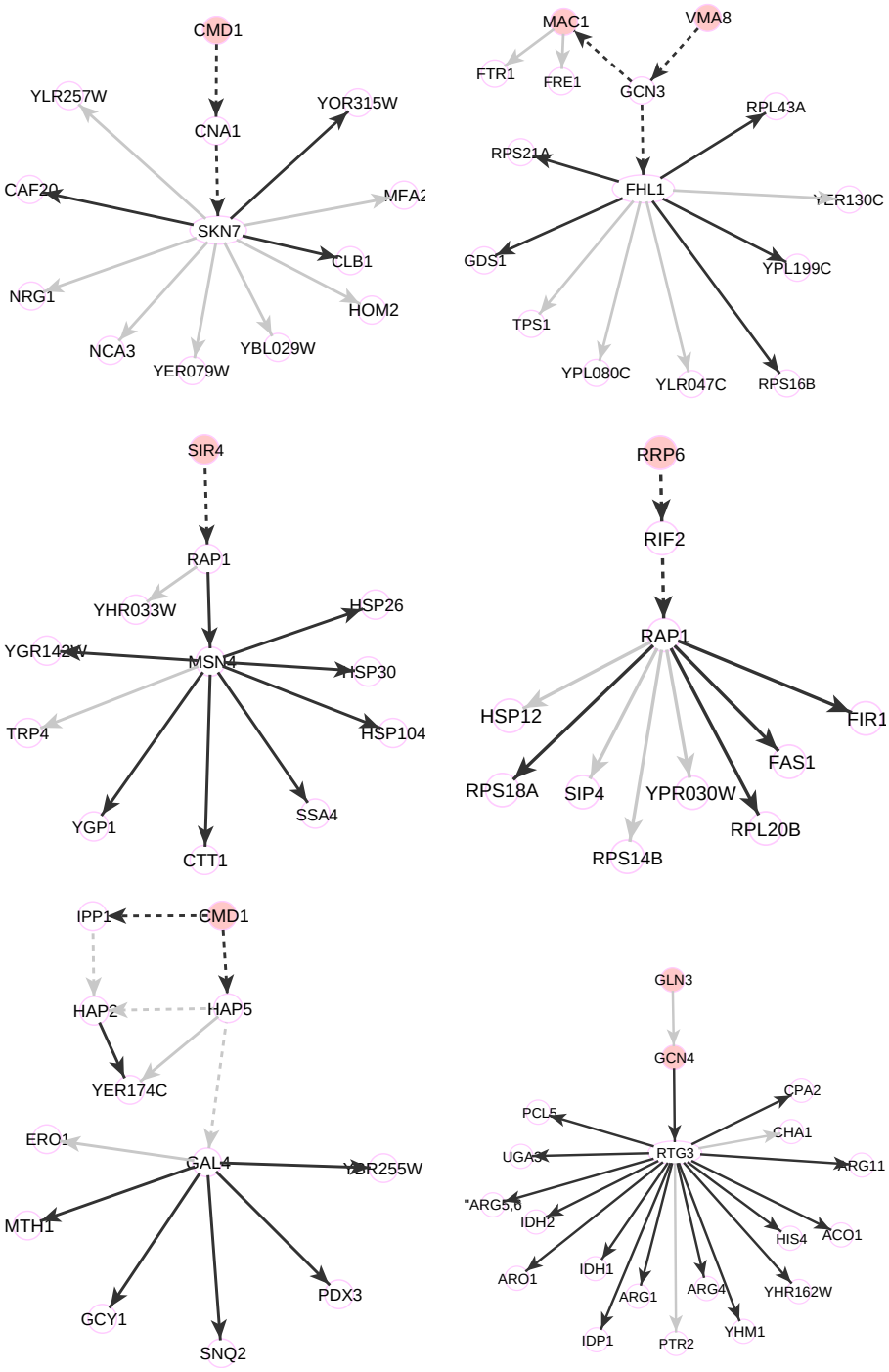


Figure 4-11: Decomposed subnetworks 24-29

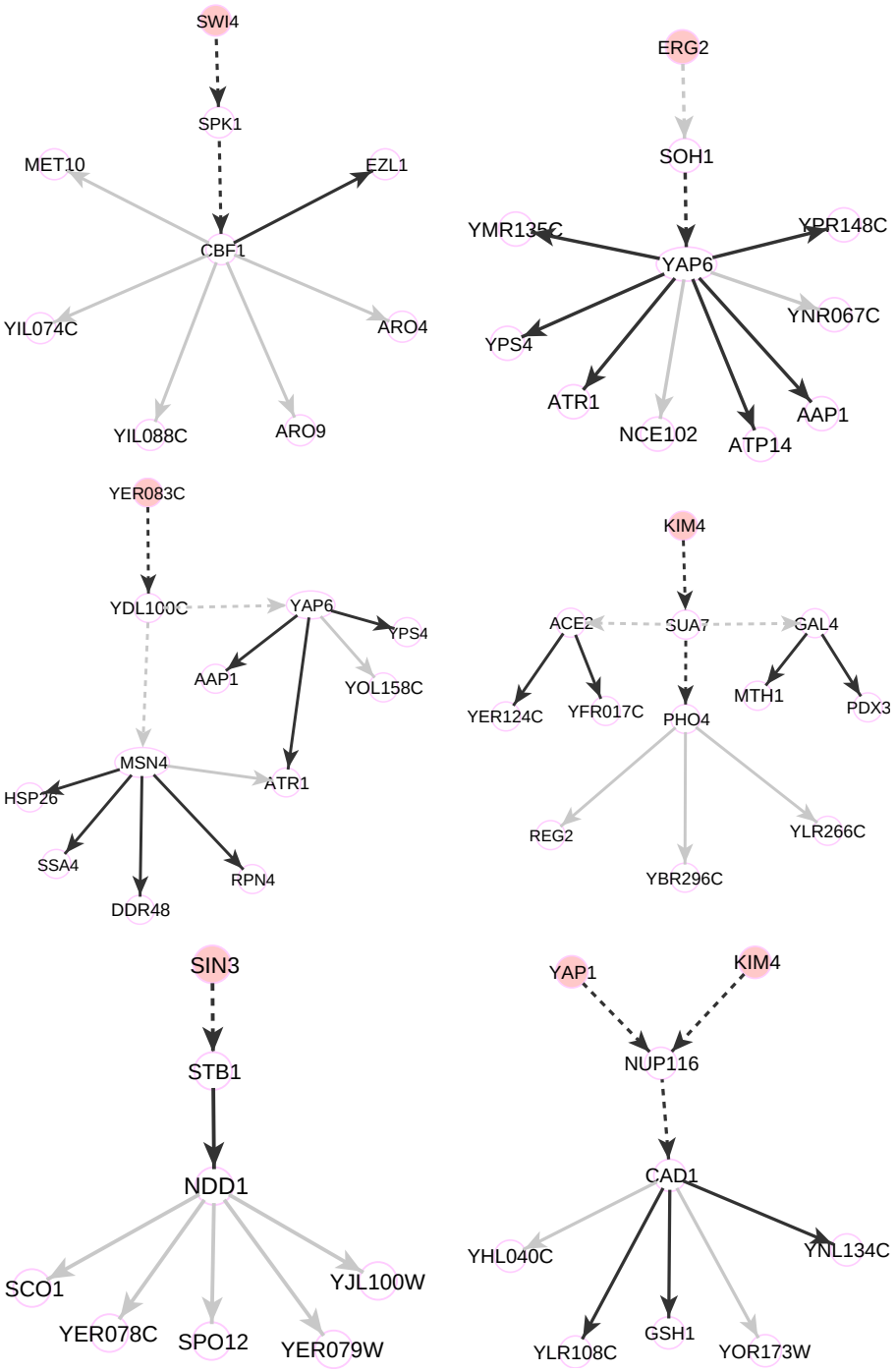


Figure 4-12: Decomposed subnetworks 30-36

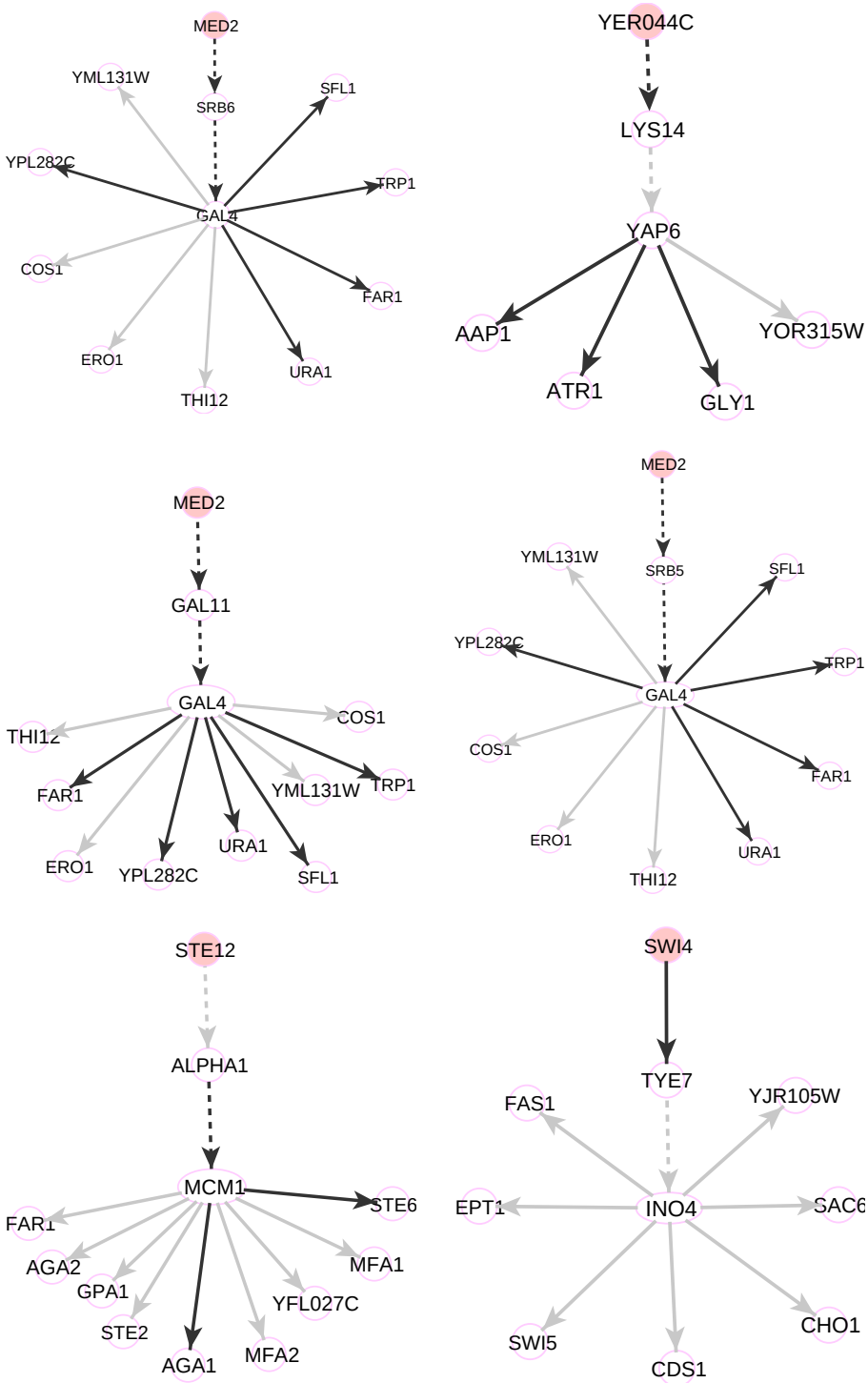
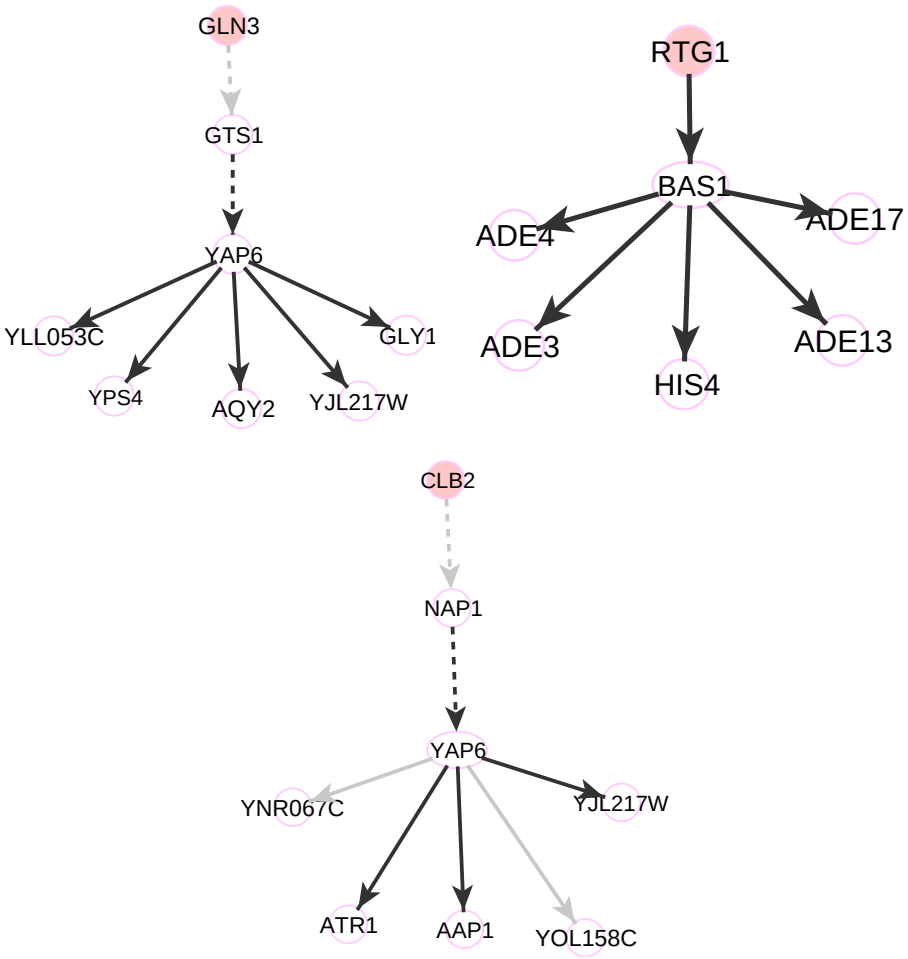


Figure 4-13: Decomposed subnetworks 39-42



Chapter 5

Experimental Design

The value of a scientific model lies on its capability of predicting the behavior of a system in addition to explaining existing data. A model is *falsified* if its prediction contradicts with data from new experiments or new evidence. Philosopher Karl Popper employed the concept of falsifiability as the criterion to distinguish science from other discourses of natural/social phenomena such as astrology ([126]). He argued that scientific theories generated from singular experience could never be logically verified as universal statements. Conversely they must be possible to be refuted by experience (where experiments are a specific realization of experience). In reality, scientists may have different strategies of dealing with contradictions. They may choose either to abandon the current model and establish a new one, or to revise the current model to make it compatible with new data. One can easily find precedents of both scenarios in science history.

From either the falsification or the modification perspective, data generated from new experiments or observations are the ultimate way of judging a scientific model. Very often experiments and even observations are expensive in terms of various resources (brain power and labor, time, money, social impact). Therefore, it is critical for experimental scientists to prioritize experiments in terms of various goals which can be scientific or not.

By applying the physical network modeling framework described in Chapters Two and Three, we have generated a collection of models – likely molecular interaction

pathways responsible for gene regulation and their functional annotations – which explain the high-throughput data of physical interactions and knock-out gene expression. There are two questions pertaining to further inquiring the inferred models. A *model discrimination* problem concerns which one among the many equally good models is true (or is consistent with the new data). A *model validation* problem concerns whether the true model is on the list of candidate models. We cannot answer either question without having the new data. Nevertheless, we can establish criteria for selecting new experiments in order to fulfill these goals without knowing the new data. The method of selecting new experiments according to existing models and data is called experimental design.

We discuss in this chapter methods of selecting experiments in order to discriminate the degenerate models inferred from existing data. We introduce an information theoretic score to prioritize new knock-out experiments. A list of experiments generated from these criteria are proposed to biologists, and a number of experiments on this list are conducted. We further analyze the data generated from the new experiments, validate and discriminate several subnetworks described in Chapter Four. We leave the experimental design for model verification for the future work and discuss it in Chapter Seven.

5.1 Overview of experimental design

Experimental design, also known as active learning, involves prioritizing or selecting new experiments or observations according to the information obtained from existing data. It has been employed in a wide range of problems in statistics and machine learning, such as design of surveys ([53]), generation of interactive queries for World Wide Web information retrieval ([87]). Recently, experimental design is applied in computational biology. A prominent example is the robot scientist which automates every step of scientific inquiry (hypothesis generation, inference, experimental design, performing experiments, and so on) ([96]). Although the methods used in these applications are different, they all follow three general principles. First, they cannot

utilize the data generated by the proposed experiments because the experiments are not yet performed. Alternatively, they predict the outcomes of experiments according to the current model and take the expectation over the hypothetical data generated by proposed experiments. Second, designed experiments can perturb and modify the target system. Perturbation data differ from observation data and need to be incorporated in the model in different ways. For instance, if variable X causes variable Y in an unperturbed system, perturbing Y breaks its causal link with X. When the goal is to learn the unperturbed system, the procedures of learning the models should also be modified in accordance with the perturbation experiments ([26, 123]). Third, although in principle batch experiments are allowed under the same experimental design method ([48]), most current approaches sequentially choose one experiment each time. Moreover, most works adopt the *myopic learning* approach: compute the scores of new experiments based on the models (or the distribution of models) learned at the previous step ([158, 159]).

A brief review about the previous works of experimental design is already covered in Chapter One and not repeated in this Chapter. We will discuss the method of prioritizing experiments for model discrimination in the next section.

Notice the term of experimental design is used in an abstract and computational sense. This definition can be very different from experimental scientists' views about experimental design – for example, designing protocols, choosing instruments, tuning the environmental factors, and so on. The details about the procedures of the experiments will not be discussed in this thesis.

5.2 Experimental design for model discrimination

5.2.1 Model uncertainty and model discrimination

We have defined in Chapter Three a configuration of the physical network model as an instantiation of variable values pertaining to the physical interaction network. We are interested in the optimal or sub-optimal model configurations which fit the current

data. In Chapter Three, we also proposed various recursive inference algorithms to either enumerate all optimal configurations or obtain a concise representation of them.

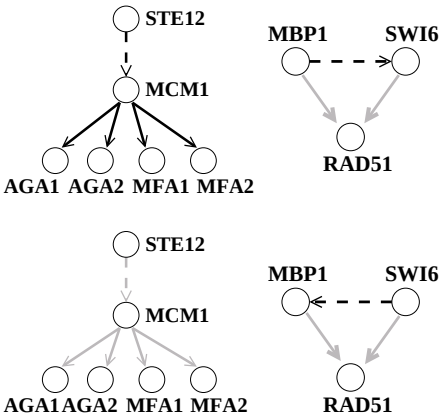
The term model discrimination typically denotes finding the model among multiple candidates which optimally fit the data. In our context, this means narrowing down the optimal configuration to a unique setting. In a more general sense model discrimination denotes reducing the uncertainty about model configurations; in other words, making the distribution of model configurations more "peaked" at a small number of configurations. These two definitions do not contradict and will eventually converge. However, they may lead to different criteria of selecting new experiments. Here we adopt the reduction of model uncertainty as the goal of model discrimination.

The amount of uncertainty about model configurations can be measured by Shannon's entropy of the posterior distribution of model configurations. The entropy is small when the posterior probability mass is concentrated on a small number of configurations. However, the entropy of the joint posterior distribution is difficult to evaluate and visualize. To facilitate understanding its network properties, we illustrate the notion of model uncertainty with extreme cases of degenerate optimal configurations. When explaining model uncertainty, we only show the number of optimal configurations in the network or the number of optimal values of single variables. However, we consider all possible configurations when prioritizing new experiments.

The number of variables in the physical network model often far exceeds the number of constraints imposed by existing data. Consequently, there are many optimal configurations which fit the data equally well. One of the most comprehensive knock-out gene expression datasets in yeasts is the Rosetta Compendium data. Despite it covers 271 single gene deletion experiments, the physical interaction network is still highly under-constrained. Less than 5% (1091 out of 23766) knock-out interactions are connected with path length ≤ 3 in the physical network, and only 5.6% (1142 out of 20361) physical interactions are on paths connecting knock-out effects. Moreover, among the 1091 knock-out interactions which are connected in the physical network, 534 are connected by paths which are not used to explain any other knock-out effects. Variables along these paths are constrained by only single knock-

Figure 5-1: Examples of edge sign and direction degeneracy.

black: positive, gray: negative.



out interactions. "Explanation" of these knock-out interactions is easily achieved by tuning edge directions and signs. However, the inferred results are unreliable because they fit single evidence with complex model configurations.

Given the sparse constraints from existing data, many model configurations are expected to fit the data equally or nearly equally well. Two model configurations are *degenerate* when both yield the maximal likelihood value. The degeneracy of an edge presence variable occurs when the evidence from one data (for instance, yeast two-hybrid data) supports its absence while the evidence from another (for instance, knock-out data) supports its presence. Currently, we incorporate the physical interactions with high confidence values in the skeleton graph. The potential functions of physical data corresponding to these edges all prefer the presence of their interactions. Under this construction, $\phi(x = 1) > \phi(x = 0)$ for all potential functions of pairwise physical interactions. Since knock-out explanations do not force the absence of edges, the scenario of contradictory evidence from physical and functional data does not occur. Degeneracy of edge signs arises when there are not sufficient knock-out interactions probing each gene along the same path. In this scenario, the aggregate sign along a path is fixed, while the signs of individual edges are not. There are multiple edge sign configurations which yield identical aggregate sign. The number

of degenerate configurations depends on the number of aggregate sign configurations and the path length. This scenario is illustrated in an example in the left part of Figure 5-1. A knock-out interaction (Ste12,Aga1,-) (deleting Ste12 down-regulates Aga1) forces the aggregate sign along the path Ste12 → Mcm1 → Aga1 to be +1. Signs of individual edges (Ste12,Mcm1) and (Mcm1,Aga1) can vary as the path sign parity conforms with the knock-out effect. Degeneracy of edge directions arises when different knock-out effects are explained by paths with opposite directions. The right part of Figure 5-1 illustrates the case. The paths Mbp1 → Rad51 and Mbp1 → Swi6 → Rad51 explain the knock-out effect (Mbp1,Rad51,+), and the paths Swi6 → Rad51 and Swi6 → Mbp1 → Rad51 explain the knock-out effect (Swi6,Rad51,+), The two paths Mbp1 → Swi6 → Rad51 and Swi6 → Mbp1 → Rad51 have opposite directions on protein-protein edge (Mbp1,Swi6). Edge direction degeneracy is not likely to occur because the confidence values of two knock-out effects are hardly identical. If one interaction has a slightly higher value, then the model would prefer one edge direction in order to explain the stronger knock-out effect. The degeneracy of a path selection variable occurs when the explanatory paths of two knock-out effects yield a contradiction of edge directions or edge signs. Figure 5-1.2 also illustrates this case. The paths Mbp1 → Swi6 → Rad51 and Swi6 → Mbp1 → Rad51 are involved in different knock-out effects. They cannot co-exist if a protein-protein edge has a unique direction. Either Mbp1 → Swi6 → Rad51 or Swi6 → Mbp1 → Rad51 is an optimal configuration for both knock-out effects are explained in each scenario. The degeneracy of a knock-out effect occurs when the evidence from knock-out gene expression measurement contradict the constraints of explanation. This is unlikely to occur because we incorporate the knock-out effects with high confidence values. In the analysis of high-throughput datasets in Chapter Four, the model contains 14876 direction variables of protein-protein edges and 20361 edge sign variables. We evaluated the max-marginal probability of each variable using the max-product algorithm. Among the 1597 variables whose max-marginal probabilities yield degenerate optimal values, 1403 of them are edge sign variables, 184 are edge direction variables and 10 are path selection variables. Hence a predominant number of degenerate model

configurations are on edge signs.

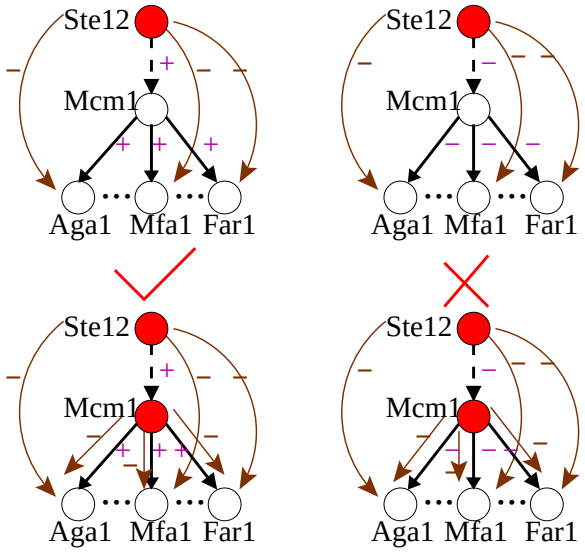
5.2.2 Prioritizing experiments for model discrimination

Our experimental design method focuses on the (mRNA) gene expression experiments of single deletion mutants. This type of data is a natural extension of the Rosetta data used in the model inference. Therefore, the incorporation of the new data into the existing model and its mechanistic interpretation can be undertaken without expanding the data association methods to other types of data. Single knock-out experiments are also easier to implement compared to other gene perturbation assays such as over-expression or double deletion experiments.

How do we define the capacity of a knock-out experiment for model discrimination? Before answering this question, we need to understand how to discriminate model configurations from the outcomes of a knock-out experiment. Roughly speaking, each model configuration predicts the response of all genes under a knock-out experiment. We can calculate the likelihood values of the new data under each configuration. The models whose predicted responses are close to the actual response from the real data yield high likelihood values. We can thus narrow down the candidate models by comparing the likelihood values of the new data.

We illustrate the procedures of model discrimination and experimental design in Figure 5-2. A physical network model on the pheromone response pathway yields two degenerate configurations ([172]). The aggregate sign along the paths from Ste12 to a number of genes via Mcm1 is positive, hence the signs of individual edges are either all positive or all negative. These two configurations explain the knock-out effects in Ste12Δ equally well, but their predicted responses of deleting Mcm1 are distinct. Thus one may compare the measurement data in the Mcm1Δ experiment to the predicted responses according to the models. Configuration 1 is preferred if more downstream genes are down-regulated and vice versa. Deleting other genes (except Ste12) in the subnetwork does not affect downstream genes of Mcm1, hence the predicted responses according to both configurations are identical (no changes). Therefore, Mcm1Δ is suggested as it discriminates between the two degenerate models.

Figure 5-2: Toy example of model discrimination



In practice, deleting Mcm1 is difficult because it is lethal to yeast cells. Elaborated approaches such as temperature sensitive mutants are required.

We define the following notation. Denote e as a new knock-out experiment and $Y^e = (Y_1^e, \cdots, Y_n^e)$ as a predicted expression profile under experiment e. Each component of Y^e is the predicted response of a gene, which takes values in $\{+1, -1, 0\}$ denoting up/down regulations or no change. $\hat{Y}^e$ is the vector of quantized, actual expression data of experiment e. M denotes the identity of model configurations. The uncertainty about model configurations M is reduced as the data from new experiments $\hat{Y}^e$ are provided. The reduction of model uncertainty is captured by the reduction of Shannon's entropy given the new evidence $\hat{Y}^e$:

$$H(M) - H(M|Y^e = \hat{Y}^e). \tag{5.1}$$

This quantity is not computable since $\hat{Y}^e$ is yet to be measured. We thus substitute it with the expected reduction of entropy conditioned on the predicted response Y^e:

$$H(M) - \sum_{y^e} P(Y^e = y^e) H(M|Y^e = y^e) = H(M) - H(M|Y^e) \equiv I(M; Y^e). \tag{5.2}$$

This quantity is the mutual information between M and Y^e:

$$\begin{aligned} I(M;Y^e) &= \sum_{m,y^e} P(m,y^e) \log\left(\frac{P(m,y^e)}{P(m)P(y^e)}\right) \\ &= \sum_{m,y^e} P(m)P(y^e|m) \log\left(\frac{P(y^e|m)}{P(y^e)}\right) \\ &= -\sum_{y^e} P(y^e) \log P(y^e) + \sum_m P(m) \sum_{y^e} P(y^e|m) \log P(y^e|m) \\ &= H(Y^e) - H(Y^e|M) = H(M) - H(M|Y^e). \end{aligned} \tag{5.3}$$

where m denotes the index of model configurations, y^e the vector of predicted responses, $H(Y^e)$ and $H(Y^e|M)$ the entropy and the conditional entropy of Y^e. It can be interpreted as the maximal amount of "information" about M that can be extracted from Y^e. Equivalently, it is the reduction of uncertainty about M by knowing Y^e. Because the mutual information is symmetric, it can also be understood as the reduction of uncertainty about Y^e by knowing M. The intuition of extracting the maximal information about model identity from their predicted responses is in line with the goal of model discrimination.

We ignore measurement noise and assume the mapping from model configuration M to predicted response Y^e is deterministic. The conditional probability $P(Y^e|M)$ can be expressed as

$$P(Y^e = y^e|M = m) = \delta(y^e = y^e(m)). \tag{5.4}$$

where $\delta(.)$ is an indicator function and $y^e(m)$ is the predicted response according to model m. We will describe how to predict the perturbation response in the next section. The conditional entropy term $H(Y^e|M)$ in equation 5.3 vanishes because $P(Y^e|M)$ is deterministic. Equation 5.3 becomes

$$I(M;Y^e) = H(Y^e). \tag{5.5}$$

5.2.3 Revision of the mutual information score

Using the mutual information score in equation 5.3 to gauge the discriminative capacities of experiments implies the model discrimination procedure utilizes evidence of

both significant and insignificant changes from $\hat{Y}^e$. Candidate models are narrowed down according to the response Y^e of the entire genome under the new experiment. The models whose predicted responses (over the entire genome) are identical to the measured response are considered as new candidate models, and those which yield different vectors of predicted responses are excluded. This procedure, however, is not how we construct physical network models. As discussed in section 3.3, we ignore negative evidence when incorporating knock-out data in the model: insignificant knock-out effects (whose p-values are above the threshold) do not constrain the model variables. Neglecting negative evidence reduces the discriminative power of new experiments since it creates more degenerate models. For example, suppose two classes of model configurations predict two distinct response vectors Y^e and $Y^{e'}$. They differ only on the first gene: $Y_1^e = -1$ while $Y_1^{e'} = 0$. If the actual response under experiment e is Y^e, then these two model classes are discriminated because they yield different likelihood values. On the other hand, if the actual response is $Y^{e'}$ instead, then the likelihood values of these two model classes are identical. This is because $Y_1^{e'} = 0$ and we ignore the evidence pertaining to gene 1 when constructing the likelihood function.

In order to be consistent with our model discrimination procedure, the mutual information score should only capture the information about the significant aspect (up/down regulation) of the predicted response. We revise the mutual information score in the following way. Denote P^e as the predicted pattern of change/no change under experiment e. For instance, $P^e = (01110)$ denotes that genes 2, 3 and 4 are changed (up or down regulated) and genes 1 and 5 remain unchanged under experiment e. In addition, denote Y_P^e as the predicted responses restricted to the genes which have significant predicted responses according to P^e. For example, if $P^e = (01110)$, then Y_P^e is a three-component vector with entries ± 1, denoting the predicted response of genes 2, 3, 4. A proper revision to the mutual information score is to condition on change/no change patterns P^e and then compute the entropy reduction given the predicted response Y_P^e restricted to significantly changed genes. The information gain is the expected reduction of model entropy over the change/no

change patterns. To be precise,

$$G = \sum_p P(P^e = p)[H(M|P^e = p) - H(M|P^e = p, Y_p^e)] = I(M; Y_P^e|P^e). \tag{5.6}$$

where Y_p^e denotes the predicted response on significantly changed genes consistent with pattern $P^e = p$. The averaging probability $P(P^e = p)$ is taken over all 2^n binary (0/1) vectors of length n where n is the number of genes. The information gain is the mutual information conditioned on the change/no change pattern. Because the predicted pattern of change/no change P^e and the predicted change of genes Y_p^e consistent with $P^e = p$ are deterministic given the model configuration, equation 5.6 can be reduced to the conditional entropy $H(Y_P^e|P^e)$ and further simplified to the difference of marginal entropies $H(Y^e) - H(P^e)$:

$$G = \sum_p P(P^e = p)H(Y_p^e|P^e = p) = H(Y^e) - H(P^e). \tag{5.7}$$

The derivation of equation 5.7 is as follows. Readers who are not interested in technical details can skip the rest of this section and the next section and directly read the empirical results.

Recall that

$$\begin{aligned} &I(M; Y_P^e|P^e) = H(M|P^e) - H(M|Y_P^e; P^e) \\ &= \textstyle\sum_p P(P^e = p)[H(M|P^e = p) - H(M|Y_p^e; P^e = p)]. \end{aligned} \tag{5.8}$$

The first term of 5.8 is

$$H(M|P^e = p) = -\sum_{m \in M} P(M = m|P^e = p) \log P(M = m|P^e = p). \tag{5.9}$$

Denote $\mathcal{F}^e(p)$ as the model subclass whose predicted change/no change pattern is p. Thus

$$\begin{aligned} &\forall m \in \mathcal{F}^e(p), P(P^e = p|M = m) = 1. \\ &\forall m \notin \mathcal{F}^e(p), P(P^e = p|M = m) = 0. \end{aligned} \tag{5.10}$$

Therefore,

$$P(M=m|P^e=p)=\begin{cases}\frac{P(M=m)}{P(P^e=p)} & \text{if } m\in\mathcal{F}^e(p),\\ 0 & \text{otherwise.}\end{cases} \tag{5.11}$$

$$H(M|P^e=p)=-\sum_{m\in\mathcal{F}^e(p)}P(M=m|P^e=p)\log P(M=m|P^e=p). \tag{5.12}$$

The second term of equation 5.8 is

$$\begin{aligned}
&H(M|Y_p^e;P^e=p)=\\
&-\textstyle\sum_{m\in M,y\in\{-1,+1\}^{|p|}}P(M=m,Y_p^e=y|P^e=p)\cdot\log P(M=m|Y_p^e=y,P^e=p)\\
&=-\textstyle\sum_{m\in M,y\in\{-1,+1\}^{|p|}}P(M=m|P^e=p)P(Y_p^e=y|M=m,P^e=p)\cdot\\
&\log P(M=m|Y_p^e=y,P^e=p)\\
&=-\textstyle\sum_{m\in M,y\in\{-1,+1\}^{|p|}}P(M=m|P^e=p)P(Y_p^e=y|M=m,P^e=p)\\
&\cdot[\log P(M=m|P^e=p)+\log P(Y_p^e=y|M=m,P^e=p)-\log P(Y_p^e=y|P^e=p)]\\
&=-\textstyle\sum_{m\in M}P(M=m|P^e=p)\log P(M=m|P^e=p)\cdot\sum_{y\in\{-1,+1\}^{|p|}}P(Y_p^e=y|M=m,P^e=p)\\
&+H(Y_p^e|M,P^e=p)-H(Y_p^e|P^e=p).
\end{aligned} \tag{5.13}$$

$H(Y_p^e|M,P^e=p)=0$ because Y_p^e is deterministic given the model identity and change/no change pattern. The term $-H(Y_p^e|P^e=p)$ is what we want to keep. Thus we want to show the first term cancels out with $H(M|P^e=p)$ in equation 5.8.

Because $P(M=m|P^e=p)=0$ for $m\notin\mathcal{F}^e(p)$,

$$\begin{aligned}
&-\textstyle\sum_{m\in M}P(M=m|P^e=p)\log P(M=m|P^e=p)\sum_{y\in\{-1,+1\}^{|p|}}P(Y_p^e=y|M=m,P^e=p)\\
&=-\textstyle\sum_{m\in\mathcal{F}^e(p)}P(M=m|P^e=p)\log P(M=m|P^e=p)\sum_{y\in\{-1,+1\}^{|p|}}P(Y_p^e=y|M=m,P^e=p)\\
&=-\textstyle\sum_{m\in\mathcal{F}^e(p)}P(M=m|P^e=p)\log P(M=m|P^e=p)\cdot 1.
\end{aligned} \tag{5.14}$$

The second equality arises from equation 5.11 and the fact that

$$\sum_{y\in\{-1,0,+1\}}P(Y_p^e=y|M=m,P^e=p)=1. \tag{5.15}$$

Substituting equation 5.14 into equation 5.13 and combining it with equation 5.12,

$$H(M|P^e=p)-H(M|Y_p^e;P^e=p)=H(Y_p^e|P^e=p). \tag{5.16}$$

Thus

$$I(M; Y_P^e | P^e) = \sum_p P(P^e = p) H(Y_p^e | P^e = p). \tag{5.17}$$

$P(Y_p^e | P^e = p)$ is the marginal probability of predicted significant responses over the models in $\mathcal{F}^e(p)$. It can be expressed as

$$\begin{aligned} &P(Y_p^e = y | P^e = p) = \textstyle\sum_{m \in \mathcal{F}^e(p)} P(M = m | P^e = p) P(Y_p^e = y | M = m; P^e = p) \\ &= \tfrac{1}{P(P^e = p)} \textstyle\sum_{m \in M} P(M = m) P(Y_p^e = y | M = m). \end{aligned} \tag{5.18}$$

where y is a vector of length $|p|$ with ± 1 entries. $P(P^e = p)$ is the sum of model probabilities over the class $\mathcal{F}^e(p)$:

$$P(P^e = p) = \sum_{m \in \mathcal{F}^e(p)} P(M = m). \tag{5.19}$$

The second equality in equation 5.18 holds since

$$\forall m \notin \mathcal{F}^e(p), P(M = m | P^e = p) = 0. \tag{5.20}$$

and

$$\forall m \in \mathcal{F}^e(p), P(P^e = p | M = m) = 1. \tag{5.21}$$

Therefore, extending the summand in equation 5.18 into the model universe M does not add extra contributions except for a normalization constant. Plug 5.18 into 5.17,

$$
\begin{aligned}
&I(M; Y_P^e | P^e) = \textstyle\sum_p P(P^e = p) H(Y_p^e | P^e = p) \\
&= -\textstyle\sum_p P(P^e = p)[\sum_y P(Y_p^e = y | P^e = p) \log P(Y_p^e = y | P^e = p)] \\
&= -\textstyle\sum_p P(P^e = p)[\sum_y (\frac{1}{P(P^e=p)} \sum_{m \in M} P(M = m) P(Y_p^e = y | M = m)) \cdot \\
&\log(\tfrac{1}{P(P^e=p)} \textstyle\sum_{m \in M} P(M = m) P(Y_p^e = y | M = m))] \\
&= -\textstyle\sum_p \sum_y [\sum_{m \in M} P(M = m) P(Y_p^e = y | M = m) \cdot \log(\sum_{m \in M} P(M = m) P(Y_p^e = y | M = m))] + \\
&\textstyle\sum_p \sum_y [\sum_{m \in M} P(M = m) P(Y_p^e = y | M = m) \cdot \log P(P^e = p)] \\
&= -\textstyle\sum_p \sum_y [\sum_{m \in M} P(M = m) P(Y^e = [y, 0] | M = m) \cdot \\
&\log(\textstyle\sum_{m \in M} P(M = m) P(Y^e = [y, 0] | M = m))] + \\
&\textstyle\sum_p P(P^e = p) \log P(P^e = p) \\
&= -\textstyle\sum_z \sum_{m \in M} P(M = m) P(Y^e = z | M = m) \log(\sum_{m \in M} P(M = m) P(Y^e = z | M = m)) + \\
&\textstyle\sum_p P(P^e = p) \log P(P^e = p) \\
&= -\textstyle\sum_z P(Y^e = z) \log P(Y^e = z) + \sum_p P(P^e = p) \log P(P^e = p) \\
&= H(Y^e) - H(P^e).
\end{aligned}
\tag{5.22}
$$

where y is over binary vectors of length $|p|$ with ± 1 entries, z is over vectors of length n with $-1, 0, +1$ entries, $[y, 0]$ denotes filling the zero entries in pattern p with 0s in z. The third equality is from equation 5.18, the fifth and sixth equalities state that marginalizing over patterns and non-zero predictions together is equivalent to marginalizing over all possible predictions.

5.2.4 Approximation of the mutual information computation

Evaluating the mutual information of random variables in a high dimensional space is generally intractable for it requires enumerating an exponential number of variable configurations. The number of optimal configurations in our model can be astronomical. For example, in our previous analysis of the genome-wide molecular interaction network ([172]), we reported more than 2^{47} optimal configurations. To resolve this problem, we approximate the joint probability of output responses as the product of marginal probabilities and evaluate each marginal probability by using the sum-

product algorithm. This section describes details of the approximation method of evaluating the mutual information scores in equations 5.3 and 5.6. The readers not interested in technical details may skip it and directly read the section of empirical analysis.

The evaluation of equation 5.7 requires computing the entropies $H(Y^e)$ and $H(P^e)$. To evaluate them we need to obtain the distribution of the predicted response and pattern vectors:

$$P(Y^e = y) = \sum_m P(M = m)P(Y^e = y|M = m). \tag{5.23}$$

$$P(P^e = p) = \sum_m \sum_{y \sim p} P(M = m)P(Y^e = y|M = m). \tag{5.24}$$

where $y \sim p$ denotes the set of predicted responses y consistent with the change/no change pattern p. The marginalization is taken over the model configurations in the candidate set. It requires enumerating an exponential number of model configurations thus is generally intractable. To simplify the problem, we approximate the joint probability function with the product of marginal distributions of individual genes. Recall that $Y^e = (Y_1^e, \cdots, Y_n^e)$ is a vector of the predicted response of n genes. We first approximate the joint distribution $P(Y^e)$ with the product of marginal distributions $\tilde{P}(Y^e)$:

$$\tilde{P}(Y^e) = \prod_{i=1}^{n} P(Y_i^e). \tag{5.25}$$

$\tilde{P}(Y^e)$ is known to be the projection of $P(Y^e)$ on the space of independent distributions $\mathcal{F}_0$:

$$\tilde{P}(Y^e) = \arg \min_{Q(Y^e) \in \mathcal{F}_0} D_{KL}(P(Y^e)|Q(Y^e)). \tag{5.26}$$

The mutual information and revised mutual information are approximated by the sum of marginal entropies:

$$H(Y^e) \leq \sum_{i=1}^{n} H(Y_i^e), H(P^e) \leq \sum_{i=1}^{n} H(P_i^e). \tag{5.27}$$

$$H(Y^e) - H(P^e) \approx \sum_{i=1}^{n} [H(Y_i^e) - H(P_i^e)]. \tag{5.28}$$

Notice the approximation in 5.28 may not be the upper bound of the actual value of the subtraction of the two terms. By definitions of Y^e and P^e,

$$H(Y_i^e) = - \sum_{y_i=\{-1,0,+1\}} P(Y_i^e = y_i) \log P(Y_i^e = y_i). \tag{5.29}$$

and

$$H(P_i^e) = -P(Y_i^e = 0) \log P(Y_i^e = 0) - P(Y_i^e = \pm 1) \log P(Y_i^e = \pm 1). \tag{5.30}$$

Subtracting 5.30 from 5.29, and substituting the result into 5.28,

$$\begin{aligned} & I(M; Y_P^e | P^e) = H(Y^e) - H(P^e) \\ & \approx - \textstyle\sum_{i=1}^{n} \sum_{y=-1,+1} P(Y_i^e = y) \log(\frac{P(Y_i^e = y)}{P(Y_i^e = \pm 1)}). \end{aligned} \tag{5.31}$$

The marginal probability of the predicted response in gene i is

$$P(Y_i^e = y_i^e) = \sum_m P(M = m) P(Y_i^e = y_i^e | M = m). \tag{5.32}$$

Evaluating $P(Y_i^e)$ is the inference problem of graphical models thus can be efficiently approximated by the sum-product algorithm without enumerating all model configurations. Denote $X = (X_1, \cdots, X_N)$ as variables in the physical network model. A model configuration m is an instantiation of values in X. Denote this instantiation as $(X_1 = x_1(m), \cdots, X_N = x_N(m))$. The probability of a model configuration is proportional to the joint likelihood value evaluated at this configuration. Recall that we express the joint likelihood function as the product of potential functions.

$$P(M = m) \propto \mathcal{L}(X_1 = x_1(m), \cdots, X_N = x_N(m)) = \prod_j \phi_{C_j}(X_{C_j} = x_{C_j}(m)). \tag{5.33}$$

where $\phi_{C_j}(.)$ is a potential function pertaining to a constraint from physical interaction

or knock-out data. By substituting 5.33 into 5.32,

$$P(Y_i^e = y_i^e) \propto \sum_m P(Y_i^e = y_i^e | M = m) \prod_j \phi_{C_j}(X_{C_j} = x_{C_j}(m)) = \\ \sum_x P(Y_i^e = y_i^e | X = x) \prod_j \phi_{C_j}(X_{C_j} = x_{C_j}). \tag{5.34}$$

The summation is taken over all configurations of X. We augment the joint likelihood function $\mathcal{L}$ by inserting a term $P(Y_i^e|X)$:

$$\mathcal{L}'(X, Y_i^e) = \mathcal{L}(X) P(Y_i^e | X). \tag{5.35}$$

The augmented likelihood function contains variables $X \cup Y_i^e$. Equation 5.34 evaluates the marginal belief of Y_i^e under the new model:

$$P(Y_i^e = y_i^e) = \frac{1}{Z} \sum_X \mathcal{L}'(X, Y_i^e). \tag{5.36}$$

The normalization constant Z is immaterial since we can do normalization after the inference. The marginal belief of a single variable can be efficiently approximated by various algorithms such as sum-product of factor graphs ([97]) and generalized belief propagation ([173]). In this thesis, we implemented the sum-product algorithm for the inference. The sum-product algorithm is described in Chapter Three.

The term $P(Y_i^e|X)$ represents the prediction of Y_i^e according to model configuration X. Similar to the potential terms of knock-out explanation, we adopt a potential term for model prediction:

$$P(Y_i^e = y_i^e | X = x) = \begin{cases} 1 & \text{if } y_i^e = \text{the prediction from model configuration } x, \\ 0 & \text{otherwise.} \end{cases} \tag{5.37}$$

To predict the response of gene g_i by deleting gene g_e, we first identify all candidate paths (the paths satisfying conditions 1-4 in section 3.3.3) connecting g_e to g_i. For each path, we then verify the following conditions: whether all edges are present, whether all edges appear in some active paths that explain existing knock-out effects, and whether directions are consistent according to the given configuration. If they

do then we predict the positive or negative response according to edge sign configurations along the path. Otherwise the predicted response along this path is that the downstream gene does not change. If all valid predictions (which predict positive or negative changes) along connecting paths agree, then we predict g_i is up or down regulated, otherwise we predict g_i does not change.

We express the predictions formally with potential functions as follows. Suppose valid paths $\pi_{i1}, \cdots, \pi_{iN}$ connect g_e to g_i, Let X_{ij}, D_{ij}, S_{ij} be the variables of edge presence, edge direction and edge sign along the path π_{ij}. Without loss of generality we define all edge directions consistent with the prediction to be 1. Denote $Y_{i1}, \cdots, Y_{iN}$ to be the predicted responses of g_i along each path. We first construct the potential term for each path prediction:

$$\phi_{ij}(Y_{ij}, X_{ij}, D_{ij}, S_{ij}) = \begin{cases} 1 & \text{if } (\forall x \in X_{ij}, x = 1) \cap (\forall d \in D_{ij}, d = 1) \cap (\prod_{s \in S_{ij}} s = -Y_{ij}), \\ 1 & \text{if } ((\exists x \in X_{ij}, x = 0) \cup (\exists d \in D_{ij}, d \neq 1)) \cap (Y_{ij} = 0), \\ 0 & \text{otherwise.} \end{cases} \tag{5.38}$$

The first scenario denotes the configuration along the path to predict a significant change of Y_{ij}. The second scenario denotes conditions for predicting a significant change are violated and the predicted response is no change. The third scenario states all cases when the predicted response is inconsistent with Y_{ij}.

Predictions along paths are then combined to give an overall prediction Y_i of g_i response. The potential function of prediction aggregation is

$$\psi_i(Y_i, Y_{i1}, \cdots, Y_{iN}) = \begin{cases} 1 & \text{if } (Y_i = +1) \cap (\forall j, Y_{ij} \in \{0, +1\}) \cap ((Y_{i1}, \cdots, Y_{iN}) \neq (0, \cdots, 0)), \\ 1 & \text{if } (Y_i = -1) \cap (\forall j, Y_{ij} \in \{0, -1\}) \cap ((Y_{i1}, \cdots, Y_{iN}) \neq (0, \cdots, 0)), \\ 1 & \text{if } (Y_i = 0) \cap (\text{all other configurations}), \\ 0 & \text{otherwise.} \end{cases} \tag{5.39}$$

The first scenario states the condition when the aggregate prediction is positive: predictions along all paths are either positive or zero, but not all zeros. The second scenario states the condition when the aggregate prediction is negative in a similar

fashion. The third scenario states all other conditions that the aggregate prediction is no change. The fourth scenario occurs when Y_i contradicts with the aggregate prediction.

Usually only a relatively small number of genes are connected to the deleted gene via valid paths in the physical network. We identify those genes and incorporate only the variables relevant to their predictions in the new potential functions. Other genes and variables will not affect the mutual information score. This pre-processing step greatly simplifies the inference procedure.

5.3 Empirical results on existing datasets

We applied the experimental design framework to large-scale datasets of physical interactions and knock-out gene expression including high-throughput chromatin IP data [100], yeast protein-protein interaction database [31] and Rosetta Compendium knock-out data [80]. Deletion experiments were ranked in terms of their mutual information scores. We first performed leave-one-out cross validation analysis of the Rosetta data. The hold-out experiments whose data are constrained by other experiments are on the top-ranking list, suggesting the mutual information score is a sensible metric for model discrimination. We then established the physical network model on the entire datasets and ranked new deletion experiments according to their mutual information scores. We investigated the subnetworks associated with several top-ranking experiments and qualitatively justified the importance of these experiments. Finally we evaluated the accuracy of inferred attributes (with respect to an artificially chosen reference model) by incrementally adding data from suggested experiments. The results – the learning curve analysis – demonstrated using the information gain for experimental design outperformed straightforward approaches such as choosing deleted genes randomly or based on the number of connections.

5.3.1 Cross validation tests on Rosetta data

The cross validation test was performed in the following way. We first held out a deletion experiment and removed the knock-out interactions associated with the experiment when building the physical network model. The mutual information scores of all single deletion experiments outside the training set were computed, and those experiments (including the leave-out experiment from Rosetta data) were ranked according to their information scores. We then checked the rank and the mutual information score of the leave-out experiment. This procedure was repeated for all single deletion experiments in Rosetta data.

To validate the mutual information score, we need an external metric of the importance of a deletion experiment which was already performed. We built the physical network model from the entire Rosetta data and counted the number of knock-out interactions from each experiment which were explained by an optimal model configuration. We chose the number of explained knock-out effects as the external metric for an experiment because it reflected the constraints imposed on the model.

The Rosetta Compendium data contain knock-out experiments of 253 single deletions (some deletion experiments were repeated). Among them only 64 experiments have knock-out interactions connected by valid paths of length ≤ 3 in the physical network. Furthermore, among the 64 experiments, only 24 contain knock-out effects which are predictable from the data of other knock-out experiments. This means the edges connecting the knock-out interactions in these experiments are utilized at least once to explain the knock-out interactions from other experiments. The mutual information scores and the rankings of these 24 knock-out experiments in the cross-validation setting are shown in Table 5.1.

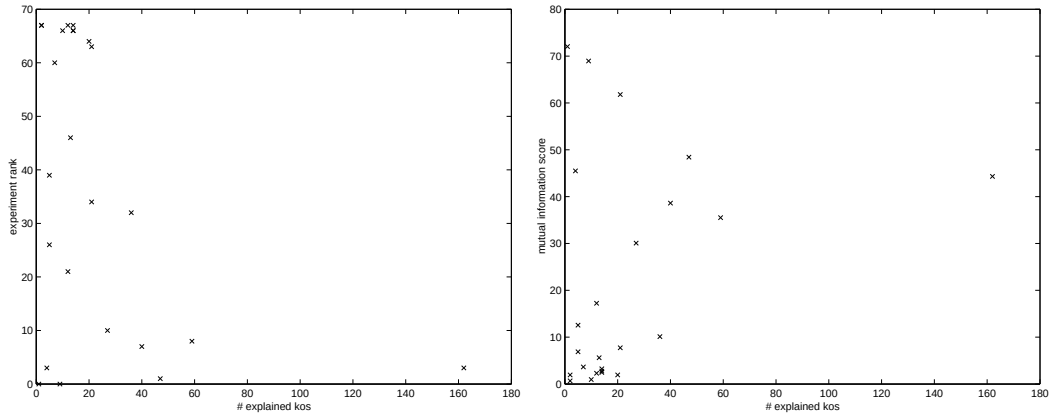
Figure 5-3 plots the number of explained knock-out pairs versus the rank and the mutual information score of the 24 experiments. Among the 9 experiments with $\geq$ 20 knock-out interactions explained by the model, 6 of them are ranked within top 10. The 3 experiments which have ≥ 20 explainable knock-out interactions but are ranked low – Cmd1Δ, Ckb2Δ, Yap1Δ – reflect the false negatives of model prediction

Table 5.1: Cross validation tests on Rosetta gene deletion experiments

Experiment	Entropy	Rank	# explained knock-outs
Fus3Δ	72.033761	0	1
Kss1Δ	68.950833	0	9
Swi6Δ	61.792096	0	21
Sin3Δ	48.420267	1	47
Mbp1Δ	45.491117	3	4
Swi4Δ	44.319638	3	162
Gcn4Δ	38.612689	7	40
Ssn6Δ	35.511490	8	59
Ste12Δ	30.079665	10	27
Dig1Δ	17.243579	21	12
Arg80Δ	12.563430	26	5
Ckb2Δ	10.129493	32	36
Yap1Δ	7.727912	34	21
Mac1Δ	6.874793	39	5
Swi5Δ	5.622311	46	13
Ade2Δ	3.652650	60	7
Cmd1Δ	1.933891	64	20
Qcr2Δ	3.254703	66	14
Pma1Δ	2.758714	66	14
YER083CΔ	0.957696	66	10
Clb2Δ	2.474916	67	14
Sst2Δ	2.301133	67	12
Kin3Δ	1.936279	67	2
Rnr1Δ	0.677027	67	2

or the false positives of model explanation. A number of knock-out interactions are connected from these genes to their downstream targets via valid paths, thus these knock-out effects can be explained when they are incorporated in the model. However, the paths connecting the deleted genes and the downstream target genes are sparsely utilized to explain the knock-out effects in other experiments. This can be due to the fact that these paths are sparsely probed (deleted) in the Rosetta data, or that these paths are indeed not active and the knock-out effects are resulted from the mechanisms not captured by the current physical interaction datasets. Conversely, among the 15 experiments with less than 20 knock-out interactions explained by the model, only 3 of them – Fus3, Kss1, Mbp1 – are ranked within top 20. These cases reflect the false positives of model prediction or the false negatives of model explanation. Many genes are predicted to change in these knock-out experiments but only few of them are changed in the real data. These anomalies can be due to the parallel pathways connecting the target genes. Both Fus3 and Kss1 phosphorylate Ste12 under different environmental conditions ([108]), but they also function in a complementary fashion. Deleting either Fus3 or Kss1 causes few changes in Ste12-

Figure 5-3: Cross validation tests on Rosetta gene deletion experiments



controlled genes while the double deletion significantly affects those genes ([46]). The protein complexes Mbp1-Swi6 and Swi4-Swi6 co-appear on the promoters of many genes and they are also known to serve parallel functions ([39, 139]). However, from our analysis of the Rosetta data Swi4Δ seems to induce more changes than Mbp1Δ. This suggests other functional roles of Swi4.

To sum up, cross validation results show that the mutual information scores faithfully reflect the relevance of a deletion experiment with respect to other experiments. False positives (experiments which are predicted to be important but not) are often caused by parallel pathways, whereas false negatives (experiments which are predicted to be unimportant but are) are caused by sparse constraints along the pathways.

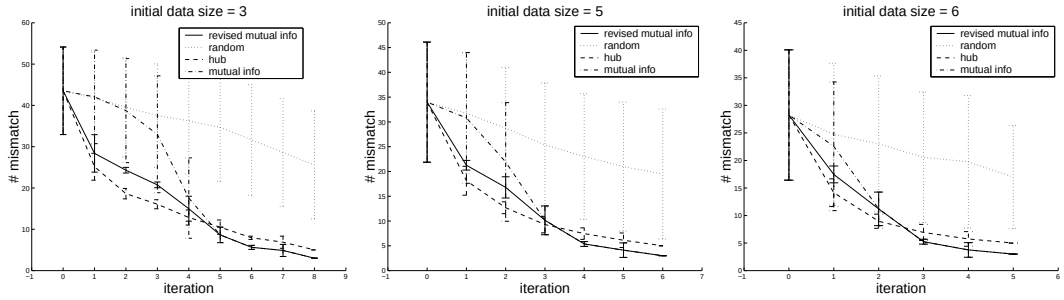
5.3.2 Analysis on suggested experiments

Cross validation test results suggest that the experimental design scheme picks up the experiments constrained by other knock-out data even though the data from these experiments are hidden. It is more convincing to show the new experiments suggested by the information scores are important. Although the importance of a new experiment cannot be really confirmed until it is performed, we can argue internally why those experiments are selected according to the inferred models from the current datasets. To fulfill this goal we applied the experimental design scheme to the entire

Table 5.2: Top ranking experiments for model discrimination

Gene	Rank	Score	Lethal	Gene	Rank	Score	Lethal	Gene	Rank	Score	Lethal
Hhf1	1	52.1429	no	Nrg1	11	31.6501	no	Adh2	21	18.2773	no
Mcm1	2	50.2682	yes	Fkh1	12	29.1195	no	Ace2	22	16.8774	no
Fhl1	3	47.0964	yes	Fkh2	13	26.7131	no	Rap1	23	16.1825	yes
Srb4	4	46.4398	yes	Slt2	14	23.4728	no	Cup9	24	14.7758	no
Sok2	5	45.0279	no	Msn4	15	21.8224	no	Gcd2	25	14.0608	yes
Cka1	6	45.0075	no	Ino4	16	21.8105	no	Gal4	26	14.0602	no
A2	7	40.9023	no	Reb1	17	21.0964	yes	Rlm1	27	12.7152	no
Abf1	8	40.0967	yes	Ckb1	18	19.0418	no	Htb1	28	12.6566	yes
Yap6	9	35.1652	no	Srp1	19	18.9938	yes	Vps1	29	12.6547	no
Ndd1	10	34.5169	yes	Hta1	20	18.9790	no	Rfx1	30	12.1417	no

Figure 5-4: Learning curves of four experimental selection criteria



Rosetta data and ranked experiments which did not appear in the Rosetta data. Table 5.2 shows the top 30 experiments according the revised mutual information scores. We also included the lethality information to indicate the feasible experiments.

By referring to the subnetworks generated by the model decomposition algorithm in [172], we can explain why these experiments are chosen. Hhf1 and Fhl2 are along the pathways of explaining the knock-out effects in Tup1Δ experiment (Figure 4-8.2). Sok2 is an important hub mediating the paths from Swi4 to many genes down-regulated in Swi4Δ experiment (Figure 4-8.1). Nrg1 is on the pathway of explaining the knock-out effects in Tup1Δ and Ssn6Δ experiments (Figure 4-8.3). The deletion of those genes help narrowing down consistent edge sign configurations along these paths. To sum up, all the top-ranking experiments probe "hub" genes on subnetworks whose model configurations are uncertain.

5.3.3 Learning curve analysis

Although the cross validation test and the analysis on suggested experiments justify the importance of selected experiments, they do not give a direct and quantitative evaluation on the experimental design algorithm. A common approach of evaluating an active learning method is to incrementally add data points according to the active learning criteria and measure the performance of the learned model at each iteration according to some loss function. This procedure is called learning curve analysis. To evaluate the performance of our experimental design approach, we compared the learning curves generated by four methods of choosing new knock-out experiments. The results of learning curve analysis are shown in Figure 5-4.

We performed the analysis on a subnetwork of yeast mating pathway introduced in Section 4.1. There are 8 experiments from Rosetta data whose knock-out effects can be explained by the physical network model. We have inferred the physical network model of the yeast mating pathway from those three datasets in Chapter Four. There are 4 configurations of model variables which explain the data equally well. We chose one of those optimal configurations as the reference model.

We obtained the learning curves of the four methods with the following procedure. For each initial subset of knock-out experiments, we tracked the history of the learning performance by incrementally adding data from newly selected experiments. As more data were incorporated, the inferred model would converge to the reference model regardless of the choice of experiments. However, a better criterion for choosing experiments should yield a faster convergence. We chose initial sets of 3, 5 and 6 knock-out experiments and enumerated all combinations of these initial sets from the 8 relevant knock-out experiments. For each initial set, the incremental learning procedure was performed. At each iteration, a model was inferred from the current dataset. The loss function of an inferred model was the number of undetermined variables in the inferred model plus the number of variables whose values mismatched the values in the reference model. We then chose a new experiment according to four different methods. The first method ranked experiments in terms of their revised mutual infor-

mation scores (equation 5.7), and selected the top-ranking experiment. The second method randomly chose a gene which was possible to affect the transcription of other genes according to the molecular cascade hypothesis. In other words, a candidate gene linked to a downstream gene in the subnetwork via valid pathways introduced in Chapter Three. The third method ranked genes according to their connectivity in the physical network and chose the top-ranking gene which was not yet incorporated. The connectivity was defined as the number of edges emanating from a gene. This excluded the protein-DNA interactions incident to the gene because the deletion of a downstream gene did not affect its upstream. The fourth method ranked experiments according to approximated typical mutual information (equation 5.3). The knock-out data of the selected experiment were then incorporated separately for the four methods. If the data of the new experiment were available in Rosetta data, then the real data were incorporated. Otherwise hypothetical data were generated by the predicted responses of the new experiment according to the reference model. The incremental learning proceeded until the mutual information criteria did not suggest new experiments (in other words, the mutual information scores of all left experiments were zero).

The incremental learning procedure was carried out over all possible combinations of initial datasets. There are $\binom{8}{3} = 56$, $\binom{8}{5} = 56$ and $\binom{8}{6} = 28$ initial datasets with 3, 5 and 6 experiments. We show the means and standard deviations of these curves in Figure 5-4. Clearly, both schemes of hub selection and revised mutual information criterion significantly outperform the random selection scheme: both the means and the variances are smaller in the non-random schemes. Moreover, the performance of the revised mutual information criterion improves as more data are incorporated in the model. Hub selection performs better than the revised mutual information criterion during the first half of the learning curve. As the incremental learning proceeds, the mutual information criterion catches up with the hub selection and eventually outperforms it. However, the difference between random selection and typical mutual information scores is small during the early iterations. This is because mutual information scores are relevant only when current models contain certain

amount of information regarding the true model. Using revised mutual information scores outperforms the random selection even at early iterations. This is probably because using the revised mutual information can avoid choosing irrelevant deletions (that would yield equal probability for +1, -1 and 0 predictions) in the beginning and directly focuses on the experiments which would generate many significant changes but the predicted changes are uncertain. The typical mutual information score, in contrast, would rank irrelevant deletions high at early stage since they would yield uncertain responses in terms of changes/no changes.

The difference of the performance between the mutual information criterion and hub selection can be understood from the properties of the physical network models. When the available dataset is small, the inferred model has weak predictive power hence it does not help to choose the critical experiment at the next step. As the learning proceeds, the inferred model contains more information about the underlying model, thus it can help to select the informative experiments which at best discriminate the current model. In contrast, connectivity becomes less important as we acquire more information about the underlying process, since many physical interactions may not carry functional roles in gene regulation. This observation is in line with the typical performance of a *myopic* active learning procedure, where the inferred model is assumed to be correct when generating the next experiment. As inferred models are closer to the true models, mutual information scores can give a better answer to select optimal experiments.

This difference suggests a more efficient way of selecting experiments for model discrimination. When the available dataset is small and the target system is relatively unprobed, the best strategy is to follow a simple yet intuitive criterion such as selecting hubs. As more information is obtained from the updated dataset, the simple method is preferred to a quantitative experimental design scenario which utilizes the updated information. The "switching point" between the naive and the elaborated schemes can be decided by the learning curve analysis.

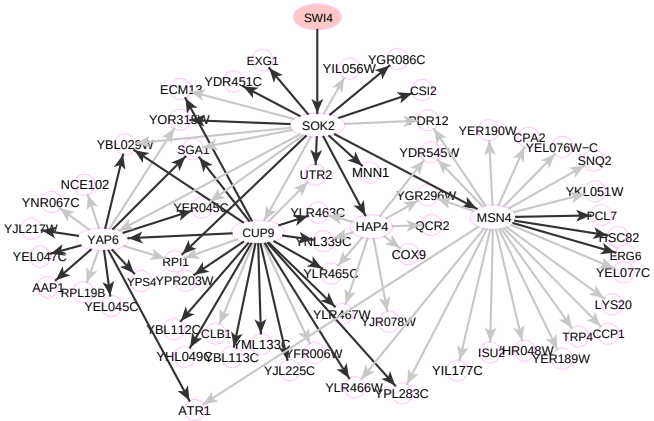
5.4 Analysis of new experimental data

The "wet experiments" are the primary (or perhaps the only) link of computational models to the real world. The discussion about experimental design is incomplete without covering the analysis of new data from suggested experiments. In order to complete the process of scientific inquiry, we collaborated with the biologists at the Ideker Laboratory at Unversity of California, San Diego to perform some suggested experiments and analyze the new data. In this section I will describe the computational methods of analyzing the new data as well as their biological findings.

5.4.1 Selection of experiments

The top-ranking experiments in Table 5.2 are prioritized according to their capacity of reducing the uncertainty of model configurations. However, if we are allowed to perform n knock-out experiments ($n > 1$), selecting the top n experiments on the list is not necessarily the best strategy. First, we have to exclude deleting lethal genes for they require more careful and costly treatment such as making temperature sensitive mutants. Second, the revised mutual information score of equation 5.8 is for single experiments. It is possible to extend the same framework to batch experiments, but the evaluation of the mutual information scores of all batch experiments would be time-consuming. Therefore, we adopt a less formal way of selecting batch experiments. There are two basic strategies for selecting batch experiments. One can focus on deleting genes within the same subnetwork or probe different parts of the entire network. Although the learning curve analysis in Section 5.3.3 suggests the latter is a better strategy when the model is very sparsely constrained, it needs a large number of scattered perturbations in order to narrow down any subnetwork configuration within a useful range. In contrast, the former strategy may be risky at the initial stage because the target subnetwork may be a false positive, but it can lead to a complete characterization of a subsystem. We decide to adopt the former strategy to obtain a complete characterization of a subnetwork rather than very limited characterizations of many subnetworks.

Figure 5-5: Subnetwork deciphered by Sok2Δ



Following these criteria we selected the deletions of Sok2 and several other genes in the same subnetwork. Sok2 ranks the fifth in Table 5.2 and is the second among all non-lethal deletions. Although Hhf1 is non-lethal and has better ranking, the subnetwork deciphered by its deletion (Figure 4-8.2) contains protein-protein interactions (Tup1,Hhf1), (Fhl1,Hhf1) reported from a high-throughput assay. Thus the validity of this subnetwork is less confident. Figure 5-5 shows the subnetwork disambiguated by Sok2 deletion. It contains purely protein-DNA interactions and explains 65 knock-out interactions in Rosetta Swi4Δ experiment. We can view this subnetwork as a concatenation of 4 pathways connecting Swi4 to the downstream genes affected in Swi4Δ: Swi4 → Sok2 → Msn4, Swi4 → Sok2 → Hap4, Swi4 → Sok2 → Yap6, and Swi4 → Sok2 → Cup9. Uncertainty of model configurations arises from the freedom of adjusting edge signs along these pathways to fit the knock-out effects. To completely determine edge signs in this subnetwork, 5 new knock-out experiments are needed: Sok2, Msn4, Hap4, Yap6 and Cup9. We also applied the mutual information scores restricted to the genes in this subnetwork and found they were the top 5 experiments within this restricted set. Therefore, we chose to perform all these 5 knock-out experiments. The post processing of experimental data indicated the qual-

Table 5.3: Top-ranking repeated experiments

Gene	Function
Swi4	cell cycle regulator
Ssn6	general repressor of RNA polII
Rpd3	histone deacetylase
Swi6	cell cycle regulator
Tup1	general repressor of RNA polII
Med2	component of RNA polII holoenzyme
Sin3	component of RNA polII holoenzyme
Ckb2	casein kinase
Gcn4	amino acid synthesis control
Gln3	amino acid synthesis control

ity of Cup9Δ data was unsatisfactory. Therefore, we only analyzed the data from the remaining 4 deletion experiments.

In addition to new knock-out experiments, we also repeated several deletion experiments which appeared in the Rosetta data in order to verify the reproducibility of knock-out expression data. We chose the repeated experiments according to their contribution to the explanatory power of the physical network model. The contribution of an existing experiment is the number of knock-out interactions which are explained by the paths containing the deleted gene in this experiment as the intermediate or terminal gene. For instance, Ste11 and Ste12 are along the pathway connecting to mating response genes, and Ste11 is the upstream of Ste12. If those genes did not change in Ste12Δ, then this pathway could not explain the knock-out effects in Ste11Δ due to the conditions of pathway explanation. Intuitively, a Rosetta experiment is chosen if it affects many genes which connect to the deleted gene via valid pathways. Table 5.3 enlists the top-ranking repeated experiments. We chose Swi4Δ and Gcn4Δ as the repeated experiments. Swi4Δ ranks top on the list. Although Gcn4Δ only ranks the ninth, other higher ranking genes either belong to the general transcription apparatus (Ssn6, Tup1, Med2, Sin3), have an overlapped function with Swi4 (Swi6), or explain knock-out effects via pathways containing high-throughput protein-protein interactions (Rpd3, Ckb2). Notice Swi4 is also the key gene in the Sok2 subnetwork.

Table 5.4: Summary statistics of comparing repeated experiments

	Swi4 total	Gcn4 total
corr.	-0.013	0.340
corr. pval	0.84	$< 10^{-4}$
rank corr.	-0.058	0.145
rank corr. pval	1.0	$< 10^{-4}$
hyper-geom. pval	0.188	4.88×10^{-21}
	Swi4 subset	Gcn4 subset
corr.	0.344	0.831
corr. pval	8×10^{-5}	$< 10^{-4}$
rank corr.	0.353	0.705
rank corr. pval	9×10^{-5}	$< 10^{-4}$
hyper-geom. pval	1.05×10^{-5}	1.80×10^{-13}

5.4.2 Analysis of repeated experimental data

The goal of comparing the knock-out data in Swi4Δ and Gcn4Δ between the two datasets is to verify the reproducibility of the knock-out gene expression data. Because the annotations of the physical network are inferred from the knock-out interactions of high-throughput gene expression data, the inferred annotations are meaningful only if the knock-out data are reliable. Ideally, the expression data generated from two laboratories should be similar, and the significant knock-out effects should also be robust across the two repeated experiments.

In reality, these two datasets are very dissimilar at genome scale. The Pearson correlation coefficient across 5901 genes is 0.34 and the rank correlation coefficient is 0.145 between the two Gcn4 datasets. For Swi4, the correlation between the two datasets is even lower. The Pearson correlation coefficient is -0.013 and the rank correlation coefficient is -0.058. By thresholding the reported p-values (Rosetta data $p \leq 0.02$, new data $p \leq 0.05$) and counting the overlap of up and down regulated genes, the two datasets are also dissimilar; the hyper-geometric p-value of the overlap in Swi4Δ is 0.188, while the hyper-geometric p-value in Gcn4Δ is very significant (4.88×10^{-21}). Table 5.4 summarizes the statistics of the comparison results.

The disparity between datasets generated from repeated experiments seems to be universal for a variety of high-throughput assays. Examples are reported in DNA mi-

croarrays ([2, 127]), CHIP-chip experiments ([100, 78]) and yeast two-hybrid systems ([160, 86]). In each example, the overlapped fraction of up/down regulated genes or significant physical interactions is very small. This disparity can be attributed to many causes such as the fluctuations of experimental conditions, measurement noise, variations of specimen, and human factors. Overall the disparity reflects the improvement space of high-throughput assays in order to achieve the quality control level similar to mature technologies such as semiconductor fabrication.

One plausible explanation for the disparity in our case is the mixture of the responses to target perturbations and fluctuations from irrelevant factors. The actual changes induced by perturbing the target regulatory systems are buried in these fluctuations. This explanation is consistent with the observation that Gcn4Δ datasets are more similar than Swi4Δ datasets. Gcn4 is a master regulator of amino acid metabolism which is known to affect many genes ([115]). Hence deleting Gcn4 should change a large number of genes. In contrast, the function of Swi4 is partially complemented by Swi6 ([39, 139]). Therefore, its deletion may induce less significant responses and these responses are more likely to be overwhelmed by noise.

To test this hypothesis, we compared the two datasets on the subsets of genes putatively regulated by each factor. For Gcn4 we chose 32 genes that were bound by Gcn4 in location analysis and that were down regulated in Rosetta data. For Swi4 we chose 82 genes which were either bound by Swi4 or were in the Sok2 subnetwork (Figure 5-5) and down regulated in the Rosetta data. These genes are likely to experience significant changes in Swi4 or Gcn4 deletions, thus they should yield more consistent changes in the repeated experiments. The lower part of Table 5.4 shows the comparison results on restricted subsets. The two deletion experiments are much more strongly correlated in the restricted subsets: the Pearson correlation coefficient is 0.831 in Gcn4Δ and 0.344 in Swi4Δ. Since correlation coefficients are sensitive to the data size, we calculated the p-values by randomly permuting the data and compared the results with the whole genome correlation. The two datasets are still significantly similar in the restricted subsets. Furthermore, hyper-geometric tests indicate a significant fraction of genes in the restricted set are up or down regulated

in repeated experiments.

5.4.3 Analysis of deletion data in Sok2 subnetwork

We analyzed the data of five knock-out experiments: Swi4Δ (from the repeated deletion experiment), Sok2Δ, Msn4Δ, Hap4Δ and Yap6Δ. The purpose of analyzing this data is two-fold. First, we want to validate or falsify the three pathways (Swi4 → Sok2 → Msn4, Swi4 → Sok2 → Hap4, Swi4 → Sok2 → Yap6) in the subnetwork shown in Figure 5-5. Second, we want to uniquely determine the edge sign configurations in the subnetwork. We will show that these two goals are inter-related and hence need to be considered together. We then revise the original models according to the analysis results.

We validate a pathway by showing that downstream genes experience significant and coherent changes by deleting each intermediate gene along the pathway. This criterion follows from the conditions of explaining a knock-out interaction using molecular cascades in Section 3.3.3. Each intermediate gene along the pathway is a necessary component for the regulation of downstream genes, hence perturbing each intermediate gene changes the downstream genes. The combinatorial effects on downstream genes (for example, redundant pathways) may invalidate some of the "true" pathways but do not nullify the pathways that pass this test. In addition, we assume the downstream transcription factors (Msn4, Hap4, Yap6) possess single functions (either activators or repressors). Thus the changes in the most downstream genes are expected to be coherent: they experience either all positive or all negative changes in each deletion experiment. Notice we do not require an intermediate gene to change significantly in the deletion of its upstream gene. For example, deleting Swi4 may not change Msn4. This property also arises from the condition of pathway explanation in Section 3.3.3. Intermediate genes are either signal transduction proteins whose activities are not modulated by mRNA abundance or transcription factors whose activities are very sensitive to protein quantities. Therefore, their activity changes may be very small in the absolute scale and are below the detection threshold of microarray technologies.

To quantitatively validate or falsify a pathway, we need to select the downstream genes of this pathway and then evaluate the significance of coherent changes on these downstream genes in each experiment. The downstream genes are putatively regulated by the downstream transcription factors (Msn4, Hap4, Yap6). We use different criteria to generate three sets of genes for each pathway. The first set contains genes which are bound by a downstream transcription factor according to CHIP-chip data and are significantly changed in Rosetta Swi4Δ. In other words, the set contains downstream genes in Figure 5-5. The second set contains all genes bound by a downstream transcription factor. The third set contains genes which are putatively regulated by a transcription factor according to previous studies.

A straightforward approach of evaluating the coherence of expression data is to check whether all genes downstream of the pathway exhibit significant changes in same the direction for each experiment. A hypothetical example is that all genes downstream of Hap4 are significantly down regulated in Hap4Δ, up regulated in Sok2Δ, and up regulated in Swi4Δ. This strong condition holds only for few genes: among the 98 genes bound by Hap4, only 6 have significant changes in all three deletion experiments, the fraction of significant genes is also small for genes bound by Msn4 (4 out of 74). Table 5.5 summarizes the number of genes bound by Msn4, Hap4 and Yap6 which experience significant and consistent changes in Rosetta Swi4Δ and new Swi4Δ, Msn4Δ, Hap4Δ and Yap6Δ experiments. In each pathway, only a small fraction of downstream genes are significantly and consistently changed in all deletion experiments along the pathway. Therefore, the stringent condition requiring that individual genes experience significant changes in all deletion experiments would falsify all three pathways in the subnetwork. However, as seen in the comparison of repeated experiments, high-throughput gene expression data are very noisy. The expression changes of individual genes are subject to fluctuations and thus are difficult to draw conclusions from. Instead, we evaluate the aggregate properties which are less sensitive to noise. Rather than setting the stringent criterion that all genes experience significant and coherent changes in all experiments, we ask whether a gene set as a whole has the propensity of significant changes in each experiment. This aggregate

Table 5.5: Consistency of two Swi4Δ data in Sok2 subnetwork

Sok2 subnetwork			
# genes	337	consistent and significant changes	23
Swi4→Sok2→Msn4 pathway			
# downstream genes	74	consistent and significant changes	4
Swi4→Sok2→Hap4 pathway			
# downstream genes	98	consistent and significant changes	6
Swi4→Sok2→Yap6 pathway			
# downstream genes	120	consistent and significant changes	8

perspective calls for a method of measuring the coherence of gene expression data on a set of genes.

Different methods of measuring the coherence of gene expressions can be used – for instance, correlation coefficients or p-values of hyper-geometric tests. Here we adjust the method adopted by Ideker et al. to identify the subnetworks which are active under certain experiments ([82]). We assume the p-values of expression changes are provided in the datasets. They reflect the significance of measured changes and can be computed by different error models. The p-values in our data are computed by assuming Gaussian additive noise on measurement data. Details can be found in [82]. Given a subset of genes G under a specific condition e, the log ratio of expression changes x_{ie} and the p-value of these changes p_{ie} are provided (i denotes gene index and e experiment index). We convert the p-values into the z-scores by applying the inverse Gaussian cumulative distribution function:

$$z_{ie} = \Phi^{-1}(1 - p_{ie}). \tag{5.40}$$

We are interested in the coherence of expression changes in a certain direction. Hence we compute the average z-score over a subset of genes. The average directional z-score over a subset of n genes is the average z-score weighted by their directions of changes. To avoid an insignificant response contributing a large negative value to the

average z-score, we consider only the significant responses where $z_{ie} > 0$:

$$z(G, e, d) = \frac{1}{n} \sum_{i=1}^{n} \delta(z_{ie} > 0) \text{sgn}(x_{ie} \cdot d) z_{ie}. \tag{5.41}$$

An absolute measure of expression coherence such as the average z-score is difficult to interpret and compare. Instead, we use the relative measure of p-values to compare the coherence on our target gene set to random sets of genes of the same size. We randomly selected gene groups of the same size of G and counted the fraction of random trials whose average z-scores were greater than the empirical value. This fraction is the p-value against random sets of genes,

$$p(G, e, d) = \frac{1}{N} \sum_{G_i random, |G_i|=G} I(z(G_i, e, d) > z(G, e, d)). \tag{5.42}$$

To ensure that the significant and coherent responses were specific to the downstream genes in the model, we also compared the expression coherence significance of downstream genes in each pathway versus genes downstream of unrelated transcription factors. For each transcription factor, we evaluated the expression coherence p-value on genes bound by it. We then checked the rank of expression coherence p-value of the target gene set among the p-values of all transcription factors. The expression coherence on the target gene set was considered significant if the p-value ranked high on the list. The rationale was that a significant portion of genes bound by a transcription factor were regulated by the factor. Hence these genes would exhibit significant and coherent changes if the transcription factor activity were altered. Due to the lack of evidence from physical interaction and Rosetta knock-out data, we assumed transcription factors outside the target pathway were not affected by all deletions along the pathway. Consequently, the genes bound by other factors were not expected to have significant and coherent changes in all deletion experiments.

Table 5.6 summarizes the testing results on three pathways: Swi4 → Sok2 → Msn4, Swi4 → Sok2 → Hap4, Swi4 → Sok2 → Yap6. The table shows the coherence p-values, ranks and the directions of changes in the downstream genes of these pathways

Table 5.6: Expression coherence on genes bound by factors in Sok2 subnetwork

exp.	factor	# genes	rank	p-value	change direction
Swi4Δ	Swi4	175	1	$< 10^{-4}$	-
Swi4Δ	Sok2	65	64	0.197	+
Swi4Δ	Msn4	74	1	$< 10^{-4}$	+
Swi4Δ	Hap4	98	13	1.3×10^{-3}	+
Swi4Δ	Yap6	120	70	0.211	-
Sok2Δ	Sok2	65	40	0.1164	-
Sok2Δ	Msn4	74	94	0.429	+
Sok2Δ	Hap4	98	17	0.0276	+
Sok2Δ	Yap6	120	7	0.0107	+
Msn4Δ	Msn4	74	101	0.433	-
Hap4Δ	Hap4	98	1	$< 10^{-4}$	-
Yap6Δ	Yap6	120	43	0.134	+

in each deletion experiment. We chose genes bound by Msn4, Hap4 and Yap6 (p-value ≤ 0.001 from location data [100]) as the downstream genes in each pathway respectively. We also show the coherence p-values in the genes bound by each of the 106 transcription factors in Tables B.3-B.7 in the Appendix.

We first perform a "sanity check" to confirm the reliability of the expression coherence p-values. Genes bound by Swi4 and Hap4 exhibit very significant down regulations (p-values $< 10^{-4}$, rank top among all transcription factors) in Swi4Δ and Hap4Δ respectively. This observation is consistent with the knowledge about Swi4 and Hap4 as transcription activators ([39, 109]). In contrast, genes bound by Sok2, Msn4 or Yap6 did not exhibit strong coherence (see Table 5.6). The lack of coherence may be due to the dual functions of transcription factors (e.g., Sok2, [138]), the existence of parallel pathways (e.g., Msn4, [61]), or the false positives in the binding data.

We then investigate the coherence scores of downstream genes in each pathway. For genes bound by the downstream transcription factor of each pathway (Msn4, Hap4, Yap6), both Msn4 and Hap4 downstream genes were strongly up regulate in Swi4Δ (Msn4-downstream genes ranks first and had p-value $< 10^{-4}$, Hap4-downstream genes ranks 13th and had p-value 0.0013). Hap4-downstream genes also exhibit moderate up regulations in Sok2Δ (p-value 0.0276, ranks 16th) and strong

Table 5.7: Genes putatively regulated by Msn4

Gene	Function	Gene	Function
Gdh3	glutamate dehydrogenase	Ssa3	chaperone
Hsp26	heat shock protein	Tkl2	transketolase
Tps1	trehalose-6-phosphate synthase	Ara1	dehydrogenase
Glk1	glucokinase	Hsp30	heat shock protein
YDL124W	NADH-dep. reductase	Hsp42	heat shock protein
Hsp78	heat shock protein	Ttr1	glutaredoxin
Hor2	glycerol phosphate phosphatase	Ssa4	chaperone
Hsp12	heat shock protein	Mdj1	chaperone
Gsy1	glycogen synthetase	Hxk1	hexokinase
Ctt1	catalase T	Trx2	Thioredoxin
Sol4	6-phosphogluconolactonase	Sod2	manganese superoxide dismutase
Dog2	2-Deoxyglucose-6-phosphate phosphatase	Gre3	aldo/keto reductase
Sps100	spore wall formation	Dot5	nuclear thiol peroxidase
Lap4	aminopeptidase	YKR011C	unknown
Hsp104	heat shock protein	Ahp1	alkyl hydroperoxide reductase
Glo1	glyoxalase	YML131W	NAD-dependent oxidoreductase
Pgm2	phosphoglucomutase	Ald3	aldehyde dehydrogenase
Ddr48	stress protein	YMR315W	oxidoreductase
Ras2	GTP-binding protein	YNL134C	dehydrogenase
YNL194C	sphingolipid metabolism	YNL200C	stress protein
YOL150C	unknown	Gre2	stress protein

Table 5.8: Genes putatively regulated by Hap4

Gene	Function	Gene	Function
Pet9	ADP/ATP carrier protein	Cor1	ubiquinol cytochrome c reductase
Atp1	ATP synthase	Atp3	ATP synthase
Atp16	ATP synthase	Cox9	cytochrome c oxidase
Atp5	ATP synthase	Atp17	ATP synthase
Qcr7	ubiquinol cytochrome c reductase	Rip1	ubiquinol cytochrome c
Qcr6	ubiquinol cytochrome c reductase	cox4	cytochrome c oxidase
Cox13	cytochrome c oxidase	Qcr9	ubiquinol cytochrome c reductase
Qcr10	ubiquinol cytochrome c reductase	Cox6	cytochrome c oxidase
Atp2	ATP synthase	Atp7	ATP synthase
Cox12	cytochrome c oxidase	Cox5A	cytochrome c oxidase
Por1	mitochondria membrane porin	Cyt1	cytochrome c1
Tuf1	mitochondria translation elongation	Atp15	ATP synthase
Atp20	ATP synthase	Qcr2	ubiquinol cytochrome c reductase

down regulations in Hap4Δ (p-value $< 10^{-4}$, ranks first). Msn4-downstream genes had weak coherence scores in both Sok2Δ (p-value 0.492, ranks 94th) and Msn4Δ (p-value 0.433, ranks 101th). Yap6-downstream genes had weak coherence scores in Swi4Δ (p-value 0.211, ranks 70th) and Yap6Δ (p-value 0.1346, ranks 43th), and moderate up regulations in Sok2Δ (p-value 0.0107, ranks 7th). From these observations, only the Swi4 $\rightarrow$ Sok2 $\rightarrow$ Hap4 pathway was validated; the other two pathways were falsified. We can also infer the sign of each edge by inspecting the aggregate changes in the three deletion experiments: (Swi4,Sok2) is positive, (Sok2,Hap4) is negative, and Hap4 activates (in aggregate sense) downstream genes.

Table 5.9: Expression coherence on genes putatively regulated by factors in Sok2 subnetwork

exp.	factor	# genes	rank	p-value	change direction
Swi4Δ	Swi4	175	1	$< 10^{-4}$	-
Swi4Δ	Sok2	65	76	0.25	+
Swi4Δ	Msn4	43	10	5×10^{-4}	+
Swi4Δ	Hap4	36	37	0.044	-
Swi4Δ	Yap6	120	61	0.154	-
Sok2Δ	Sok2	65	25	0.0578	-
Sok2Δ	Msn4	43	1	$< 10^{-4}$	+
Sok2Δ	Hap4	29	10	0.0107	+
Sok2Δ	Yap6	120	5	0.0015	+
Msn4Δ	Msn4	43	1	$< 10^{-4}$	-
Hap4Δ	Hap4	29	1	$< 10^{-4}$	-
Yap6Δ	Yap6	120	43	0.114	+

The protein-DNA binding data from CHIP-chip experiments are noisy, and some protein-DNA interactions may not play functional roles. Therefore, including all genes bound by the same transcription factor will substantially degrade the coherence score. To overcome this problem, we pulled out sets of genes which were putatively regulated by Msn4 or Hap4 according to the yeast proteome database (YPD). Tables 5.7 and 5.8 list these genes. Msn4-regulated genes are stress response genes which are down regulated in Msn2ΔMsn4Δ double deletion mutants under normal or stress conditions ([17]). These genes do not necessarily exhibit significant changes in Msn4Δ due to the complementary function of Msn2 as discussed. Hap4-regulated genes are bound by Hap4 and experience significant down regulations in Hap4Δ. Most genes are related to respiration, such as ATP synthase or ubiquinol cytochromone c oxidoreductase complex. This is consistent with the function of Hap2-Hap3-Hap4-Hap5 complex of regulating respiration genes ([109]).

In contrast to the previous sets of downstream genes, both Msn4 and Hap4 regulated genes exhibit strong or moderate coherence scores in each deletion experiment. Msn4-regulated genes are up regulated in Swi4Δ (p-value 5×10^{-4}, ranks 10th), up regulated in Sok2Δ (p-value $< 10^{-4}$, ranks first), and down regulated in Msn4Δ (p-value $< 10^{-4}$, ranks first). Hap4-regulated genes are down regulated in Swi4Δ

(p-value 0.044, ranks 37th), up regulated in Sok2Δ (p-value 0.0107, ranks 10th), and down regulated in Hap4Δ (p-value $< 10^{-4}$, ranks first). The effects of deleting genes along Swi4 → Sok2 → Msn4 is particularly strong, which suggests this pathway may indeed relay the gene regulatory effects of perturbations to downstream genes. We survey the literature and find Sok2 does repress the function of Msn4 ([141]); Sok2 is the terminal gene of PKA signal transduction pathway, and Msn4 is repressed by enabling the PKA pathway. We have direct evidence (Msn4 is up regulated in Sok2Δ) and indirect evidence (Msn4-controlled genes are up regulated in Sok2Δ) to support the inhibitory function of Sok2. Since Hap4 also has the same response in Sok2Δ, we suspect Sok2 imposes the same effect on Hap4. However, Hap4-regulated genes responded weakly in the Sok2Δ experiment. One possible explanation is that each sub-unit of the Hap complex needs to increase in order to up regulate the relevant genes. Since Sok2Δ only brings up Hap4 but not Hap2, Hap3 or Hap5, it does not suffice to increase the levels of respiration related genes. The function of Swi4 on Sok2 is reported in some works but not yet conclusive ([165, 5]). Also, Swi4 does not belong to the PKA pathway. Thus the first edge of the Swi4 → Sok2 → Msn4 pathway needs to be further verified.

Figures 5-6 and 5-7 visualize the log ratios of expression change of Msn4 and Hap4-regulated genes in the new deletion experiments, and Table 5.9 enlists the expression coherence scores on restricted genes. Msn4-regulated genes have moderate propensity for down regulation due to the redundant function of Msn2. In contrast, they exhibit strong propensity for up regulation in both Swi4Δ and Sok2Δ. Hap4-regulated genes have very strong propensity for down regulation, which supports the prior knowledge about Hap4 as an essential component for the regulatory complex Hap2p-Hap3p-Hap4p-Hap5p. They have moderate propensity for down regulation in Swi4Δ and weak propensity of up regulations in Sok2Δ.

Since the prioritization of experiments is based on the expected information from current predictions, new data from the suggested experiments are not guaranteed to reduce model uncertainty. The responses of genes from the new data, however, demonstrate that they reduce the uncertainty about edge signs along Msn4 and Hap4

Figure 5-6: Responses along Msn4 pathway

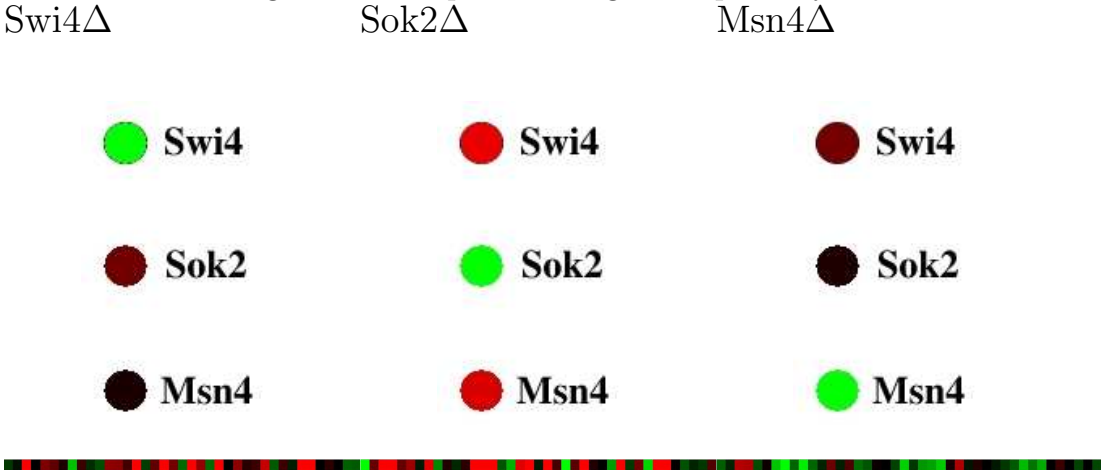
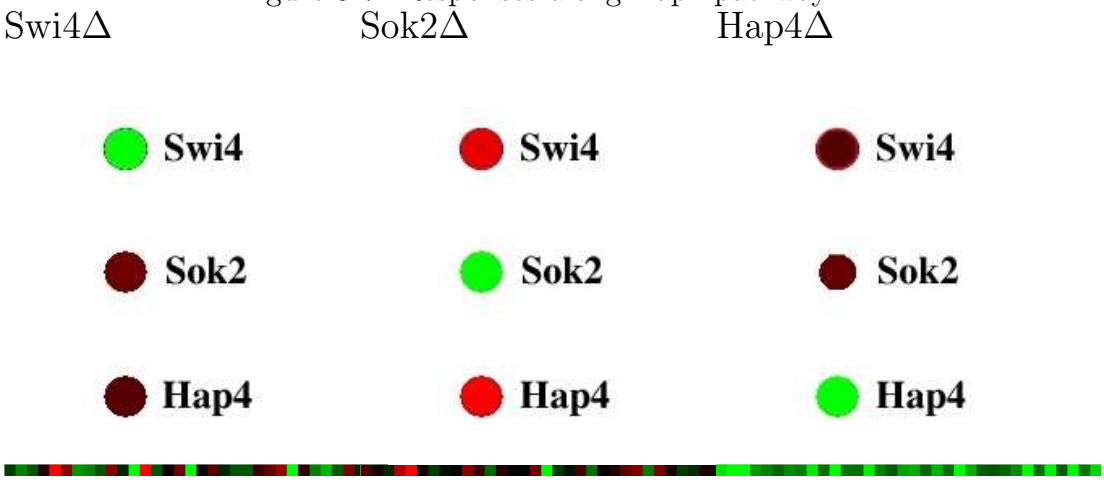
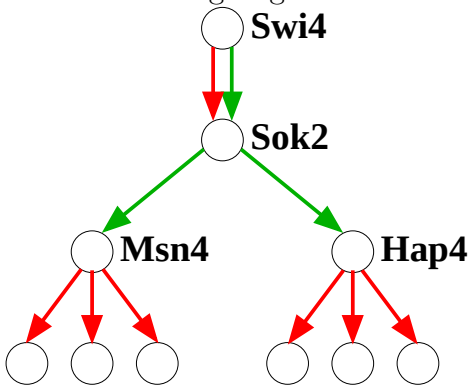


Figure 5-7: Responses along Hap4 pathway



pathways. We can now uniquely infer the edge signs along the pathways from the aggregate responses of their regulated genes. Figure 5-8 shows the inferred edge signs along the two pathways. (Swi4,Sok2) is positive along the pathway Swi4 → Sok2 → Msn4 and negative along Swi4 → Sok2 → Hap4. Sok2 inhibits both Msn4 and Hap4, and both Msn4 and Hap4 are activators. The contradictory edge sign on (Swi4,Sok2) may be due to many possible causes. One of the pathways may be invalid, Swi4 may not regulate Sok2, Swi4 may affect downstream genes via another hidden pathway, the function of Swi4 on Sok2 may depend on the pathway, the expression data may be inaccurate, and so on. More elaborate experiments are needed in order to test

Figure 5-8: Inferred edge signs of Sok2 subnetwork



these hypothesis.

The uncertainty about edge signs in the Sok2 subnetwork is reduced by incorporating new deletion data. However, the way to infer the edge signs from new data in previous discussion is by visually inspecting the gene expression changes in the deletion experiments. This "common sense reasoning" is not in line with the model inference algorithm introduced in Chapter Three, albeit their inferred values may coincide. In order to make the methodology consistent, we incorporated the new data in the physical network model with the same way as incorporating Rosetta data: breaking knock-out data into pairwise interactions and constructing potential functions to explain knock-out effects. We then applied the inference algorithm on the augmented model and identified the optimal configurations in the Sok2 subnetwork. The inferred edge signs along the two pathways were consistent with Figure 5-8 except the sign of the (Swi4,Sok2) edge was positive according to inference results. Notice we relaxed the condition of a valid pathway by not requiring that the deletion of all intermediate genes significantly alters the downstream gene (conditions in Section 3.3.3). This is because there are very few pathways satisfying this criterion as mentioned previously. We did not either impose the constraint that a transcription factor has a coherent function throughout all regulated genes (thus all protein-DNA edges emanating from the same factor have identical edge signs). Therefore, the edges emanating from Msn4 or Hap4 can have different signs. Without this constraint, the edge signs along the pathways are still uniquely determined. This is because the overall change of

the downstream genes tends to a specific direction in each experiment. Hence the evidence pertaining to the knock-out effects in one direction dominates the other direction. For example, if 9 of 10 Hap4-downstream genes are down regulated in Hap4Δ and 8 genes are up regulated in Sok2Δ, then setting the sign of (Sok2,Hap4) to negative yields much greater joint likelihood score than setting it to positive.

To sum up, our analysis on the new data validates certain pathways and falsifies the others. It also reduces the uncertainty of model configurations by uniquely determining edge signs along pathways. On the other hand, the new analysis results also indicate the contradiction on edge sign (Swi4,Sok2) and introduces new uncertainty in the model. Another scientific inquiry process would be needed in order to clarify this new uncertainty.

Chapter 6

Inferring Combinatorial Functions of Multiple Transcription Factors

It is evident that many genes are controlled by multiple transcription regulators. Both chromatin IP data and sequence data show many genes are bound by multiple proteins on their promoters. The RNA polymerase II holoenzyme, for example, consists of multiple subcomplexes and each subcomplex contains multiple proteins ([77]). Furthermore, the number of transcription factor genes is far less than the total number of genes in a genome, but most genes can be activated or repressed under multiple conditions. A plausible mechanism for a small number of transcription factors to regulate a large number of genes under a variety of responses is through the combinatorial effects of these factors. Empirically combinatorial control mechanisms have been identified in gene regulation circuitry. For instance, some transcription factors need to form a complex in order to bind to DNA promoters ([4]), homologous transcription factors complement each other function when one factor is missing ([61]), a repressor blocks the binding of an activator ([13]), and many others.

While fragmented instances of combinatorial control are discovered, biologists do not yet have a systematic understanding about the combinatorial control on genomic scale. This poses a challenging problem in both experimental and computational biology. The difficulty resides on the complexity of the underlying mechanisms, the lack of experimental technologies to reveal these mechanisms, and the insufficient

data points to even determine the functional relations of combinatorial control.

In spite of its difficulty, computational biologists have started inferring the combinatorial functions of multiple transcription factors from large scale datasets and achieved certain progress. Rather than emphasizing the complexity of the problem, the crux of the current progress relies on simplifying the problem and extracting information from limited data. For example, most of the computational studies focus on the regulatory mechanisms through modulating the abundance of transcription factors ([136, 12, 149, 150]). Other possible mechanisms such as protein modification and localization are ignored due to the lack of data.

The physical network model described in Chapter Three does not consider the combinatorial effect of multiple transcription factors. It assumes the perturbation of any regulator suffices to induce changes in the regulated gene. In this chapter, we extend this simplified assumption and consider the combinatorial effect of multiple transcription factors. We characterize the properties of single transcription factors in the context of combinatorial control. These properties are decomposed into two aspects: the functions of single factors as activators or repressors and the directions of effectiveness such as necessary or sufficient regulators. Based on this characterization, we can construct regulatory models and evaluate how well do these models fit the binding and expression data. An incremental algorithm is proposed to identify the regulatory models which best fit the data. Finally we apply this algorithm to large-scale datasets and analyze the experimental results.

6.1 Problem statement and hypotheses

There are different levels of questions regarding transcription regulation. Problems at *structural level* pertain to re-constructing the identities of members (who regulates who) in transcription regulation and detecting the "signatures" (motifs) on DNA promoters which can help to recognize these identities. CHIP-chip assays and motif analysis, for example, are meant to answer the structural questions. Problems at *functional level* pertain to characterizing the relations between the activities – such

as mRNA levels, protein levels, DNA or protein modification states – of transcription factors and regulated genes. For example, we may characterize a transcription factor as an activator or a repressor; or more specifically, whether its activating or inhibitory function relies on the presence of another protein. Problems at *mechanistic level* pertain to understanding the biophysical/biochemical mechanisms underlying transcription regulation. For example, whether a transcription factor affects transcription initiation by increasing its protein abundance, importing from cytosol to nucleus, chemically modifying the chromosome, blocking the access of other regulators to DNA promoters, or the combination of these mechanisms.

The goal of the work in this chapter is to identify the genes regulated by a set of transcription factors and to re-construct the functional relations between the mRNA levels of regulators and of regulated genes. Our work tackles problems at structural and functional levels. We do not intend to address problems at mechanistic level except simply assuming the primary mechanism is by modulating the protein (and mRNA) abundance of regulators. The reasons for avoiding the structural problems are due to the complexity of the system and the lack of data to reveal the underlying mechanisms. Nevertheless, fast-growing technologies and assays will soon provide abundant information about diverse gene regulation mechanisms at a large scale.

We apply the following simplified hypotheses in building the gene regulatory model of multiple transcription factors.

First, we assume transcription factors are the only immediate causes of transcription regulation and only concern about the effects of transcription factors which directly bind to DNA promoters. This assumption is consistent with the general picture about transcription regulation (Section 1.2) but does not cover certain exceptions. Other "direct causes" such as chromatin modification proteins may come into play. Moreover, ignoring the effects from indirect causes such as protein kinases may not be able to accurately characterize gene regulation at function level.

Second, given that a transcription factor binds to a specific promoter, we postulate the activity of the factor on the target promoter is modulated by its protein abundance. Furthermore, we assume the mRNA levels captured by DNA microarrays

faithfully reflect the abundance of their corresponding proteins. These assumptions combined allow us to use mRNA levels to indicate the activities of genes, hence directly model the dependencies of mRNA data as many previous works did. However, both assumptions may be too simple. Modulation of protein activities can be achieved by post-translational modifications or localization which are not captured by protein abundance. The binding profile acquired under one condition may be very different from the profile acquired under the other condition ([12]). Moreover, previous studies showed mRNA and protein levels of the same gene were poorly correlated ([83]). Therefore, inferred results according to these simplified assumptions – including most previous works of gene expression analysis and our work in this chapter – need to be carefully scrutinized.

Third, very often a group of genes are co-regulated by a set of transcription factors with the same function. This assumption implies *gene modules* – rather than individual genes – are the basic units of gene regulation. The module hypothesis is the foundation of many previous works including clustering gene expression, and the regulatory module works in [12] and [136]. In our work, the module assumption helps reducing over-fitting since the combinatorial functions are built from multiple genes rather than single genes.

Fourth, following the assumption of physical network models, we quantize the changes of mRNA levels with respect to a reference condition into three states: up regulation, down regulation, no change. The state "no change" can be unfolded into two possible scenarios: the state of "actually no change", meaning that the mRNAs in the majority of the cells do not change, and a uniform mixture of up and down regulations over the population. We do not distinguish between these two scenarios because they are indistinguishable at population level in the mRNA data we use.

Fifth, we hypothesize that each transcription factor has a distinct function (activator or repressor) on all genes it regulates, and its function is not inverted in the context of combinatorial control. Therefore, an activator will not become a repressor when it collaborates with other factors to regulate other genes. Nevertheless, its function can be disabled in the context of combinatorial control. For example, Hap4

forms a complex with Hap2, Hap3 and Hap5 to activate genes involved in respiration (see Section 5.4). The activating function of Hap4 is disabled when any other member of the complex is absent. This is certainly a strong assumption and unlikely to hold for most transcription factors. However, many transcription factors do have a primary function (as an activator or a repressor) on most genes they control. To a crude approximation we can thereby assume they have a consistent function.

Sixth, although a transcription factor may coordinate with other factors to regulate genes, sometimes the effect of altering a single factor can be exhibited regardless of the states of other factors. For example, if a factor has to form a complex with several other proteins in order to activate a gene, then the deletion of this factor will down-regulate the affected gene regardless of the presence or absence of other factors. We categorize the combinatorial property of a transcription factor in terms of the direction of its activity changes that alters the regulated genes. A transcription factor is a necessary regulator if its down regulation disables its function. Conversely, a transcription factor is a sufficient regulator if its up regulation enhances its function. A regulator can also take effect in both directions or neither direction. We construct the combinatorial functions in terms of the functions and combinatorial properties of single factors. Details about this construction will be elaborated in the next section.

Seventh, we encode the uncertainty/stochasticity of functional relations by assigning probabilistic outputs to combinatorial functions. However, unlike fully parameterized Bayesian networks which integrate over all possible probabilistic functions, the probabilistic functions in our model are derived from a small number of deterministic combinatorial functions. For instance, if the deterministic function is the identity operation of a single input, then its probabilistic function outputs $+1$ with a high probability and 0 with a low probability when the input is $+1$.

Eighth, we view the actual protein-DNA bindings and expression states as discrete hidden variables. They are measured via noisy experimental processes (CHIP-chip or microarrays). The relation between hidden and observed variables can be specified as a noisy sensor model and derived from the error model of the measurements.

6.2 Elements of a regulatory model

Evidence from various experiments suggest genes are regulated as groups rather than individuals. Identifying the co-regulated gene groups – which are often termed as modules – becomes an important topic in current computational biology. Several works have attacked this problem and achieved fruitful results (see Section 1.4). Different from these works, we emphasize on the interpretability pertaining to the mechanisms. We also distinguish our works from some other studies by focusing on the effects of transcription factors on regulated genes.

We view a regulatory model as a combination of three elements: a set of transcription factors, a set of genes regulated by these regulators, and the regulatory program of the model – the combinatorial function specifying the relation between the activities of regulators to the mRNA levels of regulated genes.

6.2.1 Regulators and regulated genes

The meanings of regulators and regulated genes are self evident. To establish regulator and regulated gene sets of a model, we require empirical evidence indicating their relations. An obvious choice is to include bicliques in the protein-DNA interaction network: all transcription factors (proteins) bind to all genes (promoters) in each biclique. False positives in this set are expected since physical bindings per se may not have functional roles. False positives in the initial establishment can be reduced by incorporating gene expression data. For example, one gene is excluded if its expression profile is significantly different from other members in the module. On the other hand, false negatives are not easy to be re-incorporated due to the lack of evidence of physical interactions. For instance, if only the binding data under normal condition are provided, one gene is bound by a regulator member under a perturbation condition but not under normal condition, it will not be considered as a member of the module. There are multiple pairs of regulators and regulated genes sets inferred from the data. They are not required to be disjoint.

6.2.2 Regulatory programs

A regulatory program specifies the relation between the activities of regulators and the mRNA levels of regulated genes. As discussed in Section 6.1, we choose the mRNA levels of transcription factors as the proxies to their activities due to the lack of other types of data. Therefore, the regulatory program in this work are the functions of gene expression level changes (with respect to the normal condition).

A straightforward way of generating regulatory programs is to consider all tri-state (up or down regulation, no change) Boolean functions. In spite of its expressiveness, this formulation has several shortcomings. The number of possible Boolean functions is super-exponential to the number of inputs (3^{3^n}, n is the number of inputs). The large size of combinatorial function space implies a serious over-fitting problem. To uniquely determine a function, all 3^n possible input configurations need to appear in the dataset. This is unlikely to occur in real data. Furthermore, the representation of Boolean functions is difficult to interpret in terms of the underlying mechanisms. We often need to decompose the function into smaller elements and understand each component in terms of the mechanisms. Finally, the deterministic property of Boolean function is incapable of representing the stochasticity of gene regulation and its measurements.

These problems inspire us to modify the tri-state Boolean functions in the sense of reducing the complexity of the model class and introducing probabilistic components. A key step of simplifying the regulatory programs is to investigate the properties of single transcription factors in the context of combinatorial control. In this work, we annotate each transcription factor with two properties. First, a single transcription factor possesses a consistent function throughout all its regulated genes. In other words, a factor is either an activator or a repressor for all its regulated genes. Second, in each regulatory model, a transcription factor can be categorized according to the direction of effectiveness on its affected genes. A regulator is necessary if decreasing its expression level leads to the responses inverted from the function of this regulator. One mechanistic example is a protein complex as an activator. Each subunit protein

of the complex is a necessary activator for the absence of each member will decrease the levels of its regulated genes. Conversely, the increase of a necessary regulator may not affect regulated genes. A regulator is sufficient if increasing its expression level leads to the responses along the direction of its function. A mechanistic example is two transcription factors independently activate regulated genes. The increase of any regulator suffices to up regulate the regulated genes. Conversely, the decrease of a sufficient regulator may not affect regulated genes. The categorization of necessary and sufficient regulators is neither mutually exclusive nor exhaustive. A regulator can be both necessary and sufficient if its change in each direction affects regulated genes. A regulator can be neither necessary nor sufficient if its expression level change does not affect regulated genes. This scenario occurs when the activity of a regulator is dictated by unobserved properties such as protein abundance or protein modifications. It is also possible that this regulator does not have functional roles in this model. Table 6.1 enlists the responses of regulated genes under each combination of single factor function and the direction of effectiveness.

Notice our definition of necessary or sufficient regulators does not entirely follow the convention of the same terms in biology. In biology, the necessary property of a regulator is often determined by comparing the response of deleting this regulator versus the control experiment. Similarly, the sufficient property is determined by checking whether the presence of the regulator on a promoter suffices to activate/inhibit genes. We do not use these terms to characterize the biological functions of regulators but only to delineate their directions of effectiveness in terms of gene expression data. Consequently, a regulator which is labeled as neither necessary nor sufficient does not suggest it plays no functional roles in a module. Instead, it suggests its effect on gene expression may not be revealed by its mRNA levels. Furthermore, since we are interested in the directions of mRNA level changes relative to normal conditions, the necessary and sufficient labels depend not only on the intrinsic properties of regulators but also the expression states of genes under the normal condition. For example, if the expression level of a gene is low under the normal condition, then a regulator cannot be necessary because the gene cannot further decrease. We are aware of these

Table 6.1: Responses of regulated genes in each combinatorial category

	necessary	sufficient	both	neither
activator	$f \downarrow \Rightarrow g \downarrow$	$f \uparrow \Rightarrow g \uparrow$	$f \downarrow \Rightarrow g \downarrow$, $f \uparrow \Rightarrow g \uparrow$	g any value
repressor	$f \downarrow \Rightarrow g \uparrow$	$f \uparrow \Rightarrow g \downarrow$	$f \downarrow \Rightarrow g \uparrow$, $f \uparrow \Rightarrow g \downarrow$	g any value

shortcomings in our categorization and will discuss possible improvement methods in Chapter Seven.

Table 6.1 characterizes the predicted response of regulated genes by altering single transcription factors. We want to build combinatorial functions of multiple transcription factors based on the properties of single regulators. To do that we have to specify the rules of combining the predictions from single factors. The goal is to construct a mapping from each configuration of multiple regulators to the "typical response" of regulated genes. Denote $S = \{-1, 0, +1\}$ as the state of gene expression changes. The combination rules are as follows.

1. Let the function of a regulator r be f_r ($+1$ for activator and -1 for repressor). If the direction of effectiveness of r is necessary and the input configuration on r is -1, then the output influenced by r is $x_g(r) = -f_r$.

2. If the direction of effectiveness of r is sufficient and the input configuration on r is $+1$, then the output influenced by r is $x_g(r) = f_r$.

3. For other combinations of directions of effectiveness and input configuration, the output influenced by r is $x_g(r) = 0$.

4. For each input configuration, if the output influenced by each factor is either $+1$ or 0 (but not all 0s), then the output is $+1$. If the output influenced by each factor is either -1 or 0 (but not all 0s), then the output is -1. Otherwise the output is 0.

To summarize these rules, the predicted responses of regulated genes are the consensus of the predictions according to the changes of individual regulators. Predictions of no change according to a regulator can be overwritten by predictions of significant

changes according to other regulators. In addition, when predictions from different regulators contradict (some predictions are +1 and some are -1), the output is 0. This means that we are unable to predict the response, or that the output is a mixture of positive and negative responses with an unknown proportion. Notice again that we do not distinguish between the actually no change state and the mixture state, as stated in the hypotheses.

The predictions from single regulators in Table 6.1 and the combination rules stated above generate a unique combinatorial function given the function and the direction of effectiveness of each transcription factor. This setting allows us to generate combinatorial functions from the properties of single regulators. It is certainly a simplification, and the resulting functions generated by the setting do not cover all possible tri-state Boolean functions. Some regulators are known to be activators for some genes under some conditions and repressors for other genes under other conditions (for example, Sok2, [141]). One can also easily picture a scenario that the "necessary" or "sufficient" label of a factor depends on certain input configurations of regulators. For example, two independent activator complexes are formed by factors f_1, f_2 and f_3, f_4 respectively. f_1 is a necessary activator only when the complex $f_3 f_4$ is not functioning. In addition, regulator groups f_1, f_2 and f_3, f_4 form sufficient regulators but individual factors do not. Despite these limitations, this categorization greatly reduces the complexity of combinatorial functions and provides a clear interpretation from mechanistic perspective. With fixed single factor functions (we can either infer single factor functions from data or find them from literature survey), each factor can choose one of the four combinatorial labels (necessary, sufficient, neither, both) independently. Hence the number of possible combinatorial functions in this restricted class is exponential in terms of the input size ($4^n - 1$, where n is the number of inputs; we exclude the scenario when all factors in a model are neither necessary nor sufficient for they denote regulated gene expression changes are independent of regulator gene expression changes). Although the number of possible functions is still large for large n, empirically we are able to exhaust all functions of small input sizes (e.g., 2 to 3 regulators) and find the one(s) which optimally fit the data. The models

Table 6.2: Conversion from deterministic to probabilistic outputs

x_g	y_g
0	$P(y_g = -1) = P(y_g = 0) = P(y_g = 1) = \frac{1}{3}$
+1	$P(y_g = 1) = 1 - \epsilon, P(y_g = 0) = \epsilon$
-1	$P(y_g = -1) = 1 - \epsilon, P(y_g = 0) = \epsilon$

with many regulators are biologically less interesting, for there are very few genes bound by many regulators.

An example of a two-factor function with both factors as necessary activators is shown in the first three columns in Table 6.3.

The outputs of the combinatorial functions described above are deterministic. We want to make these outputs to cope with noise and uncertainties of experiments. The most general approach is to replace the truth table of a Boolean function with a probability table specifying the probability of each output state under each input configuration (i.e., a Bayesian network). However, learning Bayesian networks with specific functions is more involved. Learning the structure of a Bayesian network is less interesting in this context for it evaluates the marginal likelihood by averaging over all possible probabilistic functions (the marginal likelihood function equation 1.5 in Chapter One). To learn the specific functions in a Bayesian network we have to partition the parameter (probability table) space according to the combinatorial functions stated above and integrate over the restricted parameter space. In this work, we adopt a simpler approach by assigning each deterministic output (the typical response predicted from multiple regulators) to a fixed probability distribution over expression states. For example, if the typical response is +1, then the probability of the "real response" to be +1 is $\frac{2}{3}$, and the probability of no change is $\frac{1}{3}$. The mapping from deterministic to probabilistic outputs is shown in Table 6.2.

ϵ can be interpreted as the expected fraction of experiments among the dataset which are consistent with the combinatorial function. This construction is much less flexible and does not require learning the probability values from data. It reduces the burden and overfitting of learning from data but also raises the concern about the accuracy of learned models. The learned models can be sensitive to the choice of

Table 6.3: A combinatorial function, both regulators are necessary activators

x_{r_1}	x_{r_2}	x_g	y_g
-1	-1	-1	$P(y_g=-1)=\frac{2}{3}, P(y_g=0)=\frac{1}{3}$
-1	0	-1	$P(y_g=-1)=\frac{2}{3}, P(y_g=0)=\frac{1}{3}$
-1	1	-1	$P(y_g=-1)=\frac{2}{3}, P(y_g=0)=\frac{1}{3}$
0	-1	-1	$P(y_g=-1)=\frac{2}{3}, P(y_g=0)=\frac{1}{3}$
0	0	0	$P(y_g=-1)=P(y_g=0)=P(y_g=1)=\frac{1}{3}$
0	1	0	$P(y_g=-1)=P(y_g=0)=P(y_g=1)=\frac{1}{3}$
1	-1	-1	$P(y_g=-1)=\frac{2}{3}, P(y_g=0)=\frac{1}{3}$
1	0	0	$P(y_g=-1)=P(y_g=0)=P(y_g=1)=\frac{1}{3}$
1	1	0	$P(y_g=-1)=P(y_g=0)=P(y_g=1)=\frac{1}{3}$

the free parameter ϵ. We will show in the empirical results that this is not the case. The mapping of no change prediction assigns equal probability to each state. This implies we are unable to predict no change of regulated genes. This seems to be a shortcoming of the model but has a desirable feature for practical purposes. In the expression data used, most genes are not significantly changed in most experiments. Allowing the model to predict no change thus would strongly bias toward the function which outputs no change regardless of input states. Our choice of probability values does not have this problem because it emphasizes on predicting significant changes of regulated genes.

Table 6.3 shows the probabilistic function derived from the deterministic function that both regulators are necessary activators.

6.3 Likelihood function of a regulatory model

The regulatory model described in previous sections is a generative model. We can thus formulate the joint likelihood function of binding and expression data according to the model. Once the joint likelihood function is defined, we adopt an incremental algorithm which identifies the regulatory models that yield the maximum likelihood. In this section we discuss the formulation of the likelihood function.

We use protein-DNA binding data under the normal condition and two-channel microarray gene expression data of perturbations versus normal conditions. Intu-

itively, a regulatory model fits a binding data if all regulators bind to all regulated genes. A regulatory model fits an expression data if the gene expression changes in each experiment conform with the combinatorial function described in the previous section. These criteria are transformed into the likelihood function of observed data. Since binding and gene expression data are independent sources, we construct the likelihoods of the two data separately and combine them by multiplication.

We first define the following notations pertaining to protein-DNA bindings. Denote $M = (R, G, f)$ as a regulatory model, where R and G are regulator and regulated gene sets and f the regulatory program. For each $r \in R$ and $g \in G$, define b_{rg} as the binary variable indicating whether r binds to g. b_{rg} is not directly observed but through a measurement outcome x_{rg}. In the CHIP-chip data, x_{rg} corresponds to the log ratio of DNA abundance between the promoters purified by immunoprecipitation and the background. We transform the conditional probabilities $P(x_{rg}|b_{rg})$ from the p-values of CHIP-chip data as stated in Section 3.3.1.

The values of all indicator variables $\{b_{rg}\}_{r\in R, g\in G}$ constitute configurations in the space $\{0,1\}^{|R|\times|G|}$. Since $\{b_{rg}\}$ are hidden, the likelihood function is the joint probability of measurements $\{x_{rg}\}$ marginalized over hidden states $\{b_{rg}\}$. Its evaluation requires specifying the prior probabilities $P(\{b_{rg}\})$. We are interested in two priors. First, the only configuration consistent with the regulatory model is that each regulator binds to the promoter of each gene. This prior concentrates the entire probability mass on a single state:

$$H_1 : P(\{b_{rg}\}) = \delta(b_{rg} = 1, \forall r \in R, g \in G). \tag{6.1}$$

where $\delta(.)$ is the indicator function. Second, as a comparison we build a uniform prior over all binding states:

$$H_0 : P(\{b_{rg}\}) = \frac{1}{2^{|R||G|}}. \tag{6.2}$$

The marginal likelihood of binding data is

$$P(\{x_{rg}\}) = \sum_{b_{rg}} P(\{b_{rg}\}, \{x_{rg}\}) = \prod_{(r,g)} \sum_{b_{rg}} P(b_{rg}) P(x_{rg}|b_{rg}). \tag{6.3}$$

The second equality arises from the independence of binding measurements. Substituting the priors of H_1 and H_0 into equation 6.3,

$$\begin{aligned} P(\{x_{rg}\}|H_1) &= \textstyle\prod_{(r,g)} P(x_{rg}|b_{rg}=1). \\ P(\{x_{rg}\}|H_0) &= \tfrac{1}{2^{|R||G|}} \textstyle\prod_{(r,g)} (P(x_{rg}|b_{rg}=1) + P(x_{rg}|b_{rg}=0)). \end{aligned} \tag{6.4}$$

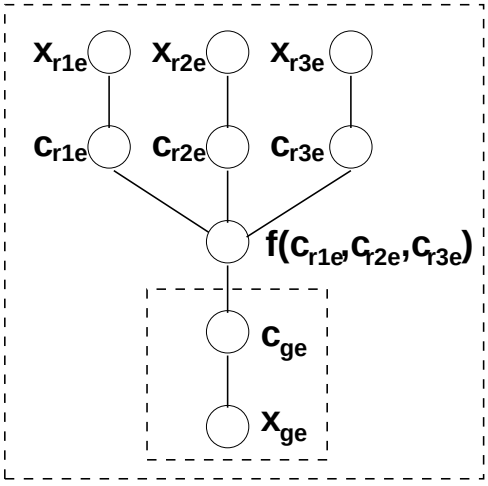
The log likelihood ratio of the model consistent with the module assumption versus the null model then becomes:

$$\begin{aligned} &L^b(R,G) = \log P(\{x_{rg}\}|H_1) - \log P(\{x_{rg}\}|H_0) = \\ &|R||G| \log 2 + \textstyle\sum_{(r,g)} [\log P(x_{rg}|b_{rg}=1) - \log(P(x_{rg}|b_{rg}=1) + P(x_{rg}|b_{rg}=0))]. \end{aligned} \tag{6.5}$$

The log likelihood ratio of gene expression data is analogously defined. Denote E as a set of experiments and $e \in E$ an experiment index, c_{re} as the actual gene expression change (with respect to the normal condition) of regulator r in experiment e, c_{ge} as the actual gene expression change of gene g in experiment e. Also denote x_{re} and x_{ge} as the measurements of c_{re} and c_{ge} respectively. Conditional probabilities $P(x_{re}|c_{re})$ and $P(x_{ge}|c_{ge})$ can be derived from p-values of gene expression measurements as described in Section 3.4. When measurement p-values are not provided, we evaluate the conditional probabilities according to Gaussian and exponential distributions. We will discuss this calculation in the Appendix.

The regulatory function described in Section 6.2 operates on the actual gene expression changes c_{re} and c_{ge}. Figure 6-1 illustrates a generative model of a regulatory function on expression data. The actual regulator expression changes $\{c_{re}\}$ are the input states of the function. The function f first maps an input state into a deterministic, intermediate state $f(\{c_{re}\})$. The value of $f(\{c_{re}\})$ belongs to $S = \{-1, 0, +1\}$ according to the rules depicted in Section 6.2. $f(\{c_{re}\})$ then generates the output

Figure 6-1: Generative model of expression data



state c_{ge} of each gene in each experiment according to the conversion rules in Table 6.2. We adopt the *plate* representation in Figure 6-1: nodes within the inner box (c_{ge}, x_{ge}) are instantiated by gene indices g and nodes within the outer box (all variables in the figure) are instantiated by experiment indices e. For notational convenience, define probability measures μ_-, μ_0, μ_+ as

$$\begin{aligned}
&\mu_-(y=-1) = 1-\epsilon, \mu_-(y=0) = \epsilon. \\
&\mu_0(y=-1) = \mu_0(y=0) = \mu_0(y=1) = \tfrac{1}{3}. \\
&\mu_+(y=+1) = 1-\epsilon, \mu_+(y=0) = \epsilon.
\end{aligned} \tag{6.6}$$

We apply the following functions to the conditional probability $P(c_{ge}|f(\{c_{re}\}))$:

$$\begin{aligned}
&P(c_{ge}|f(\{c_{re}\}) = +1) = \mu_+(c_{ge}) \equiv (1-\epsilon)\delta(c_{ge}=+1) + \epsilon\delta(c_{ge}=0). \\
&P(c_{ge}|f(\{c_{re}\}) = -1) = \mu_-(c_{ge}) \equiv (1-\epsilon)\delta(c_{ge}=-1) + \epsilon\delta(c_{ge}=0). \\
&P(c_{ge}|f(\{c_{re}\}) = 0) = \mu_0(c_{ge}) \equiv \tfrac{1}{3}\delta(c_{ge}=+1) + \tfrac{1}{3}\delta(c_{ge}=-1) + \tfrac{1}{3}\delta(c_{ge}=0).
\end{aligned} \tag{6.7}$$

μ_+ assigns a large probability to $+1$ and the remaining probability mass to 0 when the predicted value $f(\{c_{re}\}) = +1$. μ_- works analogously to -1. When $f(\{c_{re}\}) = 0$, we are uncertain about the actual expression changes of regulated genes, thereby assign an equal probability to each state.

We construct the log likelihood function of expression data similar to the binding data. The hidden variables are c_{re} and c_{ge} for each regulator $r \in R$, gene $g \in G$ and experiment $e \in E$. We are interested again in two priors of hidden states. H_1 assigns equal probability to each input hidden state $\{c_{re}\}$ and determines the probability of output hidden state $\{c_{ge}\}$ according to the regulatory function and input hidden state $\{c_{re}\}$ H_0 assigns a uniform probability to all possible hidden states.

$$\begin{aligned} H_1 &: P(\{c_{re}\}\{c_{ge}\}) = \tfrac{1}{3^{|E||R|}} P(\{c_{ge}\}|\{c_{re}\}, f). \\ H_0 &: P(\{c_{re}\}\{c_{ge}\}) = \tfrac{1}{3^{|E|(|R|+|G|)}}. \end{aligned} \tag{6.8}$$

where $P(\{c_{ge})\}|\{c_{re}\}, f) = \prod_e \prod_g P(c_{ge}|f(\{c_{re}\}))$. The marginal likelihood of gene expression data over hidden states is

$$P(\{x_{re}\}, \{x_{ge}\}|H) = \sum_{\{c_{re}\},\{c_{ge}\}} P(\{c_{re}\}\{c_{ge}\}|H) \prod_{e\in E} \prod_{r\in R} P(x_{re}|c_{re}) \prod_{g\in G} P(x_{ge}|c_{ge}). \tag{6.9}$$

Substituting equation 6.8 into equation 6.9, we obtain

$$\begin{aligned} &P(\{x_{re}\}, \{x_{ge}\}|H_0) = \tfrac{1}{3^{|E|(|R|+|G|)}} \textstyle\prod_{e\in E} \prod_{r\in R} (P(x_{re}|c_{re} = +1) + P(x_{re}|c_{re} = -1) + P(x_{re}|c_{re} = 0)) \cdot \\ &\textstyle\prod_{g\in G} (P(x_{ge}|c_{ge} = +1) + P(x_{ge}|c_{ge} = -1) + P(x_{ge}|c_{ge} = 0)). \end{aligned} \tag{6.10}$$

and

$$\begin{aligned} &P(\{x_{re}\}, \{x_{ge}\}|H_1) = \tfrac{1}{3^{|E||R|}} \textstyle\prod_{e\in E} [\sum_{\{c_{re}\}} \prod_{r\in R} P(x_{re}|c_{re}) \prod_{g\in G} \sum_{c_{ge}} P(c_{ge}|f(\{c_{re}\}))P(x_{ge}|c_{ge})] \\ &= \tfrac{1}{3^{|E||R|}} \textstyle\prod_{e\in E} [\sum_{v=\{-1,0,+1\}} P_v(e) \cdot \prod_{g\in G} \sum_{c_{ge}} P(c_{ge}|v)P(x_{ge}|c_{ge})]. \end{aligned} \tag{6.11}$$

where $P_v(e)$ denotes the probability of the regulator configurations in experiment e which generates deterministic output v:

$$P_v(e) = \sum_{\{c_{re}\}} \delta(f(\{c_{re}\}) = v) \cdot \prod_{r\in R} P(x_{re}|c_{re}). \tag{6.12}$$

The log likelihood ratio of H_1 versus H_0 is:

$$
\begin{aligned}
&L^e(R,G,f) = \log P(\{x_{re}\},\{x_{ge}\}|H_1) - \log P(\{x_{re}\},\{x_{ge}\}|H_0) \\
&= -|E||R|\log 3 + \textstyle\sum_{e\in E}[\log(\sum_{v=\{-1,0,+1\}} P_v(e)\cdot\prod_{g\in G}\sum_{c_{ge}} P(c_{ge}|v)P(x_{ge}|c_{ge}))] \\
&+|E|(|R|+|G|)\log 3 - \textstyle\sum_{e\in E}[\sum_{r\in R}\log(P(x_{re}|c_{re}=+1)+P(x_{re}|c_{re}=-1)+P(x_{re}|c_{re}=0)) \\
&+\textstyle\sum_{g\in G}\log(P(x_{ge}|c_{ge}=+1)+P(x_{ge}|c_{ge}=-1)+P(x_{ge}|c_{ge}=0))].
\end{aligned}
\tag{6.13}
$$

We define the joint log likelihood ratio as the weighted sum of the functions of binding and expression data:

$$L(R,G,f) = L^b(R,G) + \lambda L^e(R,G,f). \tag{6.14}$$

λ is a free parameter specifying the relative importance of expression data with respect to binding data. Since the number of expression experiments far exceeds the number of binding experiments, we have to degrade the importance of expression data in order to make binding data relevant. We set $\lambda = 0.1$ in the experiments but will also show modeling results are insensitive against λ values.

There are several distinctions between the joint likelihood ratio defined in equation 6.14 and a deterministic score of fitting a Boolean function to the data (for example, by counting the number of empirical instances which do not conform with the Boolean function). First, our formulation takes the uncertainty of measurements into account. Measurement uncertainty is encoded in the conditional probabilities $P(x_{rg}|b_{rg}), P(x_{re}|c_{re})$ and $P(x_{ge}|c_{ge})$ and marginalized over all consistent states of hidden variables. Second, equation 6.14 jointly considers the strength of binding and the coherence of gene expression. A gene can be included in a model if its binding confidence is low but its expression profile conforms with the regulator expression data and the combinatorial function. Conversely, a gene can be included if its expression data are not very consistent with the combinatorial function but the binding score is strong. Third, we do not require the expression data of all regulated genes conform with the predictions of the combinatorial function on all experiments. When the prediction from a deterministic function is +1, then either up regulations or no

changes yield reasonable scores. This is because we assign a non-negligible probability to 0 when the deterministic output is +1 (equation 6.9). Similar arguments hold when the deterministic prediction is -1. When the deterministic prediction is 0, then either most regulated genes do not change or regulated genes with mixed responses yield high likelihood scores. This formulation is much less stringent than the requirement that all regulated genes must conform with the combinatorial function on all experiments. For example, if the first half of the regulated genes are down regulated and the second half of the regulated genes do not change in experiment 1, and the first half of the regulated genes do not change and the second half of the regulated genes are down regulated in experiment 2, then this model will yield high likelihood score even though the regulated genes are not highly correlated.

6.4 Identifying regulatory models from data

Equation 6.14 evaluates the likelihood score of a given regulatory model. We are interested in identifying the regulatory models which maximize the likelihood ratio. We adopt a greedy algorithm to incrementally add members to the regulatory model. The algorithm first identifies regulator groups which co-bind to a considerable number of genes. For each regulator group, it incrementally adds genes which optimize the likelihood score. It stops when the improvement of adding a new gene is insignificant compared to randomly selecting genes. The algorithm then reports the models which yield significant likelihood scores.

6.4.1 Finding candidate regulator sets

A candidate regulator set contains regulators which co-bind to a number of genes according to the protein-DNA interaction data. In principle, any collection of regulators constitute a candidate set. In practice, we threshold on the binding p-values ($p \leq 0.005$) and only consider the regulator sets whose binding p-values to their targets are below the threshold.

All significant protein-DNA interactions constitute a bi-partite graph which is a

subset of the physical network in Chapter Three. A candidate regulator set and their candidate gene set are the vertex sets of a biclique in the graph. A biclique in the physical network is a maximal complete subgraph containing edges connecting between all members in the regulator set and the regulated gene set. It is maximal in the sense that there exists no extra regulator which binds to each member in the regulated gene set and no extra regulated gene which is bound by each member in the regulator set. Finding the largest bicliques in a bi-partite graph is known to be NP-hard ([124]). Hence finding all bicliques is also NP-hard. We apply a heuristic similar to [12] to identify a subset of them. Each gene is bound by a subset of transcription factors (the subset can be a null set). We call this subset the binding pattern of this gene. The number of binding patterns appeared in the data is upper bounded by the total number of genes since each gene possesses one binding pattern. It is straightforward to enumerate all the binding patterns and count the number of genes which possess each pattern. We then select the binding patterns which are possessed by a significant number of genes (≥ 10 genes). The selected binding patterns (subsets of regulators) and their corresponding genes form the candidate sets of regulators and regulated genes. Empirically, we find very few binding patterns contain more than 3 regulators and are possessed by more than 10 genes. To be able to generate all regulatory programs of each regulator set, we only consider the binding patterns with ≤ 3 regulators. Notice that each binding pattern is associated with a biclique, but not all regulator sets in bicliques are binding patterns. For example, intersections of binding patterns may also be associated with bicliques, though they may not be binding patterns.

6.4.2 Determining regulated genes and regulatory programs

Significant bindings do not suffice to generate a model. To demonstrate the regulation influence, gene expression data in the model should also be consistent with the regulatory program. In this section we will discuss an incremental algorithm of determining combinatorial functions and regulated genes.

Given a regulator set and a regulatory program, the incremental algorithm of in-

cluding regulated genes is straightforward: at each iteration, choose the gene from the candidate set which maximizes equation 6.14. The stopping criterion needs to be specified. Stop when the joint likelihood function starts decreasing is not viable because the likelihood function monotonically increases with the number of regulated genes (see the Appendix). Instead, we evaluate the p-value of the likelihood function generated from random data and stop when the p-value of adding a gene is insignificant ($p > 0.1$ in our analysis). We are able to analytically approximate the p-value without performing random sampling. The derivation of the p-value approximation is shown in the Appendix.

The incremental algorithm finds a regulated gene set for each regulatory program. We compare the likelihood functions of models corresponding to each regulatory program and identify the one with the maximum score. Because the the likelihood function increases with the size of the regulated gene set, we need to fix the size of regulated gene sets when comparing the scores of different regulatory programs. Each regulatory program has a regulated gene set with possibly different size. We fix the size of each regulated gene set to the size of the smallest gene set among all regulatory programs. Since genes in each regulated gene set are added with an decreasing order of importance (the first gene added to the set fits the combinatorial function the best), we restrict each gene set to top n genes, where n is the fixed size. We then compare the likelihood functions of all regulatory programs in the restricted gene sets and identify the ones which yield optimal or suboptimal scores.

6.4.3 Significance of a regulatory model

For each candidate regulator set, the incremental algorithm finds a combinatorial function and a regulated gene set. However, some of those regulator sets are likely to be spurious, which make the resulting regulatory models fit the data less well. We are interested only in those regulatory models which fit the binding and expression data. To filter out unfitted models, we evaluate the p-values of the likelihood score by comparing the empirical likelihood scores to the scores calculated from randomly permuted gene expression data. Only the models with significant p-values are kept

for further analysis. The computation of model p-values is discussed in the Appendix.

6.4.4 Merging multiple regulatory programs

The gene expression data on a regulatory model often cannot uniquely determine its combinatorial function because some input configurations do not appear in the data. Consequently, there are likely multiple regulatory programs which yield equal or approximately equal scores. A suboptimal function is also of interest since the log likelihood values can be affected by noise of data or specific values of free parameters. A single regulatory program in this context is not very meaningful because it may contain incomplete and misleading information. For example, if all optimal and suboptimal functions indicate activator r_1 is necessary, then r_1 is likely to be necessary. Conversely, if two optimal functions yield similar scores but one indicates r_1 is necessary and the other does not, then it is unclear whether r_1 is necessary.

We want to find a representation of inferred regulatory programs which can faithfully communicate the information about all optimal and suboptimal functions. Because the combinatorial functions are generated by the properties of single regulators, it is sensible to extract the properties of single regulators compatible with all optimal and suboptimal functions. In the example stated above, the property "r_1 is a necessary activator" holds under all optimal functions. Thus it is more reliable than any specific optimal function.

We evaluate the significance of a statement about a regulator's direction of effectiveness by investigating whether negating this statement would substantially reduce the score of the model. We calculate the gap of likelihood scores between the optimal model where the statement holds and the optimal model where the statement does not hold. For example, to evaluate the significance of "r_1 is a necessary activator", we find the optimal model M_1 where r_1 is a necessary activator and the optimal model M_0 where r_1 is not a necessary activator. We compare the empirical gap score with the gap scores obtained by randomly permuting gene expression data. Notice the gap score of each permuted data is obtained by re-optimizing the regulatory models to fit the permuted data. The p-value is the fraction of the random gap scores ex-

ceeding the empirical gap. Details about the p-values of combinatorial properties are discussed in the Appendix.

6.4.5 Model finding algorithm

We have introduced each element of the regulatory model finding algorithm. Now we can put them together and summarize the algorithm.

1. Find the candidate regulator sets by identifying all protein-DNA binding patterns which are possessed by a significant number of genes.

2. For each regulator set and each combinatorial function, incrementally add genes which maximize the likelihood score in equation 6.14. Stop adding genes when a consecutive number of newly added genes have p-values > 0.1.

3. Restricting to the size of the smallest regulated gene set, find the combinatorial functions which maximize the likelihood score.

4. Retain the models which yield significant p-values.

5. Combine optimal and suboptimal functions and identify the properties of single regulators extracted from all optimal and suboptimal functions.

6. Evaluate the significance of the combinatorial property of each factor.

7. Report information about model members and single regulator properties extracted from all optimal and suboptimal functions.

6.5 Empirical analysis and discussion

We applied the regulatory model finding algorithm on the CHIP-chip protein-DNA interaction data ([100]) and two sets of mRNA gene expression data: Rosetta compendium data of gene knock-outs ([80]) and the stress response gene expression data published by Gasch et al. ([61]). Rosetta data contains 300 single measurements

Table 6.4: Statistics of candidate regulator sets

# factors	# patterns
0	1
1	105
2	145
3	49
4	8
5	4
6	1
8	1

of different conditions, and both log ratios and p-values are provided. Gasch data contains 173 time-series data points covering 49 different conditions.

Table 6.4 shows the statistics of candidate regulator sets extracted from the protein-DNA binding data. The p-value threshold is 0.005 and only the binding patterns possessed by ≥ 10 genes are considered. There are 314 valid patterns under these criteria. A predominant fraction of these patterns contain ≤ 3 regulators. Only 14 out of 314 patterns have more than 3 regulators. Because it is time consuming to evaluate the likelihood scores on large regulator sets and the results are difficult to interpret, we only considered the candidate regulator sets of 1 to 3 regulators. These sets cover 95% of the candidate regulator sets.

The single factor functions of the 106 transcription factors are determined from literature survey. Table B.8 in the Appendix shows the single factor functions of these regulators and the sources reporting their functions. Few regulators are reported as both activators and repressors (for instance, Sok2). In this case we applied the physical network model described in Chapter Three to Rosetta data with extra constraints that all protein-DNA edges emanating from the same transcription factor have the same sign. The inferred protein-DNA edge sign emanating from a transcription factor was assigned as its function. Alternatively, we may allow single factor functions unfixed and generate the combinatorial functions covering all possible combinations of activators and repressors. Since the number of combinatorial functions multiplies by 2^n folds (n is the number of regulators) in this setting and most transcription factors

have known functions, introducing this degree of freedom may overfit the data, be time-consuming and unnecessary.

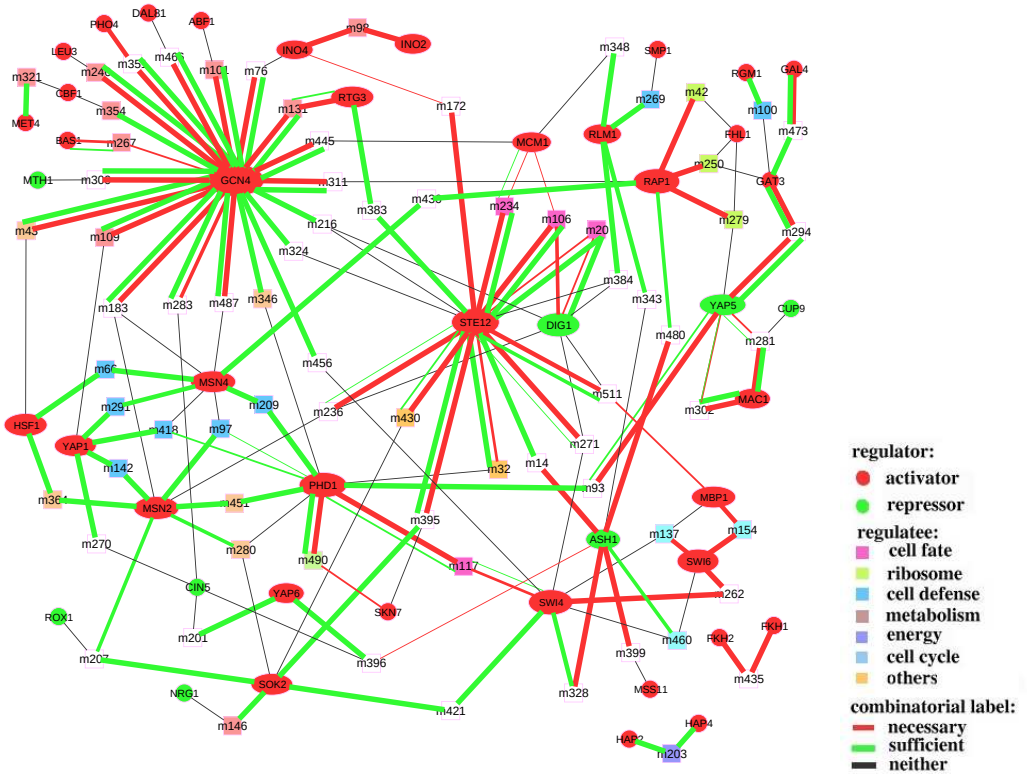
We summarize and analyze the inferred models in the following aspects. First, we visualized the regulatory models inferred from two expression datasets separately and validate them with gene function ontology and literature survey. Second, we investigated the overlap of regulatory models inferred from both expression datasets and study their biological functions. Third, we performed sensitivity analysis of inference results in terms of various free parameters.

6.5.1 Models inferred from Rosetta and Gasch data

Figures 6-2 and 6-3 exhibit various types of information of the regulatory models inferred from Rosetta and Gasch expression data respectively (both sets of models use the same binding data). We represent a regulatory model as a bi-partite subgraph between regulators (circles) and a regulated gene set (a square): a regulator and a regulated gene set are adjacent if they participate in the same model. The color of a circle (regulator) indicates its regulatory function as an activator (red) or repressor (green). The color of a square (regulated gene set) indicates the MIPS functional categories enriched in the regulated gene set. The colors of an edge indicate the direction of effectiveness of a regulator in a model: red for necessary regulator, green for sufficient regulator, and black for neither. Two edges can exist between two nodes since a regulator can be both necessary and sufficient. Notice that we obtained the directions of effectiveness not from the optimal regulatory function alone but by combining the all optimal and suboptimal functions. The width of an edge indicates the confidence about about necessity or sufficiency to expression data as described in Section 6.3.4. We only show the necessary and sufficient edges when their p-values ≤ 0.05. For visual simplicity, we only show the regulatory models with multiple regulators and whose significance of likelihood scores are high (p-value ≤ 0.02 for Rosetta models and p-values ≤ 0.001 for Gasch models). We use the visualization software Cytoscape to draw the graphs.

An immediate observation of the graph topologies in Figures 6-2 and 6-3 is the

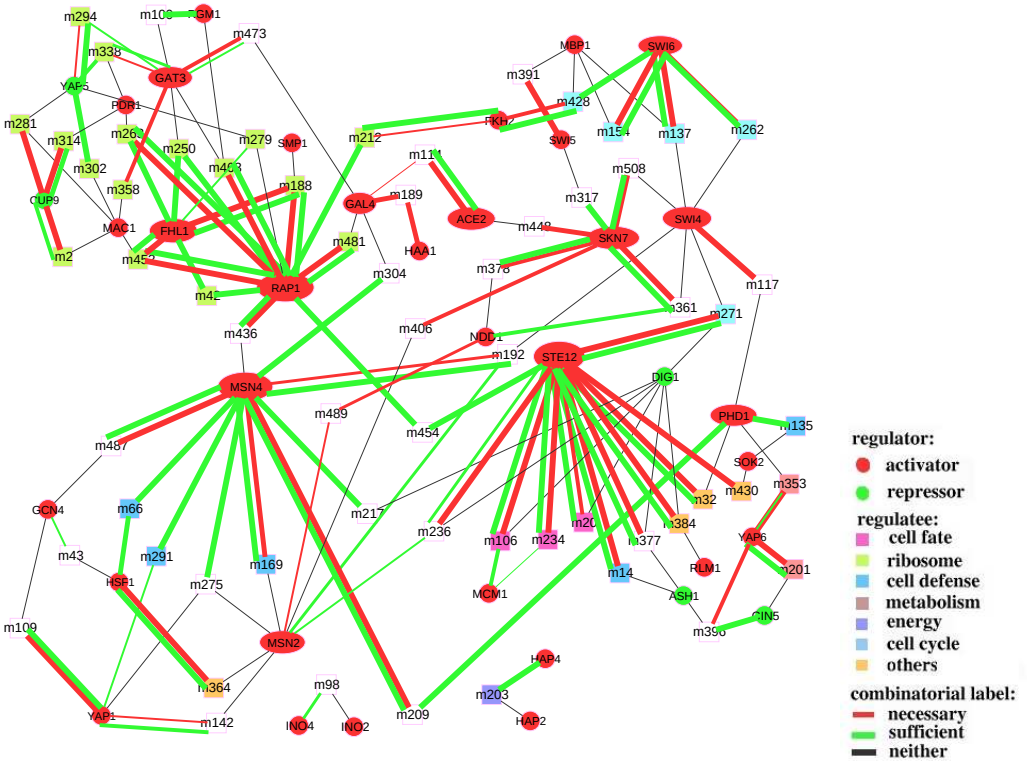
Figure 6-2: Models inferred from Rosetta data



existence of few regulators (hubs) that participate in a relatively large number of regulatory models. The "hub" regulators in Rosetta models are Gcn4, Msn4, Ste12, Dig1, and Rap1, and the hub regulators in Gasch models are Rap1, Msn4, Msn2, Ste12 and Swi6. We suspect that the existence of hubs arises from both the nature of gene regulatory systems and the bias from data. These hub regulators are certainly important for biological processes. For example, Gcn4 is a master regulator for a large number of genes involved in amino acid synthesis, and Ste12 is a key regulator for mating response and invasive growth of yeasts. On the other hand, these regulators appear the most frequently because the biological processes they are involved are intensively probed in the datasets. For example, stress response regulators Msn2 and Msn4 naturally explain a lot of expression changes in the stress response dataset.

We validated the inferred models from four aspects. First, whether regulated gene

Figure 6-3: Models inferred from Gasch data



sets were enriched with genes in the same functional categories. Second, whether regulators in the model were known to control members in the regulated gene set. Third, whether regulators participating in the same model were known to interact. Fourth, whether the inferred directions of effectiveness were consistent with previous studies.

We first validated regulated gene sets with gene function annotations in Munich Information Center for Protein Sequences (MIPS) database of yeast genome. The functional categories in MIPS formed a hierarchy. Table 6.5 enlists the top-level functional categories relevant to yeast cellular processes. We only considered the 11 functional classes when evaluating the functional enrichment of models. The "unknown" category was excluded from the analysis for it was not informative about gene functions.

Table 6.5: Top-level MIPS categories

index	function
01	metabolism
02	energy
03	DNA processing
04	transcription
05	protein synthesis
06	protein fate
08	cellular transport
10	cellular communication
11	stress response
13	regulation with environment
14	cell fate
99	unknown

We evaluated the hyper-geometric p-values of functional enrichment of regulated genes in each model. The hyper-geometric test calculates the probability of randomly choosing a subset of genes which yields the functional enrichment better than the empirical value. Because there are multiple categories, the probability of randomly selecting genes which are enriched in any category is higher than the hyper-geometric p-value from any single category. To correct the bias from multiple hypotheses we approximated the p-value of multiple hypotheses as follows. Denote $p_1, \cdots, p_k$ as the hyper-geometric p-values of n categories, and $\hat{p}_1, \cdots, \hat{p}_k$ as their empirical values. $p_1, \cdots, p_n$ are uniformly distributed within $[0, 1]$ by the definition of p-values. Let $p_{min} = min(p_1, \cdots, p_k)$ be the minimum of the k p-values and $\hat{p}_{min}$ its empirical value. The p-value for multiple categories is

$$\tilde{p} = Pr(p_{min} \leq \hat{p}_{min}) \leq \sum_{i=1}^{k} Pr(p_i \leq \hat{p}_{min}) = k\hat{p}_{min}. \tag{6.15}$$

The inequality follows from the union bound. This construction is similar to the Bonferroni correction. The number of MIPS categories involved (k) is 11.

The verification of the remaining aspects relies on reviewing previous studies. We searched the on-line PubMed database from National Library of Medicine [1] and the

[1] http://www.ncbi.nlm.nih.gov/PubMed/

Table 6.6: Validation of models inferred from Rosetta data, Table 1

Regulator 1	Regulator 2	Regulator 3	Size	Module pval	MIPS	MIPS pval	Interaction
Mcm1			5	$< 10^{-4}$	cell fate	0.032	
Ash1	Ste12		10	$< 10^{-4}$			[19]
Bas1			28	$< 10^{-4}$	metabolism	4.09×10^{-7}	
Dig1	Ste12		57	$< 10^{-4}$	cell fate	7.37×10^{-13}	[13]
Phd1	Ste12		13	$< 10^{-4}$	homeostasis	6.09×10^{-3}	[59]
Rap1			31	$< 10^{-4}$	protein synthesis	2.08×10^{-25}	
Rlm1			35	$< 10^{-4}$	cell fate	9.39×10^{-4}	
Rap1	Fhl1		28	$< 10^{-4}$	protein synthesis	1.08×10^{-27}	[12]
Gcn4	Hsf1		14	$< 10^{-4}$	transport	0.045	[41]
Ste12			49	$< 10^{-4}$	cell fate	1.04×10^{-14}	
Hap4			24	$< 10^{-4}$	energy	1.29×10^{-28}	
Gcn4			125	$< 10^{-4}$	metabolism	5.68×10^{-14}	
Gal4			16	$< 10^{-4}$	metabolism	7.26×10^{-4}	
Msn4	Hsf1		8	$< 10^{-4}$	stress response	0.0026	[61]
Msn4			13	$< 10^{-4}$	stress response	0.0018	
Phd1	Msn2	Msn4	18	$< 10^{-4}$	stress response	0.0102	
Ino2	Ino4		13	$< 10^{-4}$	metabolism	4.30×10^{-5}	[4]
Rgm1	Gat3		8	$< 10^{-4}$	stress response	0.0464	
Gcn4	Abf1		11	$< 10^{-4}$	metabolism	0.0605	[99]
Dig1	Ste12	Mcm1	19	$< 10^{-4}$	cell fate	7.59×10^{-8}	[47]
Yap1	Gcn4		18	$< 10^{-4}$	metabolism	0.0726	[49]
Phd1	Swi4		9	$< 10^{-4}$	cell fate	0.0231	[65]
Rtg3	Gcn4		25	$< 10^{-4}$	metabolism	0.0041	[12]
Phd1	Sok2		52	$< 10^{-4}$	transport	7.172×10^{-5}	[165]
Mbp1	Swi4	Swi6	9	$< 10^{-4}$	cell cycle	0.0103	[139]
Msn2	Yap1		13	$< 10^{-4}$	stress response	0.022	[61]
Nrg1	Sok2		37	$< 10^{-4}$	metabolism	$1.144 \times 10{-}4$	[12]
Mbp1	Swi6		12	$< 10^{-4}$	cell cycle	0.05	[139]
Msn2	Msn4		55	$< 10^{-4}$	stress response	8.69×10^{-7}	[61]
Ash1			48	$< 10^{-4}$	metabolism	0.0209	
Msn2	Msn4	Gcn4	14	$< 10^{-4}$			[32]
Dig1			66	$< 10^{-4}$	cell fate	0.0052	

Incyte Yeast Proteome Database [2] to identify the relations between regulators and regulated genes, between regulated genes, and the consistency of inferred properties with previous works.

Tables 6.6, 6.7 and 6.8, 6.9 show the validation results of the models inferred from Rosetta and Gasch data respectively. Overall, the inference results agree to a large extent with previous studies. By considering the models with high log likelihood values (permutation p-value ≤ 0.02 for Rosetta models and p-value ≤ 0.001 for Gasch models, including the models of single regulators), 48% of the Rosetta models (53 out of 110) and 38% of the Gasch models (84 out of 220) are enriched with at least one MIPS category (hyper-geometric p-value for multiple categories ≤ 0.06). The results suggest that genes in these models are likely to be co-regulated since many of them

[2] https://www.incyte.com/tools/proteome/databases.jsp

Table 6.7: Validation of models inferred from Rosetta data, Table 2

Regulator 1	Regulator 2	Regulator 3	Size	Module pval	MIPS	MIPS pval	Interaction
Cin5	Yap6		6	$< 10^{-4}$			[58]
Hap4	Hap2		11	8×10^{-4}	energy	8.294×10^{-9}	[109]
Phd1	Msn4		15	$< 10^{-4}$	stress response	0.004	
Ste12	Mcm1		24	$< 10^{-4}$	cell fate	9.34×10^{-9}	[47]
Met4			16	$< 10^{-4}$	metabolism	7.96×10^{-5}	
Gcn4	Leu3		8	$< 10^{-4}$	metabolism	3.35×10^{-4}	[170]
Rap1	Gat3	Fhl1	20	$< 10^{-4}$	protein synthesis	8.70×10^{-25}	[12]
Swi4	Swi6		10	$< 10^{-4}$			[139]
Gcn4	Bas1		8	$< 10^{-4}$	metabolism	3.36×10^{-4}	[7]
Smp1	Rlm1		11	$< 10^{-4}$	stress response	0.0112	[40]
Cin5	Yap1		12	10^{-4}			[49]
Msn2	Msn4	Yap1	17	$< 10^{-4}$	stress response	0.0649	[61]
Rap1	Fhl1	Yap5	20	$< 10^{-4}$	protein synthesis	3.047×10^{-22}	[12]
Cin5	Gcn4		12	$< 10^{-4}$			[43]
Hir2			7	$< 10^{-4}$	transcription	4.59×10^{-6}	
Msn4	Yap1		22	$< 10^{-4}$	stress response	3.53×10^{-4}	[61]
Rap1	Gcn4		8	$< 10^{-4}$			[36]
Cbf1	Met4		8	$< 10^{-4}$	metabolism	0.0059	[16]
Ste12	Gcn4		14	$< 10^{-4}$			[131]
Rtg3			47	$< 10^{-4}$	metabolism	0.0046	
Phd1	Gcn4		20	$< 10^{-4}$	transport	0.0035	
Cbf1	Gcn4		6	$< 10^{-4}$	metabolism	0.1086	[117]
Msn2	Hsf1		10	$< 10^{-4}$	stress response	0.0319	[61]
Msn2	Rlm1		6	0.0186			[69]
Phd1	Msn4	Yap1	7	9×10^{-4}	stress response	0.0014	
Sok2	Swi4		7	$< 10^{-4}$			[5]
Mbp1	Swi6	Fkh2	8	0.0015			[139]
Ste12	Sok2		11	$< 10^{-4}$	homeostasis	0.0469	
Fkh1	Fkh2		12	$< 10^{-4}$			[139]
Mcm1	Gcn4		6	$< 10^{-4}$			[110]
Phd1	Msn2		22	$< 10^{-4}$	transport	0.0061	
Ash1	Swi4	Swi6	8	$< 10^{-4}$	cell cycle	0.058	
Rap1	Rgm1	Gat3	12	0.001	protein synthesis	1.09×10^{-14}	
Msn4	Gcn4		20	$< 10^{-4}$			[32]
Phd1	Skn7		9	$< 10^{-4}$			[104]

Table 6.8: Validation of models inferred from Gasch data, Table 1

Regulator 1	Regulator 2	Regulator 3	Size	Module pval	MIPS	pvalue	Interaction
Phd1			38	$< 10^{-4}$	stress response	0.0143	
Mac1	Cup9		17	$< 10^{-4}$	protein synthesis	5.203×10^{-11}	[12]
Mth1			19	$< 10^{-4}$	protein synthesis	0.0385	
Cbf1			27	$< 10^{-4}$	transcription	0.0407	
Reb1			37	$< 10^{-4}$	transcription	0.0418	
Ash1	Ste12		16	$< 10^{-4}$	stress response	0.0516	[19]
Sok2			44	$< 10^{-4}$	protein synthesis	0.033	
Dig1	Ste12		46	$< 10^{-4}$	cell fate	2.365×10^{-6}	[13]
Mot3			7	$< 10^{-4}$	metabolism	0.0257	
Cad1			34	$< 10^{-4}$	stress response	0.007	
Phd1	Ste12		11	$< 10^{-4}$	cell communication	0.0517	[59]
Rap1			148	$< 10^{-4}$	protein synthesis	9.49×10^{-94}	
Put3			15	$< 10^{-4}$	metabolism	0.0027	
Zap1			24	$< 10^{-4}$	protein synthesis	0.003	
Rap1	Fhl1		121	$< 10^{-4}$	protein synthesis	1.1×10^{-107}	[12]
Gcn4	Hsf1		13	$< 10^{-4}$			[41]
Fkh1			9	$< 10^{-4}$	protein synthesis	0.0154	
Ste12			107	$< 10^{-4}$	cell fate	1.903×10^{-6}	
Hap4			40	$< 10^{-4}$	energy	5.5×10^{-32}	
Gat3			52	$< 10^{-4}$	protein synthesis	5.379×10^{-43}	
Hap3			21	$< 10^{-4}$	energy	0.0013	
Mcm1	Ndd1	Fkh2	17	$< 10^{-4}$			[139]
Msn4	Hsf1		22	$< 10^{-4}$	stress response	0.0034	[61]
Cup9			46	$< 10^{-4}$	protein synthesis	2.189×10^{-24}	
Sum1			43	$< 10^{-4}$	cell fate	1.265×10^{-4}	
Fhl1			130	$< 10^{-4}$	protein synthesis	1.463×10^{-121}	
Sfp1			39	$< 10^{-4}$	protein synthesis	5.26×10^{-17}	
Msn4			16	$< 10^{-4}$	stress response	0.0517	
Hsf1			63	$< 10^{-4}$	protein fate	0.0627	
Ino2	Ino4		22	$< 10^{-4}$			[4]
Gcn4	Abf1		6	$< 10^{-4}$	protein synthesis	0.0016	[99]
Dig1	Ste12	Mcm1	25	$< 10^{-4}$	cell fate	3.06×10^{-6}	[47]
Yap1	Gcn4		11	$< 10^{-4}$			[49]
Phd1	Swi4		15	$< 10^{-4}$			[65]
Rtg3	Gcn4		18	$< 10^{-4}$			[12]
Swi6			36	$< 10^{-4}$	cell cycle	5.26×10^{-5}	
Phd1	Sok2		18	$< 10^{-4}$	stress response	0.08	[165]
Mbp1	Swi4	Swi6	15	$< 10^{-4}$	cell cycle	3.37×10^{-5}	[139]
Msn2	Yap1		25	$< 10^{-4}$			[61]
Yap1			10	$< 10^{-4}$	stress response	4.015×10^{-4}	
Mbp1	Swi6		19	$< 10^{-4}$	cell cycle	4.147×10^{-6}	
Yap6			27	$< 10^{-4}$	metabolism	0.0023	
Yap5			62	$< 10^{-4}$	protein synthesis	4.675×10^{-39}	
Msn2	Msn4		31	$< 10^{-4}$	stress response	5.95×10^{-5}	[61]
Smp1			23	$< 10^{-4}$	protein synthesis	1.43×10^{-7}	
Msn2	Msn4	Gcn4	7			[32]	
Rap1	Smp1	Fhl1	14	$< 10^{-4}$	protein synthesis	8.95×10^{-13}	[12]
Cin5	Yap6		17	$< 10^{-4}$	metabolism	0.0015	[58]
Hap4	Hap2		11	$< 10^{-4}$	energy	8.294×10^{-9}	[109]
Met31			26	$< 10^{-4}$	protein synthesis	7.183×10^{-4}	
Rap1	Fkh2		16	$< 10^{-4}$	protein synthesis	0.0165	[12]
Rox1			12	$< 10^{-4}$	protein synthesis	0.0037	
Ime4			16	$< 10^{-4}$	stress response	0.0516	

Table 6.9: Validation of models inferred from Gasch data, Table 2

Regulator 1	Regulator 2	Regulator 3	Size	Module pval	MIPS	pvalue	Interaction
Hap3	Hap4		5	$< 10^{-4}$	energy	1.243×10^{-4}	[109]
Cbf1	Abf1		5	$< 10^{-4}$			[118]
Ste12	Mcm1		30	$< 10^{-4}$	cell fate	3.08×10^{-6}	[47]
Mcm1	Skn7		9	$< 10^{-4}$			[102]
Gcn4	Leu3		5	$< 10^{-4}$	metabolism	0.0419	[170]
Rap1	Gat3	Fhl1	61	$< 10^{-4}$	protein synthesis	5.984×10^{-54}	[12]
Rap1	Rcs1		5	$< 10^{-4}$	protein synthesis	5.478×10^{-4}	
Rcs1			13	$< 10^{-4}$	protein synthesis	0.0539	
Swi4	Swi6		22	$< 10^{-4}$	cell cycle	2.45×10^{-4}	[139]
Pdr1	Rap1	Fhl1	28	$< 10^{-4}$	protein synthesis	1.08×10^{-27}	[12]
Cin5	Rtg1		7	0.0004	metabolism	0.0253	
Gcn4	Bas1		7	$< 10^{-4}$	metabolism	0.0253	[7]
Dig1	Ste12	Swi4	15	$< 10^{-4}$	cell cycle	0.0264	
Msn2	Msn4	Yap1	24	$< 10^{-4}$			[61]
Rap1	Fhl1	Yap5	52	$< 10^{-4}$	protein synthesis	6.66×10^{-45}	[12]
Mac1	Cup9	Yap5	10	$< 10^{-4}$	protein synthesis	3.553×10^{-12}	[12]
Cin5	Gcn4		8	0.0029			[43]
Hir2			27	$< 10^{-4}$	transcription	0.0396	
Msn4	Yap1		22	$< 10^{-4}$	stress response	3.531×10^{-4}	[61]
Gat3	Yap5		38	$< 10^{-4}$	protein synthesis	5.159×10^{-30}	[12]
Mac1	Yap5		18	$< 10^{-4}$	protein synthesis	1.375×10^{-18}	[12]
Fzf1			24	$< 10^{-4}$	transcription	0.0158	
Rap1	Gcn4		5	$< 10^{-4}$	protein synthesis	0.0187	[36]
Pdr1	Cup9		16	$< 10^{-4}$	protein synthesis	1.606×10^{-11}	[12]
Pdr1	Gat3	Yap5	25	$< 10^{-4}$	protein synthesis	1.672×10^{-9}	[12]
Mcm1	Rlm1		7	$< 10^{-4}$	stress response	0.0014	[166]
Phd1	Yap6		14	$< 10^{-4}$	metabolism	0.0451	
Cbf1	Gcn4		5	0.009	metabolism	0.0418	[117]
Rap1	Pho4		6	$< 10^{-4}$	protein synthesis	0.0352	
Mac1	Gat3		14	$< 10^{-4}$	protein synthesis	8.95×10^{-13}	[12]
Msn2	Hsf1		26	$< 10^{-4}$	stress response	0.0242	[61]
Yap1	Cad1		11	$< 10^{-4}$			[24]
Mbp1	Swi5		12	$< 10^{-4}$			[139]
Rap1	Swi4		5	$< 10^{-4}$	protein synthesis	5.478×10^{-4}	
Msn2	Rlm1		6	0.0021			[69]
Sok2	Swi4		8	$< 10^{-4}$			[5]
Mbp1	Swi6	Fkh2	10	$< 10^{-4}$	cell cycle	0.0017	[139]
Ste12	Sok2		12	$< 10^{-4}$	cell communication	0.0616	
Msn4	Rtg3		8	$< 10^{-4}$			[28]
Fkh1	Fkh2		24	$< 10^{-4}$			[139]
Reb1	Abf1		5	$< 10^{-4}$			[52]
Abf1	Hsf1		5	0.0005	protein synthesis	0.0825	[90]
Phd1	Msn2		25	$< 10^{-4}$	stress response	0.0473	[119]
Mac1	Rap1	Fhl1	19	$< 10^{-4}$	protein synthesis	5.335×10^{-21}	[12]
Pdr1	Smp1	Yap5	6	$< 10^{-4}$	protein synthesis	0.0016	
Rap1	Rgm1	Gat3	38	$< 10^{-4}$	protein synthesis	5.159×10^{-30}	[12]
Rap1	Yap6		6	$< 10^{-4}$	protein synthesis	0.0016	
Rme1	Rox1		7	$< 10^{-4}$	protein synthesis	0.0596	
Rap1	Gal4		11	$< 10^{-4}$	protein synthesis	5.522×10^{-6}	
Msn4	Gcn4		10	$< 10^{-4}$			[32]
Phd1	Skn7		6	0.0039			[104]
Mcm1	YJL206C		5	0.0001	energy	0.0062	
Fkh1	Ndd1	Fkh2	7	$< 10^{-4}$			[139]

are involved in the same cellular processes. Since the functional annotations in MIPS are incomplete, we expect more models are enriched with genes involved in the same process.

We also found a strong consistency between the biological functions of regulators and the functional enrichment of regulated models. Here we summarize these relations. In Rosetta models, models enriched with metabolism genes are predominantly regulated by Gcn4. This is sensible since Gcn4 is a master regulator for genes involved in amino acid synthesis ([115]). Moreover, in each of these models Gcn4 often pairs with another biosynthesis regulator to perform specific function. Examples include Bas1 (histidine and arginine, [43]), Leu3 (leucine, [170]), Cbf1 (methionine, [117]), and Rtg3 ([12]). A small number of phospholypid synthesis genes are regulated by Ino2 and Ino4, which are known to regulate phospholypid synthesis genes ([4]). Models enriched with stress response genes are primarily regulated by Msn2, Msn4, Yap1 and Hsf1. These regulators are known to be involved in stress response ([61]). Models enriched with genes of cell fate (such as mating or invasive growth for yeasts) are mostly regulated by Ste12, Dig1 and Mcm1. These regulators are well known to control the fates of yeast cells ([47]). Furthermore, many ribosomal genes are regulated by Rap1 and Fhl1. There is yet no direct evidence in genetics or molecular biology suggesting Rap1 and Fhl1 regulate ribosomal genes. However, this is indirectly supported from other computational works of combining the same binding data but different expression data ([12]). The highly significant enrichment of ribosomal genes is unlikely due to errors or coincidence.

The relations between the known functions of regulators and regulated genes in Gasch models are similar to Rosetta models. Msn2, Msn4, Yap1 and Hsf1 are again involved in regulating stress response genes; Rap1 and Fhl1 regulate ribosomal genes; and Ste12 regulates genes involved in cell fate. There are several regulators whose functions are revealed only in Gasch data. Swi4, Swi6 and Mbp1 participate in several models enriched with cell cycle genes. This is consistent with the functions of these regulators as cell cycle transcription factors ([139]). Hap2 and Hap4 participate in a model enriched with genes involved in respiration. This is also consistent with the

functions of Hap-complex ([109]). On the other hand, the function of Gcn4 is hardly revealed in Gasch data. Although the amino acid starvation experiments in Gasch data should invoke the activation of Gcn4, Gcn4 is not significantly changed in these experiments. The function of Dig1 is also not revealed in Gasch data due to similar reasons.

Tables 6.6, 6.7 and 6.8, 6.9 also show the regulatory models whose regulators participate in the same biological functions (regulators interact functionally) according to previous studies. Here we only consider the models with multiple regulators. 44 out of 85 models inferred from Rosetta data and 61 out of 114 models from Gasch data contain regulators which are known to interact. Many interacting regulators are already discussed earlier, for example, Ste12, Dig1 and Mcm1, Msn2, Msn4, Hsf1 and Yap1, Gcn4 and other biosynthesis regulators, Ino2 and Ino4, Rap1 and Fhl1, Hap2 and Hap4, Swi4, Swi6 and Mbp1. Several other interacting regulators include Ste12 and Phd1 (filamentous growth, [59]), Cbf1 and Met4 (methionine synthesis, [16]), Fkh1 and Fkh2 (control genes expressed at G2/M phase during cell cycle, [139]), and Ash1 and Ste12 (pseudohyphal growth, [19]).

We then investigated the inferred directions of effectiveness and compared them to previous studies. Interestingly, despite we allowed the directions of effectiveness to be model dependent, most regulators possessed a consistent direction across the models in which they participated. In Rosetta data, for example, Ste12 is a necessary activator (edge colors are red) for most of its regulated models. This is compatible with our knowledge that Ste12 is required for the activation of mating response genes. Similarly, stress response regulators (Msn2, Msn4, Yap1, Hsf1) are predominantly sufficient activators. This is also consistent with the knowledge that each regulator suffices to activate stress response. Rap1 is a necessary activator for ribosomal genes. Gcn4 is primarily a sufficient activator in models shown in Figure 6-3. This seems to contradict with the fact that Gcn4Δ experiment is in Rosetta and genes bound by Gcn4 are down-regulated in this experiment. By inspecting the model regulated by Gcn4 alone, we find Gcn4 was also a necessary activator with high confidence. Thus we suspect most models in Figure 6-3 do not respond in Gcn4Δ but are up-

Table 6.10: Directions of effectiveness of models inferred from Rosetta data, Table 1

Regulator 1	Nece pval	Suff pval	Regulator 2	Nece pval	Suff pval	Regulator 3	Nece pval	Suff pval
Msn2	1.0	10^{-4}	Msn4	1.0	0.7	Hsf1	1.0	0.002
Ash1	$< 10^{-4}$	1.0	Ste12	1.0	$< 10^{-4}$			
Dig1	0.044	$< 10^{-4}$	Ste12	0.019	$< 10^{-4}$			
Phd1	1.0	0.184	Ste12	0.0059	$< 10^{-4}$			
Rap1	$< 10^{-4}$	1.0	Fhl1	1.0	1.0			
Gcn4	$< 10^{-4}$	$< 10^{-4}$	Hsf1	1.0	1.0			
Msn4	1.0	$< 10^{-4}$	Hsf1	1.0	$< 10^{-4}$			
Phd1	1.0	0.055	Msn2	1.0	$< 10^{-4}$	Msn4	1.0	0.96
Ino2	$< 10^{-4}$	1.0	Ino4	$< 10^{-4}$	1.0			
Rgm1	1.0	$< 10^{-4}$	Gat3	1.0	1.0			
Gcn4	$< 10^{-4}$	$< 10^{-4}$	Abf1	1.0	1.0			
Dig1	0.0041	0.181	Ste12	$< 10^{-4}$	$< 10^{-4}$	Mcm1	0.076	1.0
Yap1	1.0	0.99	Gcn4	$< 10^{-4}$	$< 10^{-4}$			
Phd1	$< 10^{-4}$	0.0368	Swi4	0.0052	0.0511			
Rtg3	$< 10^{-4}$	0.049	Gcn4	$< 10^{-4}$	$< 10^{-4}$			
Phd1	1.0	0.89	Sok2	1.0	0.84			
Mbp1	0.28	1.0	Swi4	0.96	1.00	Swi6	8×10^{-4}	1.0
Msn2	1.0	$< 10^{-4}$	Yap1	1.0	$< 10^{-4}$			
Nrg1	0.64	1.0	Sok2	1.0	$< 10^{-4}$			
Mbp1	6×10^{-4}	1.0	Swi6	$< 10^{-4}$	1.0			
Msn2	1.0	0.97	Msn4	1.0	$< 10^{-4}$	Gcn4	$< 10^{-4}$	$< 10^{-4}$
Cin5	0.985	1.0	Yap6	1.0	$< 10^{-4}$			
Hap4	1.0	$< 10^{-4}$	Hap2	1.0	$< 10^{-4}$			
Phd1	1.0	$< 10^{-4}$	Msn4	1.0	$< 10^{-4}$			
Ste12	$< 10^{-4}$	$< 10^{-4}$	Mcm1	0.073	0.069			
Gcn4	$< 10^{-4}$	$< 10^{-4}$	Leu3	0.13	0.23			
Rap1	$< 10^{-4}$	1.0	Gat3	1.0	1.0	Fhl1	1.0	1.0
Swi4	$< 10^{-4}$	1.0	Swi6	$< 10^{-4}$	1.0			
Gcn4	0.0241	0.72	Bas1	0.009	0.03			
Smp1	1.0	0.925	Rlm1	1.0	$< 10^{-4}$			
Cin5	0.88	1.0	Yap1	1.0	$< 10^{-4}$			
Msn2	1.0	0.054	Msn4	1.0	0.179	Yap1	1.0	0.243
Rap1	$< 10^{-4}$	1.0	Fhl1	1.0	1.0	Yap5	1.0	0.946
Phd1	1.0	0.253	Msn2	1.0	0.003	Sok2	1.0	0.935
Cin5	0.99	1.0	Gcn4	0.0026	$< 10^{-4}$			

Table 6.11: Directions of effectiveness of models inferred from Rosetta data, Table 2

Regulator 1	Nece pval	Suff pval	Regulator 2	Nece pval	Suff pval	Regulator 3	Nece pval	Suff pval
Msn4	1.0	10^{-4}	Yap1	1.0	$< 10^{-4}$			
Mac1	$< 10^{-4}$	$< 10^{-4}$	Yap5	0.02	0.078			
Rap1	1.0	1.0	Gcn4	$< 10^{-4}$	$< 10^{-4}$			
Cbf1	1.0	0.99	Met4	1.0	$< 10^{-4}$			
Phd1	1.0	0.98	Gcn4	1.0	$< 10^{-4}$			
Cbf1	1.0	1.0	Gcn4	1.0	$< 10^{-4}$			
Msn2	1.0	$< 10^{-4}$	Hsf1	1.0	$< 10^{-4}$			
Phd1	1.0	0.0175	Msn4	1.0	0.95	Yap1	1.0	$< 10^{-4}$
Sok2	1.0	0.0175	Swi4	1.0	$< 10^{-4}$			
Mbp1	0.01	1.0	Swi6	0.0245	1.0	Fkh2	0.98	1.0
Ste12	$< 10^{-4}$	0.045	Sok2	0.99	1.0			
Fkh1	$< 10^{-4}$	1.0	Fkh2	$< 10^{-4}$	1.0			
Mcm1	1.0	1.0	Gcn4	$< 10^{-4}$	$< 10^{-4}$			
Phd1	1.0	$< 10^{-4}$	Msn2	1.0	$< 10^{-4}$			
Ash1	1.0	0.0046	Swi4	0.172	1.0	Swi6	0.14	1.0
Rap1	$< 10^{-4}$	1.0	Rgm1	0.0034	1.0	Gat3	1.0	1.0
Phd1	$< 10^{-4}$	$< 10^{-4}$	Skn7	0.028	0.103			

Table 6.12: Directions of effectiveness of models inferred from Gasch data, Table 1

Regulator 1	Nece pval	Suff pval	Regulator 2	Nece pval	Suff pval	Regulator 3	Nece pval	Suff pval
Msn2	1.0	9×10^{-4}	Msn4	9×10^{-4}	$< 10^{-4}$	Hsf1	1.0	0.98
Mac1	0.96	1.0	Cup9	$< 10^{-4}$	2.10×10^{-3}			
Ash1	1.0	1.0	Ste12	$< 10^{-4}$	$< 10^{-4}$			
Dig1	1.0	1.0	Ste12	0.001	$< 10^{-4}$			
Phd1	1.0	1.0	Ste12	$< 10^{-4}$	$< 10^{-4}$			
Rap1	0.994	$< 10^{-4}$	Fhl1	1.0	$< 10^{-4}$			
Gcn4	1.0	0.0316	Hsf1	0.636	0.515			
Mcm1	0.880	1.0	Ndd1	0.685	1.0	Fkh2	0.132	1.0
Msn4	1.0	$< 10^{-4}$	Hsf1	0.99	5×10^{-4}			
Ino2	1.0	0.878	Ino4	1.0	8.30×10^{-3}			
Gcn4	0.0121	1.0	Abf1	3.00×10^{-4}	$< 10^{-4}$			
Dig1	1.0	1.0	Ste12	$< 10^{-4}$	$< 10^{-4}$	Mcm1	1.0	0.00270
Yap1	$< 10^{-4}$	$< 10^{-4}$	Gcn4	1.0	1.0			
Phd1	1.0	0.369	Swi4	$< 10^{-4}$	0.99			
Rtg3	1.0	0.291	Gcn4	1.0	0.290			
Phd1	1.0	$< 10^{-4}$	Sok2	1.0	0.907			
Mbp1	0.863	1.0	Swi4	1.0	1.0	Swi6	$< 10^{-4}$	$< 10^{-4}$
Msn2	1.0	0.979	Yap1	0.0271	0.002			
Nrg1	$< 10^{-4}$	$< 10^{-4}$	Sok2	1.0	1.0			
Mbp1	1.0	1.0	Swi6	$< 10^{-4}$	$< 10^{-4}$			
Msn2	1.0	1.0	Msn4	3×10^{-4}	$< 10^{-4}$			
Msn2	1.0	0.0043	Msn4	$< 10^{-4}$	$< 10^{-4}$	Gcn4	1.0	1.0
Rap1	$< 10^{-4}$	$< 10^{-4}$	Smp1	1.0	1.0	Fhl1	$< 10^{-4}$	$< 10^{-4}$
Cin5	0.386	1.0	Yap6	$< 10^{-4}$	$< 10^{-4}$			
Hap4	1.0	$< 10^{-4}$	Hap2	1.0	1.0			
Phd1	1.0	$< 10^{-4}$	Msn4	$< 10^{-4}$	$< 10^{-4}$			
Hap3	$< 10^{-4}$	2×10^{-4}	Hap2	$< 10^{-4}$	0.390			
Cbf1	$< 10^{-4}$	1.0	Abf1	0.447	0.363			
Ste12	$< 10^{-4}$	$< 10^{-4}$	Mcm1	0.834	0.0705			
Mcm1	0.931	1.0	Skn7	$< 10^{-4}$	$< 10^{-4}$			
Gcn4	0.0694	1.0	Leu3	$< 10^{-4}$	$< 10^{-4}$			
Rap1	0.993	$< 10^{-4}$	Gat3	1.0	1.0	Fhl1	1.0	$< 10^{-4}$
Rap1	$< 10^{-4}$	0.0224	Rcs1	1.0	1.0			
Swi4	1.0	1.0	Swi6	0.0361	$< 10^{-4}$			
Pdr1	1.0	1.0	Rap1	$< 10^{-4}$	$< 10^{-4}$	Fhl1	1.0	$< 10^{-4}$
Cin5	1.0	0.0037	Rgt1	1.0	0.277			
Gcn4	10^{-4}	0.226	Bas1	0.0138	0.0668			
Dig1	1.0	1.0	Ste12	$< 10^{-4}$	$< 10^{-4}$	Swi4	1.0	1.0
Msn2	1.0	1.0	Msn4	1.0	$< 10^{-4}$	Yap1	0.138	0.263
Rap1	0.449	0.449	Fhl1	0.925	0.0438	Yap5	1.0	1.0
Mac1	0.965	1.0	Cup9	$< 10^{-4}$	0.999	Yap5	1.0	0.236
Cin5	1.0	$< 10^{-4}$	Gcn4	1.0	1.0			
Msn4	1.0	$< 10^{-4}$	Yap1	0.597	0.0295			
Gat3	0.764	0.0199	Yap5	0.0461	$< 10^{-4}$			
Rap1	$< 10^{-4}$	$< 10^{-4}$	Gcn4	0.979	1.0			
Pdr1	1.0	1.0	Cup9	$< 10^{-4}$	10^{-4}			
Pdr1	1.0	1.0	Gat3	0.0155	0.0073	Yap5	1.0	0.0029
Mcm1	$< 10^{-4}$	2×10^{-4}	Rlm1	1.0	0.567			
Phd1	1.0	1.0	Yap6	$< 10^{-4}$	0.0101			
Cbf1	0.0024	0.378	Gcn4	1.0	0.005			
Rap1	$< 10^{-4}$	$< 10^{-4}$	Pho4	1.0	1.0			
Mac1	0.538	0.399	Gat3	0.0012	1.0			
Msn2	1.0	0.879	Hsf1	$< 10^{-4}$	$< 10^{-4}$			
Dig1	1.0	1.0	Ste12	$< 10^{-4}$	$< 10^{-4}$	Rlm1	1.0	1.0
Yap1	1.0	0.969	Cad1	0.131	0.501			
Mbp1	0.661	1.0	Swi5	$< 10^{-4}$	0.661			

Table 6.13: Directions of effectiveness of models inferred from Gasch data, Table 2

Regulator 1	Nece pval	Suff pval	Regulator 2	Nece pval	Suff pval	Regulator 3	Nece pval	Suff pval
Rap1	$< 10^{-4}$	1.0	Swi4	1.0	1.0			
Msn2	0.0045	$< 10^{-4}$	Rlm1	0.997	1.0			
Sok2	$< 10^{-4}$	1.0	Swi4	2×10^{-4}	0.742			
Mbp1	1.0	1.0	Swi6	0.978	$< 10^{-4}$	Fkh2	0.0029	$< 10^{-4}$
Ste12	$< 10^{-4}$	0.139	Sok2	0.678	1.0			
Msn4	1.0	$< 10^{-4}$	Rtg3	0.009	0.223			
Fkh1	0.638	1.0	Fkh2	1.0	0.130			
Reb1	6×10^{-4}	0.935	Abf1	1.0	1.0			
Abf1	1.0	0.961	Hsf1	$< 10^{-4}$	$< 10^{-4}$			
Phd1	1.0	0.540	Msn2	1.0	0.0649			
Mac1	1.0	1.0	Rap1	$< 10^{-4}$	$< 10^{-4}$	Fhl1	$< 10^{-4}$	$< 10^{-4}$
Pdr1	1.0	1.0	Smp1	0.148	1.0	Yap5	0.0018	$< 10^{-4}$
Rap1	$< 10^{-4}$	$< 10^{-4}$	Rgm1	1.0	1.0	Gat3	1.0	1.0
Rap1	$< 10^{-4}$	$< 10^{-4}$	Yap6	1.0	1.0			
Rme1	1.0	1.0	Rox1	$< 10^{-4}$	$< 10^{-4}$			
Rap1	$< 10^{-4}$	$< 10^{-4}$	Gal4	1.0	1.0			
Msn4	$< 10^{-4}$	$< 10^{-4}$	Gcn4	1.0	1.0			
Phd1	1.0	$< 10^{-4}$	Skn7	1.0	1.0			
Mcm1	0.117	1.0	YJL206C	$< 10^{-4}$	1.0			
Fkh1	0.0525	1.0	Ndd1	0.960	1.0	Fkh2	0.169	0.0031

regulated in other experiments where Gcn4 was also up-regulated. The properties of these regulators are preserved in Gasch data. Msn2, Msn4, Yap1 and Hsf1 remain sufficient activators for stress response models, Rap1 is a necessary activator, and Ste12 is a necessary activator for mating response genes. However, there is moderate evidence to support these regulators also function in the opposite direction. For example, Rap1 is also a sufficient regulator for ribosomal genes, the model regulated by Msn2 and Msn4 shows they are both necessary and sufficient. The same statement is true for Ste12 and Swi6.

By investigating the directions of effectiveness of multiple regulators in different models, we summarize three patterns of regulation scenarios and suggest possible mechanisms for these scenarios.

- Each regulator in a model is a sufficient regulator. This scenario is common for stress response genes regulated by Msn2, Msn4, Hsf1, Yap1 and Cad1. In this scenario, each regulator suffices to activate genes. A possible mechanism is that these regulators bind independently on the promoters and serve as redundant pathways.

- A model contains a significant necessary or sufficient regulator and other regu-

lators which are not well correlated with regulated genes in expression data. For instance, in Rosetta data Gcn4 pairs up with several other regulators to control a number of models. In these models Gcn4 is both necessary and sufficient, but the confidence about the effectiveness direction of the other regulator is low. The interactions between Rap1 and Fhl1 and Yap5 on ribosomal genes for Rosetta data, and between Swi6, Swi4 and Mbp1 on cell cycle genes for Gasch data also fall in this category. This other regulator may not play any functional roles thus can be discarded. However, they may also control transcription not by modulating their mRNA abundance but through other mechanisms. For example, a regulator may constituently stay on promoters and its function is not manifested without its deletion.

- Another scenario is each regulator is a necessary regulator. Examples include Ino2 and Ino4, Swi4 and Swi6, Fkh1 and Fkh2 in Rosetta data. A variant of this scenario is each regulator is necessary but some are also sufficient. An example is Ste12 and Mcm1 in Rosetta data. A likely mechanism is that they form a complex when binding to promoters. This is indeed the case for Ino2-Ino4 complex ([4]), Hap2-Hap4 complex ([109]), Fkh1-Fkh2 complex ([139]), and Ste12-Dig1 complex ([13]).

6.5.2 Overlap between Rosetta and Gasch models

The quality of inferred models can be judged by the robustness of models with respect to different datasets. We have generated two sets of regulatory models from Rosetta and Gasch expression data respectively. A natural question is to what extent these models are overlapped. The consensus parts of the models suggest they are more likely to reflect the underlying system. The parts where they differ may be due to condition-specific properties of gene regulation: certain regulatory systems are revealed only in one dataset. However, without further validation these results are less confident.

We investigated the overlapped parts between Rosetta and Gasch models. There were 110 Rosetta models and 220 Gasch models which fit the data well (permutation

p-value ≤ 0.02 for Rosetta models and ≤ 0.001 for Gasch models). We considered the intersection of the regulator sets of these models and found 89 regulator sets appear in both sets of models. We then compared Rosetta and Gasch models corresponding to these 89 regulator sets in two aspects. First we checked the overlap between the regulated gene sets of the corresponding models. Second we inspected the consensus of inferred directions of effectiveness of the corresponding models.

Tables 6.14 and 6.15 shows the comparison results of Rosetta and Gasch models. The semantics of columns is as follows. SizeR denotes the number of regulated genes of Rosetta models and SizeG the number of regulated genes of Gasch models. Each bit in NeceR and SuffR columns indicates whether each regulator is necessary or sufficient in Rosetta models. NeceG and SuffG denote the same properties about Gasch models. Overlap rate 1 is the number of genes which appear in both Rosetta and Gasch models divided by the number of genes in the Rosetta model. Overlap rate 2 is the overlap with respect to the number of genes in the Gasch model. The results suggest that these two sets of models are significantly overlapped. Among the 89 regulatory model pairs, about 55% of them (49 out of 89) are significantly overlapped in their regulated gene sets (more than 40% members with respect to either model are overlapped). 55% of model pairs (49 out of 89) agree upon either direction of effectiveness for all their regulators. That means, all necessary regulators or all sufficient regulators in the two models coincide.

By inspecting the overlapped regulatory models, we found they corresponded to the regulatory processes which were better captured in both datasets. Among the 89 significant models whose regulator sets appear in both datasets, 61 of them are enriched with genes belonging to certain MIPS categories. The regulatory processes involved in these overlapped models include stress responses, mating responses, ribosomal regulation and cell cycle.

6.5.3 Sensitivity analysis of inferred models

The validity of a model is questionable if the modeling results are sensitive to specific settings of its parameters. In this section we show the inferred models are robust

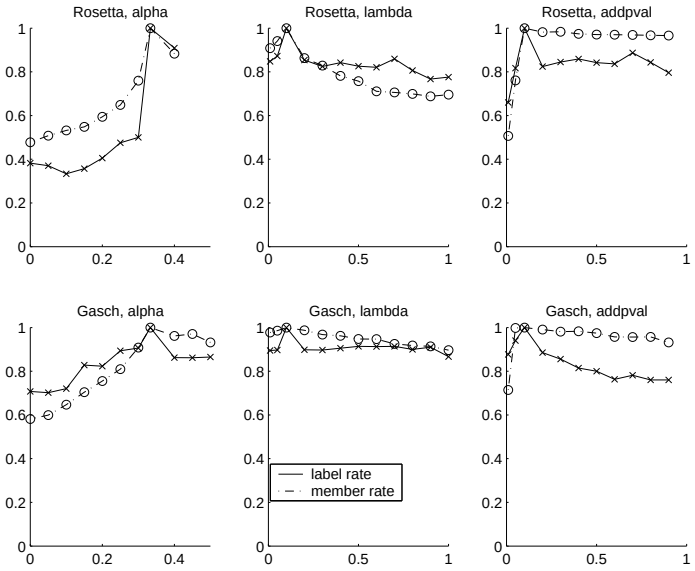
Table 6.14: Overlap of inferred models between Rosetta and Gasch data, Table 1

Regulator 1	Regulator 2	Regulator 3	SizeR	SizeG	NeceR	NeceG	SuffR	SuffG	Overlap rate1	Overlap rate2
Mcm1			5	7	1	1	0	1	20%	14.3%
Abf1			5	39	1	1	0	1	0%	0%
Fkh2			7	50	1	1	0	1	42.9%	6%
Reb1			27	37	1	1	0	1	48.1%	35.1%
Ash1	Ste12		10	16	10	00	01	11	20%	12.5%
Bas1			28	19	1	1	1	1	10.7%	15.8%
Sok2			54	44	0	1	1	1	5.6%	6.8%
Dig1	Ste12		57	46	11	00	11	11	31.6%	39.1%
Mac1			12	7	0	1	1	0	0%	0%
Phd1	Ste12		13	11	00	00	11	11	30.8%	36.4%
Rap1			31	148	1	1	0	1	93.5%	19.6%
Rlm1			35	9	0	1	1	0	0%	0%
Rap1	Fhl1		28	121	10	00	00	11	100%	23.1%
Gcn4	Hsf1		14	13	10	01	10	10	64.3%	69.2%
Fkh1			12	9	1	1	0	1	16.7%	22.2%
Ste12			49	107	1	1	1	1	28.6%	13.1%
Hap4			24	40	0	1	1	1	95.8%	57.5%
Gcn4			125	12	1	0	1	1	6.4%	66.7%
Ino4			5	11	0	1	1	1	20%	9.09%
Gal4			16	56	0	1	1	1	43.8%	12.5%
Msn4	Hsf1		8	22	00	00	11	11	37.5%	13.6%
Uga3			26	11	0	0	1	1	15.4%	36.4%
Msn4			13	16	0	1	1	1	23.1%	18.8%
Ino2	Ino4		13	22	11	00	00	01	15.4% 9.09%	
Rgm1	Gat3		8	22	00	00	10	10	0%	0%
Gcn4	Abf1		11	6	10	11	10	01	0%	0%
Dig1	Ste12	Mcm1	19	25	110	010	010	011	57.9%	44.0%
Yap1	Gcn4		18	11	01	10	01	10	50%	81.8%
Phd1	Swi4		9	15	11	01	10	00	11.1%	6.67%
Swi5			19	29	1	1	1	1	42.1%	27.6%
Swi4			5	10	0	1	1	1	0%	0%
Swi6			5	36	1	1	0	1	60%	8.3%
Phd1	Sok2		52	18	00	00	00	10	21.1%	61.1%
Mbp1	Swi4	Swi6	9	15	001	001	000	001	44.4%	26.7%
Rfx1			10	9	0	1	1	1	10%	11.1%
Msn2	Yap1		13	25	00	01	11	01	69.2%	36.0%
Nrg1	Sok2		37	6	00	10	01	10	5.4%	33.3%
Yap1			10	10	0	1	1	1	10%	10%
Mbp1	Swi6		12	19	11	01	00	01	41.7%	26.3%
Yap5			7	62	1	1	1	1	0%	0%
Msn2	Msn4		55	31	00	01	00	01	36.4%	64.5%
Ste12	Ino4		5	6	10	10	00	10	80%	66.7%
Aro80			9	25	0	1	1	1	77.8%	28%
Ash1			48	22	1	1	0	1	18.8%	40.9%

Table 6.15: Overlap of inferred models between Rosetta and Gasch data, Table 2

Regulator 1	Regulator 2	Regulator 3	SizeR	SizeG	NeceR	NeceG	SuffR	SuffG	Overlap rate1	Overlap rate2
Msn2	Msn4	Gcn4	14	7	001	010	001	110	35.7%	71.4%
Dig1			66	23	1	1	1	1	18.2%	52.2%
Cin5	Yap6		6	17	00	01	01	01	16.7%	5.88%
Hap4	Hap2		7	11	00	00	11	10	100%	63.6%
Phd1	Msn4		15	24	00	01	11	11	46.7%	29.2%
Usv1			6	20	0	1	1	1	50%	15%
Stp2			6	5	0	1	1	1	0%	0%
Ste12	Mcm1		24	30	10	10	10	10	54.2%	43.3%
Dig1	Ste12	Msn2	8	15	010	010	000	011	75%	40%
Met4			16	16	1	1	1	1	31.25%	31.25%
Rim101			8	6	0	1	1	1	37.5%	50%
Gcn4	Leu3		8	5	10	01	10	01	50%	80%
Rap1	Gat3	Fhl1	20	61	100	000	000	101	100%	32.8%
Swi4	Swi6		10	22	11	01	00	01	50%	22.7%
Gcn4	Bas1		8	7	11	11	01	01	50%	57.1%
Dig1	Ste12	Swi4	14	15	010	010	010	010	28.6%	26.7%
Msn2	Msn4	Yap1	17	24	000	000	100	010	70.6%	50%
Rap1	Fhl1	Yap5	20	52	100	000	000	010	100%	38.5%
Mac1	Cup9	Yap5	5	10	101	010	100	000	0%	0%
Hir2			7	27	1	1	1	1	100%	25.9%
Msn4	Yap1		22	22	00	00	11	11	59.1%	59.1%
Gat3	Yap5		7	38	11	01	11	11	0%	0%
Mth1	Gcn4		8	9	01	10	01	10	0%	0%
Mac1	Yap5		5	18	11	00	10	01	0%	0%
Rap1	Gcn4		8	5	01	10	01	10	0%	0%
Rtg3			47	29	1	1	1	1	31.9%	51.7%
Ash1	Rlm1		14	9	00	10	01	00	42.9%	66.7%
Mcm1	Rlm1		6	7	00	10	01	10	50%	42.9%
Msn2	Hsf1		10	26	00	01	11	01	30%	11.5%
Dig1	Ste12	Rlm1	18	10	000	010	001	010	22.2%	40%
Ste12	Skn7	Sok2	5	7	100	100	101	110	60%	42.9%
Ash1	Cin5	Yap6	15	10	000	001	001	010	33.3%	50%
Ash1	Mss11		14	6	10	10	00	00	35.7%	83.3%
Phd1	Msn4	Yap1	7	8	000	000	101	000	57.1%	50%
Sok2	Swi4		7	8	00	11	11	00	0%	0%
Ste12	Sok2		11	12	10	10	10	00	45.5%	41.7%
Fkh1	Fkh2		12	24	11	00	00	00	75%	37.5%
Msn4	Rap1		7	10	00	01	11	01	0%	0%
Mcm1	Gcn4		6	8	01	01	01	00	33%	25%
Phd1	Msn2		22	25	00	00	11	00	36.4%	32%
Swi4	Gcn4		11	8	00	10	01	10	54.5%	75%
Gal4	Gat3		8	13	10	01	11	01	12.5%	7.69%
Ash1	Rap1		6	6	10	01	00	01	0%	0%
Msn4	Gcn4		20	10	01	10	01	10	35%	70%

Figure 6-4: Robustness tests on parameters



against the variations of several free parameters.

We consider the following three parameters. λ appearing in the joint likelihood function (equation 6.14) is the relative weight of importance between expression and binding data. Larger λ puts more weight on expression data. ϵ in Table 6.2 relates the the prediction of a regulatory program to the hidden states of expression changes. Smaller ϵ makes the regulatory program more deterministic. p^{stop} in the greedy algorithm specifies the stopping criterion of the p-values of adding genes (Section 6.4.2). The smaller p^{stop} is, the earlier the greedy algorithm stops incorporating genes, hence the smaller the regulated gene set is. The default setting of these parameters is $\lambda = 0.1, \epsilon = \frac{1}{3}, p^{stop} = 0.1$. We performed three robustness tests by varying each parameter while fixing the other two as the default values. Inferred models generated from the new parameter settings were compared to the default models in two aspects. First, we calculated the average overlap rate of regulated gene sets (with respect to the default models) over all models. Second, we counted the fraction of new models which had identical inferred directions of effectiveness to the default models. Figure 6-4 shows the sensitivity of parameters in Rosetta and Gasch models. Either sensitivity measure is very robust against each parameter in each dataset except ϵ

on Rosetta data. For example, when varying λ from 0.01 to 0.9, the average overlap rate of Gasch models ranges between 90% and 100% and more than 85% of inferred models agree on directions of effectiveness. In contrast, models inferred from Rosetta data are sensitive to ϵ: the average overlap rate drops to 50% when ϵ varies from $\frac{1}{3}$ to 0.1.

Chapter 7

Conclusion

We conclude this dissertation by discussing the contribution and limitations of the current models. We subsequently propose several improvement directions for future work.

7.1 Contribution and limitations of current models

The physical network model described in Chapters Two and Three is the foundation of all the works in the dissertation. In general, this model provides a systematic framework for integrating explicit hypothesis of gene regulation and multiple types of data. Hypotheses about the underlying processes and confidence of measurements are expressed as constraints on the variables in the model. By applying approximate inference algorithms, we are able to identify variable configurations consistent with the constraints. From a biological perspective, the inferred results have clear interpretations since the hypotheses about the underlying process are explicit. From a computational perspective, on the other hand, this approach avoids the difficult model selection problem and translates the computational problem into an inference problem. Specifically, in this dissertation we adopt the hypothesis that the regulatory effects of gene perturbation propagate along pathways of molecular interactions. By

applying the physical network modeling framework and the inference algorithms, we are able to find the properties of the physical network which are consistent with the pathway hypothesis and measurement data. Moreover, by applying various validation tests described in Chapter Four, we have shown that the models can accurately predict new knock-out effects and the inferred results are consistent with known biological processes.

Despite these advantages, the current physical network models are also limited in many ways. The constraint-based framework requires concrete hypotheses specifying relations between the variables or properties in the model. Often simple assumptions which can capture the complexity of a biological system are not available. It would be beneficial to extract novel information without strong assumptions about the underlying processes. While most statistical models in gene expression analysis can capture these novel information without specifying the biophysical process, the current physical network model requires more specific hypotheses pertaining to gene regulation. For example, a Bayesian network model might capture the feedback relation from a transcription factor to the kinases upstream of this factor by identifying their statistical dependencies. In contrast, unless the location data shows that the transcription factor binds to the promoters of these kinases, physical network models are not able to recover the feedback relation.

The current physical network model is built on a very simple pathway hypothesis and may be inadequate in many cases. For example, only 1/20 of the knock-out interactions are connected via short pathways of molecular interactions. While it is possible to extend the current model to incorporate refined assumptions and vast amount of knowledge about gene regulation, various technical issues may become prohibitive as the models become more complex. For instance, the computational cost will be very high if we want to model detailed processes at individual molecular level. Moreover, less data will be available as the model targets low level biophysical processes.

The experimental design work depicted in Chapter Five can be viewed as a natural extension to the physical network models. We automate the selection of experi-

ments according to inferred models for the purpose of reconstructing gene regulatory networks. Moreover, we have empirically demonstrated that automatically selected experiments confirmed several putative pathways predicted by the model. While a number of previous works exist in prioritizing new experiments or applying experimental design methods in various problems, we are among the first authors to apply experimental design to study gene regulation at genomic scale and demonstrate this approach empirically.

Since the experimental design method is tightly coupled with the physical network models, it has similar limitations as the physical network models. The use of the mutual information score depends on the information available about the system. As shown from the learning curve analysis (Figure 5-4), the performance of the mutual information score is not superior to other criteria when data are limited. This is an intrinsic limitation of using the expected reduction of model entropy as the criteria for selecting experiments. When the data from less than 4 knock-out experiments are available, we may prefer other criteria for selecting experiments. Another restriction of the current experimental design method is the goal of discriminating candidate models. In most studies in computational biology, the purpose of doing wet experiments is to verify predictions generated from models. Although we have shown that the empirical outcomes of selected experiments also confirmed our predictions of putative pathways, the mutual information criterion is not designed for model verification. Moreover, the current method only considers single knock-out experiments. In practice, we can also select double deletion experiments or vary growth conditions of the cells among others.

The regulatory models described in Chapter Six relate gene expression levels. There are already a large number of studies dedicated to this topic; some of them are discussed in the literature review in Chapter One. The primary contribution of our models is a simplified way of characterizing the regulatory programs of multiple transcription factors. We characterize the regulatory programs in terms of the properties of single regulators – the functions of directions of effectiveness of single transcription factors. This simple characterization gives an intuitive interpretation of a complex

combinatorial function and reduces computational cost and over-fitting due to the restricted class of possible functions. Empirically, we found that about half of the inferred regulatory models were confirmed in previous studies. The results indicate that we are able to explain a large fraction of gene expression data with aggregate effects of single regulators.

The major limitation of this regulatory model is its simplicity. The model is built on several strong assumptions depicted in Section 6.1. In reality, these assumptions may hold only in special cases. For example, some transcription factors may not need to adjust their mRNA expressions in order to activate or inhibit genes; the necessary or sufficient property of a regulator may depend on the presence or absence of other regulators. Another limitation is the simple characterization and the discrete nature of the models. To fully specify the combinatorial effects of multiple regulators, efficient and large-scale assays that can capture many configurations of multiple regulators together seem to be required. For example, if we are able to perform all possible combinations of deletions and over-expressions of multiple regulators, then we can reconstruct the combinatorial function automatically from the data.

7.2 Future extensions

There are many possible directions to extend the current framework of physical network models. We close this chapter by discussing some of the primary extensions. These fall into two categories: overcoming the current limitations described earlier and broadening the types of data that the models can process.

- Incorporating regulatory models of multiple regulators within the physical network models

The physical network models discussed in Chapters Two to Four do not consider the combinatorial effects of multiple pathways. When multiple pathways connect to the deleted and affected genes, we require that any pathway suffices to explain the knock-out effect. This setting implies that each pathway is essential to regulate the

downstream gene. Redundant pathways, for example, are not modeled under this formulation. On the other hand, the regulatory models discussed in Chapter Six only consider the combinatorial effects between transcription factors and regulated genes. We want to extend the combinatorial effects to pathways in addition to protein-DNA bindings. A natural extension of combining the two models is to treat the physical network as a hypergraph: protein-DNA interactions incident to a gene (or a gene module) are taken as a hyperedge. Relevant functional data include multi-way interactions from single or double knock-out, over expression, or general expression data. Previously negative evidence (insignificant responses in knock-out experiments) were ignored, but negative evidence is important in the new scheme. For example, insignificant responses in single deletions are essential for inferring redundant pathways.

- Incorporating other types of data

Current physical network models are limited to clear causal effects on genes – namely pairwise knock-out interactions. Since most gene expression data are not under gene deletion or over expression conditions, it is important to extend the current model to incorporate general expression data. This requires generalizing causal explanation. In a general expression dataset, genes are changed under certain environmental conditions (for instance, heat shock). We have to link environmental changes to pathways of physical interactions, but there is a large degree of freedom for building those links due to the lack of direct evidence. Therefore, a major issue is to limit the assignment of pathways to environmental changes.

There are a vast amount of data capturing different aspects of cellular processes. Some examples include sequence data, mass spectrometry data of protein abundance and modification, information about protein complexes and localization, fluxes of metabolic reactions. In order to build a large-scale model of gene regulation, each type of data is important. As mentioned in Chapter Two, the advantages of data fusion are not only reducing errors from independent observations but also constraining the model space from complementary sources. In the long run, we plan to build models which can account for more types of data.

- Including spatial and temporal aspects in the model

Both physical network models and regulatory models of multiple regulators currently ignore the spatial and temporal dimensions of gene regulation. This is certainly unrealistic for they are essential elements of gene regulation. When extending to multi-cellular organisms, the spatio-temporal effects are the determining factors for development. Despite their importance, modeling spatio-temporal effects require much more fine-grained data and quantitative models. Many challenges exist in solving this problem. For example, the behavior of a complex, dynamic system depends on specific values of parameters. However, it is difficult to learn the exact values of these parameters from a small number of samples. In the setting of physical network models, an immediate extension to incorporate temporal aspects is to use dynamic data to restrict causal orders along pathways. In the long run, a new characterization of the model is required if exact values of parameters such as reaction rates are of interest.

- Generalizing experimental design

The current experimental design method focuses on a specific type of experiments (single knock-outs) on a specific task – to reduce uncertainty of model configurations. Experimental design can be generalized to other types of experiments and to fulfill other goals. For example, in addition to model discrimination we are also interested in choosing experiments to validate whether the true model is contained in a class of candidate models. Any type of perturbation experiments can be designed by the current framework as long as their effects can be predicted by the models. Over-expression experiments can be included as single knock-outs. Including double mutant experiments requires the model to cover combinatorial effects of pathways. Including external perturbations of environment requires modeling the links between environmental changes and molecular pathways.

- Incorporating physical mechanisms of combinatorial control

The combinatorial control scenarios depicted in Chapter Six cover only limited functional aspects. We want to endow the inferred functional relations with mechanistic basis like molecular cascades as the basis of knock-out effects. A possible direction of inferring the physical mechanisms is to categorize them into simplified "primitives". For example, two transcription factors can compete at a specific binding site, cooperatively bind at two sites, or form a complex to bind to a site. The recruitment of factors at promoters depends on the binding affinity of promoters, protein abundance and localization of factors, and the presence of other factors. We can arrange the settings of these primitives to fit the binding and expression data, like fitting the network configurations to explain physical and knock-out interaction data. However, the over-fitting problem is more serious since there are many combinations of these primitives and current datasets only probe very limited aspects of these mechanisms.

- Improving error models of data

To ensure the accuracy of inferred models, high quality datasets are essential. Except sequence data, data from high-throughput experiments is often questionable. A better characterization of their experimental errors is essential to make them useful. The current error models of CHIP-chip and knock-out data are not satisfactory for they are based on asymptotic statistics from large sample size. This contradicts with the fact that the reported data are drawn from few experiments. The error model of protein-protein interaction data is also unsatisfactory for the problematic hypothesis about EPR and PVM tests. Although error model improvement of individual datasets is independent of the modeling framework and can be dealt with separately, it still needs to be undertaken.

- Inferring missing links of molecular pathways

Due to the high false negative rate of physical interaction data, many effective pathways may not be present in the skeleton network constructed from existing datasets. Restricting inference to pathways of observed physical interactions may

greatly hinder the power of physical network models. We want to include the links which substantially empower the model to explain functional data even though the direct evidence pertaining to physical interactions are weak. For example, we can incrementally add links to the physical network which maximize the number of explained knock-out interactions.

Appendix A

Simplifying marginalization calculations

In this section we will discuss the methods of simplifying the marginalization calculations in equations 3.38 and 3.40:

$$m_{f\to x}(x) = \sum_{N(f)\backslash\{x\}} f(x, N(f)\backslash\{x\}) \prod_{x_i\in N(f)\backslash\{x\}} m_{x_i\to f}(x_i). \tag{A.1}$$

$$m_{f\to x}(x) = \max_{N(f)\backslash\{x\}} f(x, N(f)\backslash\{x\}) \prod_{x_i\in N(f)\backslash\{x\}} m_{x_i\to f}(x_i). \tag{A.2}$$

The main body of the max-product and sum-product algorithms comprises message updates. The update of messages from a variable to a factor (equation 3.37) is simply the product of single-variable functions. Computational bottlenecks occur at the update of messages from factors to variables (equations 3.38 and 3.40). Consider the maximization marginalization in equation 3.40 and assume all variables are binary. The maximization is carried out over all configurations in $N(f)\backslash\{x\}$ variables. We can enumerate all these $2^{|N(f)-1|}$ configurations and find the one which yields the maximum of $f(x, N(f)\backslash\{x\}) \prod_{x_i\in N(f)\backslash\{x\}} m_{x_i\to f}(x_i)$. This naive implementation requires the overall time complexity for each iteration to be $O(|V_f|2^{(\max|N(f)|-1)})$, where $|V_f|$ is the number of factor nodes and $\max|N(f)|$ is the largest size of factor arguments. A similar implementation can apply to the sum marginalization and

the time complexity is of the same order. Furthermore, if potential functions are lookup tables, then the algorithm also requires $O(|V_f|2^{(\max|N(f)|-1)})$ amount of space to store the potential functions. This naive implementation leads to inefficient use of resources. Consider our problem of model inference from large-scale datasets. When restricting the max path length to 3, there are 23771 potential function terms. The largest potential function can contain 10 variables (3 edge presence, 2 directions, 3 signs, 1 path selection, 1 knock-out effect). Hence in the worst case it requires about 1.2×10^7 units of computational time and storage space at each iteration.

This obstacle is insurmountable if potential terms are arbitrary functions. Fortunately, most potential functions in our model are relaxations of Boolean functions. We can apply basic logic deductions to simplify the evaluations of equations 3.38 and 3.40. Rather than exhausting all configurations, we only need to consider few configurations according to the deduction. Hence the running time is greatly reduced. The required space is also highly compressed because we only need to store the type of the potential function and its variable indices. The returned values under different configurations can be automatically deduced from simple rules.

As mentioned in Section 3.3, there are three types of potential functions in the physical network model: $\phi(.)$ pertaining to the confidence of physical interactions or knock-out effects, $\psi(.)$ for explaining knock-out effects via paths, and $\psi^{OR}(.)$ for enforcing the selection of at least one path to explain a knock-out effect. We discuss the simplification of marginalizations in different types of potential functions separately.

A.1 Potential functions of measurement confidence

$\phi(.)$ links a measurement with the underlying variable that the measurement is targeted to capture: $\phi_{e_i}(x_{e_i}; y_{e_i})$ for location data links the enrichment of protein-binding DNAs with the protein-DNA binding, or for protein-protein data links the categorizations of the reported protein pair with the actual protein-protein binding, and $\phi_{ij}(k_{ij}; \mathcal{E}_{ij})$ links knock-out expression data with the actual knock-out effect. These functions are depicted in equations 3.2, 3.18 and 3.19. The returned values of $\phi(.)$

under different configurations reflect the confidence about measurements or observations.

$\phi(.)$ is not a relaxation of a logical function, hence its message update equations 3.38 and 3.40 cannot be simplified. However, the factor-to-variable message $m_{f\to x}(x)$ is simply the potential function $f(x)$ itself for all $\phi(.)$s are functions of single variables. This holds for both max and sum marginalizations.

A.2 Potential functions of knock-out explanation

$\psi(.)$ links a knock-out effect with variables along a path connecting the cause and effect. The function is depicted in equation 3.28. It returns 1 when the knock-out effect is explained, ϵ_1 when the path is unselected, and ϵ_2 otherwise. It is a relaxed Boolean function, thus the marginalization can be simplified by applying logic deduction.

We first describe the simplification for max marginalization then for sum marginalization. Within each scheme we discuss the message updates of different types of variables separately.

A.2.1 Max marginalization

We first consider the message update from $\psi(.)$ to an edge presence variable x_1. From equation 3.40, the maximization marginalization equation is

$$\begin{aligned} &m_{\psi\to x_1}(x_1) = \max_{U_\psi\setminus\{x_1\}} \psi(x_1, x_2, \cdots, x_n, d_1, \cdots, d_n, s_1, \cdots, s_n, \sigma, k)\cdot \\ &\prod_{i=2}^{n} m_{x_i\to\psi}(x_i) \cdot \prod_{i=1}^{n} m_{d_i\to\psi}(d_i) \cdot \prod_{i=1}^{n} m_{s_i\to\psi}(s_i) \cdot m_{\sigma\to\psi}(\sigma) \cdot m_{k\to\psi}(k). \end{aligned} \tag{A.3}$$

where U_ψ denotes arguments in ψ, x_i, d_i, s_i are variables of edge presence, edge direction, edge sign respectively, σ is path selection variable and k is knock-out effect variable. For brevity we may write a message $m_{x\to\psi}(x)$ as $m(x)$. For $x_1 = 0$ and $x_1 = 1$, we want to find the configurations which maximize the term in equation A.3. Because of the property of ψ, we can summarize the candidate configurations as

follows.

- When $x_1 = 0$, the path cannot explain k. Thus the best scenario occurs at either $\sigma = 0$ (the path is not active) or $k = 0$ (the knock-out effect is insignificant). Under these settings, $\psi(.)$ no longer constrains other values. Instead, their best values are determined by their incident messages separately. For example, the best value of x_2 is $\arg\max_{x_2} m(x_2)$. Denote the best values of these variables as $\hat{y}$, then $m_{\psi \to x_1}(x_1 = 0) = \epsilon_1 \cdot \prod_{y \in U_\psi \setminus \{x_1, \sigma\}} m(y = \hat{y})$.

- When $x_1 = 1$, the path may or may not explain k. The optimal scenario is the supremum of the two cases. If the path explains k, then $\sigma = 1$, $k \neq 0$, all edge presence variables $x_i = 1$, and all edge direction variables follow the direction of k. We exhaust all configurations of edge signs and the knock-out effect which are consistent, and identify the one which yields the best value of $m(k) \cdot \prod_i m(s_i)$. The best scenario of case 1 is consequently obtained.

- When $x_1 = 1$ and the path does not explain k, then either $\sigma = 0$ or $k = 0$. The optimal values of the remaining variables are determined by their incident messages separately. The outcome is the best scenario of case 2. We choose the supremum of scenarios 1 and 2 to be $m_{\psi \to x_1}(x_1 = 1)$.

The message update for an edge direction variable d_1 is very similar to that of an edge presence variable. d_1 has two possible values $+1$ and -1, where one of them is consistent with the direction of the knock-out effect k. Without loss of generality we assume $d_1 = +1$ is consistent with k. Consequently, the message update simplification is exactly the same as the edge presence x_1 if we treat -1 in d_1 as 0 in x_1.

An edge sign variable s_1 also has two values $+1$ and -1. At $s_1 = -1$, we consider two cases identical to previous discussions.

1. In the first case, the path explains k. Hence $\sigma = 1$, $k \neq 0$, all x_i's are 1 and all d_i's are consistent with the knock-out effect. We have the freedom to change $s_2, \cdots, s_n$ under the constraint that the aggregate sign (together with $s_1 = -1$)

is $-k$. Hence we choose the subconfiguration of $s_2, \cdots, s_n$ which satisfies this condition and yields the best product of incident messages $\prod_{i=2}^{n} m(s_i)$.

2. In the second case, the path does not explain k. Hence $\sigma = 0$ or $k = 0$. The values of the remaining variables are determined by their incident messages separately.

The message $m_{\psi \to s_1}(s_1 = -1)$ is the supremum from these two scenarios. $m_{\psi \to s_1}(s_1 = +1)$ is obtained analogously.

A path selection variable σ has two possible values 0 and 1. When $\sigma = 0$, the path is not selected hence the constraint of explanation is removed. The best configuration of other variables can be obtained by finding their best values separately according to their incident messages. When $\sigma = 1$, the path may or may not explain the knock-out effect. $k = 0$ if the path does not explain k, and other variables are separately determined as before. If the path explains k, then we fix k, x_i and d_i according to explanation constraints and find the best edge sign configurations which are consistent with k. The best scenario of these two cases gives $m_{\psi \to \sigma}(\sigma = 1)$.

A knock-out variable k has three possible values $-1, 0, +1$. The message update procedure for $k = 0$ is exactly the same as $\sigma = 0$. When $k = +1$ or -1, we again consider the cases when the path is selected to explain k or not. If the path is selected, then all variables except edge signs are fixed, and we find the best edge sign configurations as before. If the path is not selected, then we find the best value of each variable separately according to its incident message.

A.2.2 Sum marginalization

The sum marginalization in equation 3.38 can also be simplified for path explanation potential functions. By expressing a potential function in terms of the configurations of variables, we can efficiently evaluate the sum marginals without enumerating all configurations.

We normalize the returned values of $\psi(.)$ under three conditions (the path explains the knock-out effect, the path is not selected, the path does not explain the knock-out

effect) to be v_1, v_2, v_3. The sum of the returned values over all configurations is 1. This function can be expressed in terms of the indicator function $I(.)$:

$$\psi(\text{config}) = (v_1 - v_3)I(\text{config explains } k) + (v_2 - v_3)[I(\sigma = 0) + I(k = 0) - I(\sigma = 0, k = 0)] + v_3. \tag{A.4}$$

The message update from equation 3.38 has the form of

$$m_{\psi \to x}(x) = \sum_{U_\psi \setminus \{x\}} \psi(U_\psi) \prod_{y \in U_\psi \setminus \{x\}} m(y). \tag{A.5}$$

If all potential functions are normalized and messages are normalized after update, then they can be treated as probability mass functions. The following equality holds for any set of variables U.

$$\sum_{val(U)} \prod_{y_i \in U} m(y_i = val(U)_i) = 1. \tag{A.6}$$

This is because $m(y)$'s can be treated as probability mass functions of independent random variables and the sum of the joint probabilities equals to 1. Equation A.4 greatly simplifies the sum marginal computation in equation 3.38. By substituting equation A.4 into the potential function in equation 3.38, the sum marginal of the second and the third terms is immediately obtained.

$$\begin{aligned} &\textstyle\sum_{U_\psi \setminus \{x\}} [(v_2 - v_3)(I(\sigma = 0) + I(k = 0) - I(\sigma = 0, k = 0)) + v_3] \cdot \prod_{y \in U_\psi \setminus \{x\}} m(y) \\ &= (v_2 - v_3)[m(\sigma = 0) + m(k = 0) - m(\sigma = 0)m(k = 0)] + v_3 \equiv Q_2 + v_3. \end{aligned} \tag{A.7}$$

Similar to previous discussions, the marginal of the first term in equation A.4 requires summing over edge sign configurations only. The constraint of explaining a knock-out effect forces $\sigma = 1$, all edge presence variables $= 1$, and all edge direction variables consistent with the knock-out effect. Thus,

$$I(\text{config explains } k) = I(\sigma = 1, x_1 = 1, \cdots, x_n = 1, d_1 = \hat{d}_1, \cdots, d_n = \hat{d}_n) \cdot I(k \neq 0, \prod_{i=1}^{n} s_i = -k). \tag{A.8}$$

Thus the sum marginal of first term in equation A.4 is

$$\begin{aligned}
&\textstyle\sum_{U_\psi \setminus \{x\}} (v_1 - v_3) I(\text{config explains } k) \cdot \prod_{y \in U_\psi \setminus \{x\}} m(y) = \\
&(v_1 - v_3) \cdot \textstyle\prod_{i=1}^{n} m(x_i = 1) \prod_{i=1}^{n} m(d_i = \hat{d}_i) m(\sigma = 1) \cdot \\
&(m(k = -1) \textstyle\sum_{\hat{s}_1 \cdot \hat{s}_n = +1} \prod_{i=1}^{n} m(s_i = \hat{s}_i) + m(k = +1) \sum_{\hat{s}_1 \cdot \hat{s}_n = -1} \prod_{i=1}^{n} m(s_i = \hat{s}_i)) \equiv Q_1.
\end{aligned} \tag{A.9}$$

Equations A.9 and A.7 can be applied to the message update of all types of variables. For edge presence variable x_1,

$$\begin{aligned}
m(x_1 = 0) &= Q_2 + v_3. \\
m(x_1 = 1) &= Q_1 + Q_2 + v_3.
\end{aligned} \tag{A.10}$$

For edge direction variable d_1 (assume $d_1 = +1$ is consistent with k),

$$\begin{aligned}
m(d_1 = -1) &= Q_2 + v_3. \\
m(d_1 = +1) &= Q_1 + Q_2 + v_3.
\end{aligned} \tag{A.11}$$

For edge sign variable s_1,

$$\begin{aligned}
&m(s_1 = -1) = (v_1 - v_3) \cdot \textstyle\prod_{i=1}^{n} m(x_i = 1) \prod_{i=1}^{n} m(d_i = \hat{d}_i) m(\sigma = 1) \cdot \\
&(m(k = -1) \textstyle\sum_{\hat{s}_2 \cdot \hat{s}_n = -1} \prod_{i=2}^{n} m(s_i = \hat{s}_i) + m(k = +1) \sum_{\hat{s}_2 \cdot \hat{s}_n = +1} \prod_{i=2}^{n} m(s_i = \hat{s}_i)) \\
&+ Q_2 + v_3. \\
&m(s_1 = +1) = (v_1 - v_3) \cdot \textstyle\prod_{i=1}^{n} m(x_i = 1) \prod_{i=1}^{n} m(d_i = \hat{d}_i) m(\sigma = 1) \cdot \\
&(m(k = -1) \textstyle\sum_{\hat{s}_2 \cdot \hat{s}_n = +1} \prod_{i=2}^{n} m(s_i = \hat{s}_i) + m(k = +1) \sum_{\hat{s}_2 \cdot \hat{s}_n = -1} \prod_{i=2}^{n} m(s_i = \hat{s}_i)) \\
&+ Q_2 + v_3.
\end{aligned} \tag{A.12}$$

For path selection variable σ,

$$\begin{aligned}
m(\sigma = 0) &= v_2. \\
m(\sigma = 1) &= Q_1 + (v_2 - v_3) m(k = 0) + v_3.
\end{aligned} \tag{A.13}$$

For knock-out effect variable k,

$$
\begin{aligned}
&m(k=0) = v_2. \\
&m(k=-1) = \textstyle\sum_{U_\psi \setminus \{x\}} (v_1 - v_3) I(\text{config explains } k) \cdot \prod_{y \in U_\psi \setminus \{x\}} m(y) \\
&= (v_1 - v_3) \cdot \textstyle\prod_{i=1}^{n} m(x_i = 1) \prod_{i=1}^{n} m(d_i = \hat{d}_i) m(\sigma = 1) \cdot \\
&\textstyle\sum_{\hat{s}_1 \cdot \hat{s}_n = +1} \prod_{i=1}^{n} m(s_i = \hat{s}_i) + (v_2 - v_3) m(\sigma = 0) + v_3. \\
&m(k=+1) = \textstyle\sum_{U_\psi \setminus \{x\}} (v_1 - v_3) I(\text{config explains } k) \cdot \prod_{y \in U_\psi \setminus \{x\}} m(y) \\
&= (v_1 - v_3) \cdot \textstyle\prod_{i=1}^{n} m(x_i = 1) \prod_{i=1}^{n} m(d_i = \hat{d}_i) m(\sigma = 1) \cdot \\
&\textstyle\sum_{\hat{s}_1 \cdot \hat{s}_n = -1} \prod_{i=1}^{n} m(s_i = \hat{s}_i) + (v_2 - v_3) m(\sigma = 0) + v_3.
\end{aligned}
\tag{A.14}
$$

A.3 Potential functions for noisy OR

$\psi^{OR}(.)$ is a noisy-OR function that returns a small value when all input arguments are 0 and a large value otherwise. Similar to $\psi(.)$, the marginalization of $\psi^{OR}(.)$ can also be simplified by applying deduction.

A.3.1 Max marginalization

The message update equation of the max-product algorithm is

$$
m_{\psi^{OR} \to \sigma_1}(\sigma_1) = \max_{\sigma_2, \cdots, \sigma_n} \psi^{OR}(\sigma_1, \cdots, \sigma_n) \prod_{i=2}^{n} m(\sigma_i). \tag{A.15}
$$

The max configuration in the case $\sigma_1 = 0$ is either (1)all other σ_i''s are separately determined from their incident messages or (2)one σ_i is fixed to 1 and all other σ_i''s are separately determined from their incident messages. We find the supremum configuration of (1) and (2) and compute the updated message. The max configuration in the case $\sigma_1 = 1$ is when all other σ_i's are separately determined from their incident messages. This is because the constraint encoded in ψ^{OR} is satisfied when $\sigma_1 = 1$.

A.3.2 Sum marginalization

The message update equation of the sum-product algorithm is

$$m_{\psi^{OR}\to\sigma_1}(\sigma_1) = \sum_{\sigma_2,\cdots,\sigma_n} \psi^{OR}(\sigma_1,\cdots,\sigma_n)\prod_{i=2}^{n} m(\sigma_i). \tag{A.16}$$

Similar to equation A.4, ψ^{OR} can be expressed as

$$\psi^{OR}(\sigma_1,\cdots,\sigma_n) = (v_3 - v_1)I(\sigma_1 = 0,\cdots,\sigma_n = 0) + v_1. \tag{A.17}$$

and the normalization equation A.6 holds. The message update becomes

$$\begin{array}{rcl} m_{\psi^{OR}\to\sigma_1}(\sigma_1 = 0) & = & (v_3 - v_1)\prod_{i=2}^{n} m(\sigma_i = 0) + v_1. \\ m_{\psi^{OR}\to\sigma_1}(\sigma_1 = 1) & = & v_1. \end{array} \tag{A.18}$$

A.4 Potential functions for model prediction

When evaluating the mutual information scores of new experiments, we have to predict the responses of single genes under a deletion perturbation. This is achieved by augmenting the factor graph model with potential functions pertaining to predictions and applying the sum-product algorithm. In this section we describe the simplification of evaluating the marginal belief functions of predicted responses. The max-marginalization simplification is not discussed since it is not used.

Recall the augmented potential function pertaining to the prediction along a single path is equation 5.38:

$$\phi_{ij}(Y_{ij}, X_{ij}, D_{ij}, S_{ij}) = \begin{cases} 1 & \text{if } (\forall x \in X_{ij}, x = 1) \cap (\forall d \in D_{ij}, d = 1) \cap (\prod_{s\in S_{ij}} s = -Y_{ij}), \\ 1 & \text{if } ((\exists x \in X_{ij}, x = 0) \cup (\exists d \in D_{ij}, d \neq 1)) \cap (Y_{ij} = 0), \\ 0 & \text{otherwise.} \end{cases} \tag{A.19}$$

X_{ij}, D_{ij}, S_{ij} are the variables of edge presence, edge direction and edge sign along the path π_{ij}. Y_{ij} is the predicted response of gene i along path j. We can evaluate the

marginal probability of Y_{ij} according to the belief functions of other variables.

$$P(Y_{ij}) = \sum_{X_{ij},D_{ij},S_{ij}} \phi_{ij}(Y_{ij}, X_{ij}, D_{ij}, S_{ij}) \cdot P(X_{ij}, D_{ij}, S_{ij}) \approx \sum_{X_{ij},D_{ij},S_{ij}} \phi_{ij}(Y_{ij}, X_{ij}, D_{ij}, S_{ij}) \cdot P(X_{ij})P(D_{ij})P(S_{ij}). \tag{A.20}$$

For simplicity we approximate the joint probability with the product of marginal beliefs of single variables. Substituting 5.38 into the equation, the approximated marginal probability of Y_{ij} then becomes

$$\begin{aligned} P(Y_{ij} = +1) &\approx \prod_{x \in X_{ij}} b(x = 1) \cdot \prod_{d \in D_{ij}} b(d = 1) \cdot P(\prod_{s \in S_{ij}} s = -1). \\ P(Y_{ij} = -1) &\approx \prod_{x \in X_{ij}} b(x = 1) \cdot \prod_{d \in D_{ij}} b(d = 1) \cdot P(\prod_{s \in S_{ij}} s = +1). \\ P(Y_{ij} = 0) &\approx 1 - \prod_{x \in X_{ij}} b(x = 1) \cdot \prod_{d \in D_{ij}} b(d = 1). \end{aligned} \tag{A.21}$$

These probabilities can be directly evaluated from the marginal beliefs of model variables.

To synthesize the predictions along all paths connecting to the same gene, we couple the predictions along single paths with the potential function equation 5.39:

$$\psi_i(Y_i, Y_{i1}, \cdots, Y_{iN}) = \begin{cases} 1 & \text{if } (Y_i = +1) \cap (\forall j, Y_{ij} \in \{0, +1\}) \cap ((Y_{i1}, \cdots, Y_{iN}) \neq (0, \cdots, 0)), \\ 1 & \text{if } (Y_i = -1) \cap (\forall j, Y_{ij} \in \{0, -1\}) \cap ((Y_{i1}, \cdots, Y_{iN}) \neq (0, \cdots, 0)), \\ 1 & \text{if } (Y_i = 0) \cap (\text{all other configurations}), \\ 0 & \text{otherwise.} \end{cases} \tag{A.22}$$

The marginal probability of Y_i is

$$P(Y_i) = \sum_{Y_{i1},\cdots,Y_{iN}} \psi_i(Y_i, Y_{i1}, \cdots, Y_{iN}) \cdot \prod_{j=1}^{N} P(Y_{ij}). \tag{A.23}$$

With some algebra,

$$\begin{aligned} P(Y_i = +1) &= \prod_{j=1}^{N} [P(Y_{ij} = +1) + P(Y_{ij} = 0)] - \prod_{j=1}^{N} P(Y_{ij} = 0), \\ P(Y_i = -1) &= \prod_{j=1}^{N} [P(Y_{ij} = -1) + P(Y_{ij} = 0)] - \prod_{j=1}^{N} P(Y_{ij} = 0), \\ P(Y_i = 0) &= 1 - P(Y_i = +1) - P(Y_i = -1). \end{aligned} \tag{A.24}$$

Appendix B

Addendum of empirical results

In this appendix, we include various empirical results which are not reported in the main text of the thesis. They include pairwise physical and knock-out interactions in the yeast pheromone response subnetwork (Chapter Four), the significance of expression change coherence among the genes bound by each transcription factor (Chapter Five), and the single factor function of each transcription factor (Chapter Six).

B.1 Pairwise interactions in the empirical analysis

We demonstrate the lists of physical interactions and knock-out interactions of the yeast mating pathway in Tables B.1 and B.2. The "type" columns in Table B.1 refer to the types of physical interactions: pd for protein-DNA interactions and pp for protein-protein interactions. For conciseness we do not show the physical and knock-out interactions of the entire physical network.

B.2 Significance of expression coherence

We introduce the method of measuring the expression coherence of a group of genes in Section 5.4. The p-value of expression coherence (equation 5.42) measures the significance of coherence against random sets of genes. However, the null model of random sets may be too weak to reflect the coherence significance. To make sure genes

Table B.1: Physical interactions

type	gene 1	gene 2	type	gene 1	gene 2	type	gene 1	gene 2
pd	STE12	SST2	pd	STE12	FAR1	pd	STE12	SCW10
pd	STE12	GPA1	pd	STE12	BAR1	pd	STE12	MFA2
pd	STE12	KAR4	pd	STE12	STE2	pd	STE12	YMR046C
pd	STE12	GIC2	pd	STE12	FIG1	pd	STE12	AGA2
pd	STE12	TEC1	pd	STE12	KAR5	pd	STE12	ASG7
pd	STE12	FUS1	pd	STE12	MFA1	pd	STE12	YNL279W
pd	STE12	PRY2	pd	STE12	STE6	pd	STE12	FUS3
pd	STE12	AGA1	pd	STE12	BEM2	pd	STE12	MSB2
pd	MCM1	FAR1	pd	MCM1	GPA1	pd	MCM1	BAR1
pd	MCM1	MFA2	pd	MCM1	STE6	pd	MCM1	STE2
pd	MCM1	AGA2	pd	MCM1	MFA1	pd	MCM1	AGA1
pp	FAR1	STE4	pp	FUS3	GPA1	pp	FUS3	STE11
pp	FUS3	STE5	pp	FUS3	STE7	pp	FUS3	YIL169C
pp	GIC2	STE50	pp	GPA1	SST2	pp	GPA1	STE11
pp	GPA1	STE4	pp	KSS1	SST2	pp	KSS1	STE11
pp	KSS1	STE12	pp	KSS1	STE5	pp	KSS1	STE7
pp	KSS1	TEC1	pp	MCM1	STE12	pp	SIN3	TUP1
pp	STE11	STE50	pp	STE11	STE5	pp	STE18	STE4
pp	STE4	STE5	pp	STE50	STE5	pp	STE5	STE7
pp	FUS3	STE12						

Table B.2: Knock-out interactions

deleted	affected	effect	deleted	affected	effect	deleted	affected	effect
STE4	STE2	-	STE4	GPA1	-	STE4	FUS3	-
STE4	MFA1	-	STE4	MFA2	-	STE4	STE6	-
STE4	FAR1	-	STE4	FUS1	-	STE4	AGA1	-
STE4	SST2	-	STE4	TEC1	-	STE4	KAR4	-
STE4	MSB2	-	STE4	GIC2	-	STE4	ASG7	-
STE4	SCW10	-	STE18	STE2	-	STE18	GPA1	-
STE18	FUS3	-	STE18	MFA1	-	STE18	MFA2	-
STE18	STE6	-	STE18	FAR1	-	STE18	AGA1	-
STE18	SST2	-	STE18	TEC1	-	STE18	KAR4	-
STE18	MSB2	-	STE18	GIC2	-	STE18	SCW10	-
FUS3	PRY2	+	STE7	STE2	-	STE7	GPA1	-
STE7	FUS3	-	STE7	STE6	-	STE7	FUS1	-
STE7	AGA1	-	STE7	AGA2	-	STE7	SST2	-
STE7	TEC1	-	STE7	KAR4	-	STE7	MSB2	-
STE7	PRY2	-	STE7	FIG1	-	STE7	GIC2	-
STE7	ASG7	-	STE7	YNL279W	-	STE11	STE2	-
STE11	GPA1	-	STE11	FUS3	-	STE11	STE6	-
STE11	FAR1	-	STE11	FUS1	-	STE11	AGA1	-
STE11	AGA2	-	STE11	SST2	-	STE11	TEC1	-
STE5	STE2	-	STE5	GPA1	-	STE5	FUS3	-
STE5	STE6	-	STE5	FAR1	-	STE5	FUS1	-
STE5	AGA1	-	STE5	AGA2	-	STE5	SST2	-
STE5	TEC1	-	STE5	KAR4	-	STE12	STE2	-
STE12	GPA1	-	STE12	FUS3	-	STE12	MFA1	-
STE12	MFA2	-	STE12	STE6	-	STE12	FAR1	-
STE12	FUS1	-	STE12	AGA1	-	STE12	AGA2	-
STE12	SST2	-	STE12	TEC1	-	STE12	KAR4	-
STE12	MSB2	-	STE12	PRY2	-	STE12	FIG1	-
STE12	GIC2	-	STE12	BEM2	-	STE12	ASG7	-
STE12	KAR5	-	STE12	SCW10	-	STE12	YNL279W	-
KSS1	FUS1	-	KSS1	FIG1	-	KSS1	ASG7	-
SST2	FUS3	+	SST2	FUS1	+	SST2	AGA1	+
SST2	AGA2	+	SST2	KAR4	+	SST2	FIG1	+
SST2	ASG7	+	SST2	KAR5	+	SST2	SCW10	+
SST2	YNL279W	+						

regulated by Msn4 or Hap4 undergo significant and coherence expression changes in each deletion experiment, we also compare their coherence p-values with sets of genes bound by each transcription factor. If the effects of deleting genes along a pathway are specific to this pathway rather than on a wide range of the network, then we shall see downstream genes of this pathway are more coherent than genes (putatively) regulated by other factors.

Tables B.3-B.7 rank the coherence p-values of genes putatively regulated by each factor in the five deletion experiments: Swi4Δ, Sok2Δ, Msn4Δ, Hap4Δ and Yap6Δ. Genes putatively regulated by Hap4 and Msn4 are selected according to literature review (see Section 5.4), and genes putatively regulated by other factors are based on location data (binding p-values ≤ 0.001). The semantics of the tables is as follows: the first column denotes factor name, the second column denotes the number of genes putatively regulated by the factor, the third column denotes the coherent direction of expression changes, and the fourth column denotes the coherence p-value. Factors are ranked according to the coherent p-values.

B.3 Single factor functions

We determine the single factor functions of transcription factors according to literature review and simple correlations. We first check the description about each transcription factor in the Yeast Proteome Database (YPD) [1]. If the factor is reported as either an activator or a repressor, we categorize it accordingly. Otherwise we impose a hard constraint that all protein-DNA edges emanating from the same transcription factor have the same sign. This is equivalent to the assumption that a factor has one consistent function throughout all its regulated genes. We then run the physical network model inference algorithm with these hard constraints and determine the signs (single factor functions) of the transcription factors. Table B.8 enlists the single factor functions of the 107 transcription factors.

[1] https://www.incyte.com/tools/proteome/databases.jsp

Table B.3: Coherence significance in Swi4Δ

factor	N	dir	pvalue	factor	N	dir	pvalue	factor	N	dir	pvalue
YAP5	141	+	< 1.00e-04	SWI4	175	-	< 1.00e-04	RGM1	72	+	< 1.00e-04
PDR1	100	+	< 1.00e-04	MCM1	141	+	< 1.00e-04	GCN4	115	-	< 1.00e-04
GAT3	139	+	< 1.00e-04	FHL1	192	-	< 1.00e-04	DIG1	54	+	< 1.00e-04
MSN4	42	+	5.00e-04	ARG81	36	-	5.00e-04	NRG1	100	-	1.00e-03
GAL4	54	+	1.10e-03	HIR2	46	-	1.80e-03	MAL13	32	+	2.50e-03
ARG80	49	-	3.00e-03	PUT3	20	-	3.40e-03	RPH1	15	+	5.30e-03
MOT3	44	+	5.40e-03	PHD1	137	+	7.20e-03	GRF10(Pho	70	-	7.20e-03
SKN7	158	-	8.30e-03	SWI5	132	+	1.13e-02	GAT1	18	-	1.13e-02
CIN5	219	-	1.20e-02	STE12	88	+	1.43e-02	YAP1	71	-	1.54e-02
RME1	45	+	1.69e-02	RGT1	19	+	2.01e-02	INO4	84	-	2.17e-02
RLM1	76	-	2.31e-02	ACE2	89	-	2.47e-02	DAL81	68	-	3.71e-02
LEU3	39	-	4.24e-02	IME4	52	+	4.33e-02	HAP4	29	-	4.44e-02
SFL1	32	+	4.54e-02	MIG1	36	+	6.11e-02	RAP1	353	-	6.20e-02
CBF1	91	-	6.92e-02	SFP1	65	-	7.09e-02	HIR1	66	-	7.49e-02
YAP7	4	+	7.52e-02	RCS1	65	+	7.54e-02	FKH2	158	-	8.28e-02
RTG3	47	+	8.43e-02	RTG1	55	+	8.94e-02	DOT6	71	+	9.02e-02
SUM1	86	+	9.06e-02	ROX1	76	-	9.37e-02	MAC1	74	+	1.00e-01
RTS2	19	+	1.04e-01	SRD1	44	+	1.07e-01	CHA4	40	+	1.18e-01
HAP2	34	+	1.21e-01	INO2	39	-	1.23e-01	AZF1	33	-	1.35e-01
CUP9	51	-	1.37e-01	IXR1	64	+	1.41e-01	STB1	53	-	1.53e-01
YAP6	120	-	1.54e-01	HSF1	100	-	1.60e-01	SWI6	145	+	1.69e-01
HAL9	31	-	1.71e-01	ECM22	1	-	1.76e-01	SIG1	43	-	1.86e-01
CRZ1	27	-	1.93e-01	MET31	43	+	2.05e-01	THI2	21	+	2.18e-01
BAS1	69	-	2.18e-01	GTS1	43	+	2.19e-01	CAD1	58	-	2.21e-01
MBP1	144	+	2.33e-01	NDD1	125	+	2.42e-01	ASH1	22	-	2.45e-01
SOK2	65	+	2.50e-01	MET4	56	-	2.65e-01	ZAP1	46	+	2.67e-01
SKO1	23	-	2.68e-01	MATa1	28	+	2.72e-01	PHO4	92	-	2.79e-01
RIM101	21	-	2.79e-01	GCR2	58	-	2.88e-01	MAL33	8	+	2.97e-01
STP1	40	+	2.99e-01	UGA3	34	+	3.15e-01	HAP5	35	+	3.15e-01
DAL82	47	-	3.23e-01	ABF1	441	-	3.36e-01	HAP3	62	+	3.37e-01
YAP3	7	+	3.65e-01	YFL044C	62	-	3.83e-01	GCR1	40	+	4.08e-01
STP2	37	+	4.09e-01	RFX1	55	-	4.09e-01	YJL206C	41	+	4.12e-01
FZF1	61	-	4.19e-01	MTH1	91	-	4.22e-01	SIP4	52	+	4.22e-01
MSS11	61	-	4.25e-01	MSN1	11	-	4.32e-01	SMP1	96	+	4.34e-01
GLN3	40	+	4.37e-01	ADR1	24	-	4.38e-01	USV1	42	-	4.39e-01
MSN2	13	+	4.40e-01	ZMS1	34	-	4.51e-01	HMS1	29	+	4.62e-01
ARO80	63	-	4.68e-01	REB1	235	+	4.90e-01	FKH1	129	-	4.92e-01
HAA1	2	-	5.29e-01	YBR267W	0	-	1.00e+00				

Table B.4: Coherence significance in Sok2Δ

factor	N	dir	pvalue	factor	N	dir	pvalue	factor	N	dir	pvalue
MSN4	42	+	< 1.00e-04	FHL1	192	+	< 1.00e-04	INO2	39	-	1.00e-04
INO4	84	-	2.00e-04	YAP6	120	+	1.50e-03	ARG81	36	-	2.60e-03
ARG80	49	-	3.80e-03	RAP1	353	+	4.10e-03	IXR1	64	+	7.80e-03
HAP4	29	+	1.07e-02	STE12	88	-	1.10e-02	REB1	235	-	1.10e-02
DAL81	68	-	1.64e-02	GCN4	115	-	1.82e-02	MET31	43	+	1.85e-02
SWI4	175	-	1.97e-02	NRG1	100	+	2.46e-02	FKH1	129	-	3.43e-02
YFL044C	62	+	3.47e-02	PHO4	92	-	3.67e-02	PDR1	100	+	5.26e-02
CUP9	51	+	5.70e-02	SUM1	86	-	5.72e-02	AZF1	33	+	5.75e-02
SOK2	65	+	5.78e-02	HSF1	100	+	6.02e-02	SFP1	65	+	6.34e-02
ECM22	1	-	6.50e-02	FKH2	158	-	6.74e-02	HAL9	31	+	7.35e-02
MBP1	144	-	7.39e-02	STB1	53	-	7.62e-02	GCR2	58	+	8.43e-02
HAP2	34	+	9.04e-02	MTH1	91	-	9.13e-02	ZAP1	46	-	9.17e-02
BAS1	69	+	9.39e-02	MCM1	141	-	9.56e-02	SWI6	145	-	1.13e-01
PHD1	137	+	1.17e-01	SKO1	23	+	1.47e-01	RGM1	72	-	1.57e-01
DIG1	54	-	1.58e-01	RIM101	21	-	1.59e-01	YAP1	71	-	1.65e-01
RTS2	19	-	1.88e-01	GRF10(Pho	70	-	1.93e-01	MSS11	61	-	2.02e-01
MSN1	11	-	2.05e-01	ARO80	63	+	2.16e-01	PUT3	20	-	2.20e-01
ROX1	76	+	2.22e-01	YAP3	7	-	2.40e-01	MET4	56	+	2.42e-01
RFX1	55	+	2.51e-01	GAL4	54	-	2.54e-01	YAP5	141	+	2.55e-01
STP2	37	+	2.63e-01	HIR2	46	-	2.74e-01	RLM1	76	+	2.76e-01
FZF1	61	+	2.77e-01	MAC1	74	+	2.79e-01	USV1	42	-	2.85e-01
SFL1	32	+	2.87e-01	RME1	45	-	2.87e-01	SIG1	43	-	2.87e-01
RCS1	65	+	2.88e-01	GAT1	18	-	2.92e-01	SMP1	96	+	2.95e-01
HAA1	2	+	2.99e-01	DAL82	47	+	3.00e-01	SRD1	44	+	3.04e-01
SIP4	52	-	3.07e-01	NDD1	125	-	3.08e-01	HAP3	62	-	3.10e-01
IME4	52	+	3.11e-01	SWI5	132	+	3.23e-01	CIN5	219	+	3.24e-01
GLN3	40	+	3.30e-01	GAT3	139	+	3.36e-01	STP1	40	-	3.38e-01
HAP5	35	+	3.63e-01	SKN7	158	+	3.83e-01	GCR1	40	-	3.87e-01
ADR1	24	+	4.07e-01	RTG3	47	-	4.08e-01	ABF1	441	+	4.10e-01
RPH1	15	-	4.15e-01	GTS1	43	-	4.22e-01	HMS1	29	-	4.23e-01
THI2	21	+	4.26e-01	RTG1	55	-	4.32e-01	DOT6	71	+	4.33e-01
MAL33	8	-	4.42e-01	ASH1	22	+	4.42e-01	ZMS1	34	-	4.44e-01
RGT1	19	+	4.45e-01	UGA3	34	-	4.49e-01	CHA4	40	-	4.54e-01
YJL206C	41	+	4.57e-01	MATa1	28	-	4.62e-01	LEU3	39	+	4.62e-01
MAL13	32	+	4.64e-01	MIG1	36	-	4.65e-01	CBF1	91	-	4.72e-01
YAP7	4	-	4.76e-01	ACE2	89	-	4.84e-01	CAD1	58	-	4.85e-01
CRZ1	27	-	4.91e-01	HIR1	66	+	4.95e-01	MOT3	44	-	4.96e-01
MSN2	13	-	4.96e-01	YBR267W	0	-	1.00e+00				

Table B.5: Coherence significance in Msn4Δ

factor	N	dir	pvalue	factor	N	dir	pvalue	factor	N	dir	pvalue
MSN4	42	-	< 1.00e-04	MBP1	144	+	< 1.00e-04	HSF1	100	-	< 1.00e-04
GCN4	115	-	< 1.00e-04	ARG81	36	-	< 1.00e-04	ARG80	49	-	1.00e-04
SWI6	145	+	2.00e-04	FHL1	192	+	3.00e-04	PUT3	20	-	6.00e-04
FKH2	158	+	1.00e-03	FKH1	129	+	1.00e-03	ABF1	441	+	1.50e-03
MTH1	91	-	2.90e-03	DAL81	68	-	3.00e-03	SKN7	158	-	3.70e-03
CHA4	40	+	4.10e-03	MAC1	74	+	1.22e-02	GTS1	43	+	1.25e-02
INO4	84	-	1.90e-02	HIR2	46	+	1.96e-02	GCR2	58	+	3.26e-02
NDD1	125	+	3.91e-02	SRD1	44	+	4.43e-02	CAD1	58	-	4.53e-02
CBF1	91	+	4.56e-02	CIN5	219	-	4.72e-02	HAP4	29	-	5.28e-02
RPH1	15	+	5.76e-02	RAP1	353	+	5.93e-02	RCS1	65	+	5.99e-02
SWI4	175	+	6.85e-02	HIR1	66	+	6.88e-02	RTG1	55	+	6.96e-02
GCR1	40	+	7.34e-02	BAS1	69	+	7.51e-02	USV1	42	+	7.80e-02
RTG3	47	+	7.95e-02	ZAP1	46	+	8.22e-02	YJL206C	41	-	9.23e-02
STP1	40	+	9.83e-02	DOT6	71	+	1.09e-01	ARO80	63	+	1.10e-01
YAP6	120	+	1.15e-01	CRZ1	27	-	1.20e-01	IXR1	64	+	1.34e-01
RTS2	19	+	1.35e-01	UGA3	34	+	1.36e-01	SUM1	86	-	1.42e-01
MIG1	36	+	1.43e-01	ECM22	1	-	1.56e-01	IME4	52	+	1.60e-01
SWI5	132	-	1.60e-01	ADR1	24	+	1.64e-01	SIP4	52	+	1.66e-01
GAL4	54	-	1.72e-01	INO2	39	-	1.81e-01	HAA1	2	+	1.92e-01
SMP1	96	+	2.07e-01	NRG1	100	-	2.11e-01	SIG1	43	+	2.12e-01
DIG1	54	+	2.16e-01	YFL044C	62	-	2.16e-01	GAT3	139	+	2.49e-01
GLN3	40	+	2.71e-01	PHD1	137	+	2.75e-01	RGM1	72	-	2.77e-01
GAT1	18	-	2.85e-01	RLM1	76	+	2.94e-01	ROX1	76	+	2.94e-01
STB1	53	+	3.02e-01	MSS11	61	+	3.04e-01	HAP3	62	-	3.07e-01
YAP1	71	-	3.12e-01	RME1	45	+	3.14e-01	RFX1	55	+	3.19e-01
MAL13	32	+	3.35e-01	ZMS1	34	-	3.37e-01	FZF1	61	-	3.41e-01
PDR1	100	+	3.44e-01	DAL82	47	-	3.49e-01	AZF1	33	-	3.52e-01
MSN2	13	+	3.60e-01	STP2	37	+	3.62e-01	MET31	43	+	3.64e-01
MCM1	141	-	3.68e-01	MET4	56	+	3.69e-01	LEU3	39	-	3.73e-01
YAP3	7	+	3.79e-01	SFL1	32	+	3.80e-01	RIM101	21	-	3.89e-01
SKO1	23	+	3.95e-01	CUP9	51	+	3.96e-01	ASH1	22	+	4.10e-01
MSN1	11	-	4.12e-01	MAL33	8	-	4.12e-01	HAP5	35	+	4.13e-01
THI2	21	+	4.17e-01	HAP2	34	+	4.22e-01	HAL9	31	-	4.25e-01
YAP5	141	+	4.27e-01	SOK2	65	-	4.39e-01	HMS1	29	+	4.40e-01
YAP7	4	+	4.49e-01	MOT3	44	-	4.51e-01	PHO4	92	+	4.52e-01
SFP1	65	-	4.53e-01	REB1	235	-	4.63e-01	STE12	88	-	4.65e-01
RGT1	19	-	4.65e-01	GRF10(Pho	70	+	4.71e-01	MATa1	28	-	4.75e-01
ACE2	89	+	4.97e-01	YBR267W	0	-	1.00e+00				

Table B.6: Coherence significance in Hap4Δ

factor	N	dir	pvalue	factor	N	dir	pvalue	factor	N	dir	pvalue
HAP4	29	-	< 1.00e-04	FHL1	192	+	< 1.00e-04	SWI6	145	+	1.00e-04
HAP2	34	-	5.00e-04	GAT1	18	-	1.10e-03	BAS1	69	+	1.20e-03
HSF1	100	-	2.90e-03	RGM1	72	-	3.60e-03	MET4	56	+	5.40e-03
FKH2	158	+	1.10e-02	GCR2	58	+	1.16e-02	MAC1	74	+	1.22e-02
CIN5	219	-	1.75e-02	CHA4	40	+	1.83e-02	ZAP1	46	+	1.90e-02
GCR1	40	+	2.19e-02	PHD1	137	-	2.39e-02	ABF1	441	+	2.44e-02
FKH1	129	+	2.59e-02	SWI5	132	-	3.33e-02	RCS1	65	+	3.54e-02
RFX1	55	+	3.56e-02	MBP1	144	+	3.61e-02	MET31	43	+	3.62e-02
RAP1	353	+	4.93e-02	MOT3	44	+	4.98e-02	HAP3	62	-	5.38e-02
ARG81	36	+	5.43e-02	STP1	40	+	6.19e-02	GAT3	139	-	7.09e-02
HIR1	66	+	7.80e-02	INO4	84	-	7.97e-02	GAL4	54	-	8.37e-02
SKN7	158	-	8.51e-02	DOT6	71	+	8.55e-02	HIR2	46	-	8.63e-02
PUT3	20	-	9.43e-02	YAP1	71	-	1.00e-01	GRF10(Pho	70	+	1.02e-01
HAP5	35	-	1.04e-01	GLN3	40	+	1.04e-01	RTG1	55	+	1.12e-01
DIG1	54	+	1.20e-01	HAL9	31	+	1.42e-01	SWI4	175	+	1.44e-01
CBF1	91	-	1.46e-01	SRD1	44	+	1.48e-01	RME1	45	+	1.50e-01
RPH1	15	+	1.55e-01	REB1	235	-	1.66e-01	DAL81	68	-	1.67e-01
SIG1	43	+	1.68e-01	AZF1	33	+	1.73e-01	HAA1	2	+	1.84e-01
IXR1	64	+	1.90e-01	MAL13	32	-	1.93e-01	MSN4	42	-	1.98e-01
RIM101	21	-	2.04e-01	STB1	53	+	2.05e-01	MSN1	11	+	2.10e-01
MATa1	28	+	2.20e-01	YJL206C	41	+	2.21e-01	YAP5	141	-	2.21e-01
ADR1	24	+	2.26e-01	ARO80	63	-	2.46e-01	SOK2	65	-	2.56e-01
CUP9	51	-	2.57e-01	INO2	39	-	2.61e-01	RTS2	19	+	2.65e-01
CAD1	58	-	2.73e-01	NDD1	125	+	2.82e-01	ROX1	76	+	2.87e-01
SKO1	23	+	2.89e-01	MTH1	91	-	2.92e-01	SFL1	32	-	2.98e-01
YAP3	7	+	3.01e-01	MIG1	36	+	3.06e-01	SUM1	86	+	3.09e-01
NRG1	100	-	3.11e-01	ZMS1	34	-	3.14e-01	PHO4	92	+	3.19e-01
MSS11	61	+	3.30e-01	RTG3	47	-	3.36e-01	DAL82	47	-	3.41e-01
ACE2	89	+	3.48e-01	FZF1	61	-	3.53e-01	MCM1	141	+	3.54e-01
CRZ1	27	-	3.57e-01	MSN2	13	-	3.66e-01	MAL33	8	-	3.79e-01
USV1	42	+	3.79e-01	SIP4	52	+	3.82e-01	ARG80	49	+	3.86e-01
RGT1	19	+	4.00e-01	STP2	37	+	4.02e-01	LEU3	39	-	4.08e-01
GTS1	43	+	4.11e-01	SFP1	65	-	4.15e-01	IME4	52	+	4.25e-01
RLM1	76	-	4.27e-01	YAP7	4	-	4.38e-01	THI2	21	+	4.44e-01
GCN4	115	+	4.44e-01	YAP6	120	-	4.66e-01	SMP1	96	-	4.74e-01
UGA3	34	-	4.80e-01	HMS1	29	+	4.80e-01	ASH1	22	+	4.80e-01
STE12	88	+	4.84e-01	PDR1	100	+	4.85e-01	YFL044C	62	-	4.89e-01
ECM22	1	+	7.26e-01	YBR267W	0	-	1.00e+00				

Table B.7: Coherence significance in Yap6Δ

factor	N	dir	pvalue	factor	N	dir	pvalue	factor	N	dir	pvalue
SWI4	175	+	< 1.00e-04	NDD1	125	+	< 1.00e-04	MSN4	42	+	< 1.00e-04
HAP4	29	+	< 1.00e-04	FKH2	158	+	< 1.00e-04	FHL1	192	-	< 1.00e-04
ABF1	441	-	< 1.00e-04	LEU3	39	-	1.00e-04	SWI5	132	+	3.00e-04
MBP1	144	+	3.00e-04	ACE2	89	+	5.00e-04	SKN7	158	+	7.00e-04
GCN4	115	-	1.50e-03	FKH1	129	+	1.90e-03	NRG1	100	+	4.80e-03
PHD1	137	+	5.40e-03	MAL13	32	+	1.77e-02	RAP1	353	-	2.22e-02
HIR2	46	+	2.32e-02	ARG81	36	-	2.85e-02	AZF1	33	-	3.37e-02
DAL81	68	-	3.43e-02	STB1	53	+	3.52e-02	MSS11	61	-	3.60e-02
SRD1	44	-	3.68e-02	MOT3	44	-	4.05e-02	ZAP1	46	+	4.63e-02
MAC1	74	+	5.04e-02	SIG1	43	-	5.16e-02	ARG80	49	-	5.47e-02
UGA3	34	-	5.59e-02	RTG1	55	+	5.86e-02	RGT1	19	-	6.51e-02
MET31	43	-	6.59e-02	SOK2	65	+	7.58e-02	SMP1	96	+	7.59e-02
YAP3	7	-	7.87e-02	RFX1	55	-	7.90e-02	MIG1	36	+	8.52e-02
PUT3	20	+	9.20e-02	IME4	52	+	1.01e-01	BAS1	69	-	1.06e-01
YAP6	120	+	1.14e-01	HIR1	66	+	1.34e-01	MSN1	11	+	1.39e-01
SWI6	145	+	1.74e-01	GAT1	18	+	1.79e-01	SIP4	52	-	1.82e-01
FZF1	61	-	1.88e-01	CUP9	51	+	1.97e-01	GLN3	40	-	1.97e-01
DAL82	47	+	1.98e-01	YFL044C	62	-	2.10e-01	MATa1	28	-	2.11e-01
GTS1	43	+	2.14e-01	RPH1	15	-	2.16e-01	MET4	56	+	2.22e-01
MSN2	13	-	2.25e-01	GCR1	40	-	2.25e-01	DOT6	71	-	2.26e-01
STE12	88	+	2.33e-01	DIG1	54	-	2.34e-01	HAP2	34	-	2.40e-01
CRZ1	27	-	2.54e-01	YAP7	4	-	2.61e-01	USV1	42	-	2.62e-01
STP1	40	-	2.64e-01	RTS2	19	-	2.88e-01	HSF1	100	+	2.89e-01
GRF10(Pho	70	-	2.90e-01	ASH1	22	+	2.95e-01	HAP3	62	-	2.96e-01
RCS1	65	+	2.97e-01	HAL9	31	-	3.00e-01	SUM1	86	-	3.04e-01
RME1	45	-	3.05e-01	PDR1	100	+	3.06e-01	PHO4	92	+	3.06e-01
RLM1	76	-	3.14e-01	YJL206C	41	+	3.16e-01	SFP1	65	+	3.16e-01
ZMS1	34	+	3.18e-01	INO4	84	-	3.23e-01	RTG3	47	-	3.44e-01
SKO1	23	-	3.44e-01	CBF1	91	-	3.49e-01	ROX1	76	+	3.53e-01
REB1	235	-	3.54e-01	GCR2	58	-	3.58e-01	MAL33	8	-	3.64e-01
THI2	21	-	3.66e-01	HMS1	29	+	3.80e-01	RIM101	21	-	3.86e-01
CIN5	219	+	3.89e-01	MCM1	141	+	3.96e-01	ARO80	63	-	4.03e-01
YAP1	71	+	4.12e-01	STP2	37	-	4.14e-01	YAP5	141	+	4.20e-01
GAL4	54	+	4.22e-01	INO2	39	-	4.25e-01	HAP5	35	-	4.43e-01
IXR1	64	+	4.43e-01	SFL1	32	-	4.54e-01	GAT3	139	+	4.59e-01
MTH1	91	-	4.62e-01	CAD1	58	+	4.72e-01	ADR1	24	-	4.76e-01
RGM1	72	+	4.88e-01	CHA4	40	-	5.00e-01	HAA1	2	+	5.64e-01
ECM22	1	+	6.87e-01	YBR267W	0	-	1.00e+00				

Table B.8: Single factor functions

factor	function	factor	function	factor	function	factor	function
ASH1	-	AZF1	+	CBF1	+	CHA4	+
DIG1	-	PHD1	+	STE12	+	CIN5	-
DAL82	-	ECM22	+	FZF1	+	GAT1	+
GCR2	+	GTS1	+	HMS1	+	INO2	+
IXR1	+	MAC1	+	MAL13	+	MAL33	-
MCM1	+	MET4	+	MIG1	+	MSN2	+
MSN4	+	MTH1	-	NRG1	-	PDR1	+
PUT3	-	RAP1	+	RFX1	+	RGM1	+
RGT1	-	RIM101	+	RME1	+	ROX1	-
RTG1	+	RTG3	+	RTS2	-	SFL1	-
SIG1	-	SKN7	+	SKO1	-	SMP1	+
SOK2	+	SRD1	+	STB1	+	THI2	+
UGA3	+	YAP1	+	YAP3	+	YAP6	+
YAP7	+	ZMS1	+	MSS11	+	INO4	+
PHO4	+	USV1	+	CUP9	-	GLN3	-
HAP3	+	REB1	+	ACE2	+	FKH1	+
MBP1	+	NDD1	+	SWI4	+	SWI5	+
SWI6	+	FKH2	+	GCN4	+	ARG80	-
ARG81	-	HAP4	+	HAP5	+	LEU3	+
MET31	+	CRZ1	+	GAL4	+	GAT3	+
GCR1	+	HIR2	+	RPH1	+	SUM1	+
MSN1	+	SIP4	+	HAA1	+	RLM1	+
ZAP1	+	ABF1	+	HSF1	+	IME4	+
MATa1	+	FHL1	+	SFP1	+	YFL044C	+
YJL206C	+	DOT6	+	RCS1	+	HIR1	+
ADR1	+	ARO80	+	BAS1	+	DAL81	+
HAL9	+	HAP2	+	STP1	+	STP2	-
CAD1	+	MOT3	+	YAP5	-		

Appendix C

Computational derivations regarding regulatory models

We describe four computational derivations in Chapter Six. First, we propose a simple error model of time-course gene expression measurements and demonstrate the calculations of conditional probabilities of measurements given the actual expression changes. Second, we show that the likelihood score of a regulatory model defined in equation 6.14 monotonically increases with the size of a module under certain conditions. The results suggest penalties on the size of regulated genes are required in order to construct regulatory models. Third, we propose a definition of the confidence values of incorporating genes in a regulatory model and demonstrate the analytic evaluation of the p-values. Fourth, we describe the permutation tests of evaluating the significance about the likelihood score of a regulatory model and the significance about the combinatorial property of a regulator.

C.1 Computing conditional probabilities from data

Computing the joint likelihood score in equation 6.14 requires us to know the conditional probabilities $P(x_{rg}|b_{rg}), P(x_{re}|c_{re})$ and $P(x_{ge}|c_{ge})$ a priori. These probabilities can be approximately transformed from the p-values or directly calculated if the error models of data are provided. We convert the p-values of location analysis data and

Rosetta gene expression data into conditional probabilities using the χ^2 approximation with Bayesian score. The conversion exactly follows the procedures described in Section 3.3.1.

Computing $P(x_{re}|c_{re})$ and $P(x_{ge}|c_{ge})$ from the gene expression data which do not provide p-values is more problematic. We use the stress response gene expression data ([61]) as the second expression dataset. It contains the time-course measurements of expression log ratios over 49 stress conditions. Because error models are not specified in the dataset, we use a simple parametric distribution to model the uncertainty of measurements. Let $y \in \{-1, 0, +1\}$ denotes the actual, quantized expression change of a gene under one experimental condition, and $x(t_1), \cdots, x(t_n)$ are its n time-course measurements. We relate discrete state y to measurements $x(t_1), \cdots, x(t_n)$ with a two-level process. The discrete state y generates a continuous time-course expression profile $m(t_1), \cdots, m(t_n)$; and $x(t_1), \cdots, x(t_n)$ are noisy measurements of $m(t_1), \cdots, m(t_n)$. We model measurement errors $x(t_1) - m(t_1), \cdots, x(t_n) - m(t_n)$ as iid Gaussian random variables with variance σ^2.

$$P(x(t)|m(t)) = \left(\frac{1}{2\pi\sigma^2}\right) e^{-\frac{(x(t)-m(t))^2}{2\sigma^2}}. \tag{C.1}$$

The actual expression profile $m(t_1), \cdots, m(t_n)$ is a zero vector given $y = 0$. Thus $P(x(t_1), \cdots, x(t_n)|y = 0)$ is the product of normal densities:

$$P(x(t_1), \cdots, x(t_n)|y = 0) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \prod_{i=1}^{n} e^{-\frac{x(t_i)^2}{2\sigma^2}}. \tag{C.2}$$

To model $P(x(t_1), \cdots, x(t_n)|y = \pm 1)$ we have to specify prior probabilities $P(m(t_1), \cdots, m(t_n)|y = \pm 1)$. We model the prior probabilities with an iid exponential distribution:

$$\begin{aligned} &P(m(t_1), \cdots, m(t_n)|y = +1) = \textstyle\prod_{t_i=1}^{n} P(m(t_i)|y = +1). \\ &P(m(t_i)|y = +1) = \begin{cases} \gamma e^{-\gamma m(t_i)} & \text{if } m(t_i) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{C.3}$$

$P(m(t_1), \cdots, m(t_n)|y = +1)$ assigns a non-zero probability to each non-negative ex-

pression profile, and penalizes the expression profiles deviating from 0. $P(m(t_1), \cdots, m(t_n)|y = -1)$ is defined analogously. By marginalizing over $m(t_i)$, the conditional probability $P(x(t_1), \cdots, x(t_n)|y = +1)$ becomes

$$\begin{aligned} &P(x(t_1), \cdots, x(t_n)|y = +1) = \prod_{i=1}^{n} \int_0^{\infty} P(m(t_i)|y = +1) P(x(t_i)|m(t_i)) dm(t_i) \\ &= \prod_{i=1}^{n} \gamma e^{(-\gamma x(t_i) + \frac{1}{2}\gamma^2\sigma^2)} (1 - \Phi(\frac{-(x(t_i) - \gamma\sigma^2)}{\sigma})). \end{aligned} \tag{C.4}$$

where $\Phi(.)$ is the standard normal cumulative distribution function. Similarly,

$$\begin{aligned} &P(x(t_1), \cdots, x(t_n)|y = -1) = \prod_{i=1}^{n} \int_{-\infty}^{0} P(m(t_i)|y = -1) P(x(t_i)|m(t_i)) dm(t_i) \\ &= \prod_{i=1}^{n} \gamma e^{(\gamma x(t_i) + \frac{1}{2}\gamma^2\sigma^2)} (\Phi(\frac{-(x(t_i) + \gamma\sigma^2)}{\sigma})). \end{aligned} \tag{C.5}$$

σ and γ are free parameters. In the empirical analysis we set $\sigma = \gamma = 0.5$ for they are close to the variance of the entire Gasch data.

C.2 Monotonicity property of fitness scores

Theorem Suppose the following conditions hold for the binding and expression probabilities on a regulatory model:

1. The probabilities of binding ($P(x_{rg}|b_{rg} = 1) \equiv p_{rg}$ in Section 6.3) of each regulator-gene pair satisfy $p_{rg} > 2^{-|R|}$, where $|R|$ is the number of regulators in the model.

2. The probability of expression changes ($p_{ge} = P(x_{ge}|c_{ge} = +1), q_{ge} = P(x_{ge}|c_{ge} = -1), u_{ge} = P(x_{ge}|c_{ge} = 0)$) for each gene g in each experiment e satisfies either $(2p_{ge} + u_{ge}) > 1$ or $(2q_{ge} + u_{ge}) > 1$.

3. The conditional probabilities of binding and expression are normalized. In other words, $p_{rg} + q_{rg} + u_{rg} = p_{ge} + q_{ge} + u_{ge} = 1$.

4. For each gene g under each experiment e, either $p_{ge}+u_{ge} \gg q_{ge}$ or $q_{ge}+u_{ge} \gg p_{ge}$.

Then the likelihood score defined in equation 6.14 monotonically increases with the number of regulated genes in the model.

Proof We will show both L^b and L^e increase with the size of the regulated gene set. According to equation 6.5, the change of L^b by adding a regulated gene g is

$$\Delta L^b = |R| \log 2 + \log p_{rg} - \log(p_{rg} + q_{rg}). \tag{C.6}$$

Since $p_{rg} > 2^{-|R|}$ according to condition 1 and $p_{rg} + q_{rg} = 1$ according to condition 3, $\Delta L^b > 0$.

Similarly, according to equation 6.13, the change of L^e on a specific experiment e by adding a regulated gene is

$$\begin{aligned}
&\Delta L^e(e) = \log 3 + \log[a \cdot (\tfrac{2}{3}p_{ge} + \tfrac{1}{3}u_{ge})P_1(e) + b \cdot (\tfrac{2}{3}q_{ge} + \tfrac{1}{3}u_{ge})P_{-1}(e) + \\
&c \cdot (\tfrac{1}{3}p_{ge} + \tfrac{1}{3}q_{ge} + \tfrac{1}{3}u_{ge})P_0(e)] \\
&- \log[aP_1(e) + bP_{-1}(e) + cP_0(e)].
\end{aligned} \tag{C.7}$$

where $a = \prod_{g'}(\frac{2}{3}p_{g'e} + \frac{1}{3}u_{g'e})$, $b = \prod_{g'}(\frac{2}{3}q_{g'e} + \frac{1}{3}u_{g'e})$, $c = \prod_{g'}(\frac{1}{3}p_{g'e} + \frac{1}{3}q_{g'e} + \frac{1}{3}u_{g'e})$. The definitions of $P_v(e)$s follow equation 6.13. By condition 4, either a or b is negligible. Without loss of generality assume $b \ll 1$. Substituting it into equation C.7,

$$\Delta L^e(e) \approx -\log[aP_1(e) + cP_0(e)] + \log[a \cdot (2p_{ge} + u_{ge}) \cdot P_1(e) + cP_0(e)] > 0. \tag{C.8}$$

The inequality arises from condition 2. Therefore, $\Delta L > 0$ by adding a gene into the model. Q.E.D.

Conditions 1-4 are reasonable assumptions for binding and expression data. Condition 1 states that the candidate genes have decent conditional probabilities for binding. Conditions 2 and 4 assume that the probabilities of up and down regulations cannot be simultaneously non-negligible. Condition 3 can be achieved by normalizing the conditional probabilities to sum to 1.

C.3 Confidence measures of incorporating a new gene into the model

The incremental algorithm of finding the regulated gene set stops when p-value of adding a new gene to the model is insignificant. In this section we describe the method of analytically computing the p-value of gene addition.

We consider only the p-value of the expression log likelihood ratio in equation 6.13 because each regulator-gene pair in the candidate set is already supported by the binding data. We establish the following scenario of generating random data for the likelihood score p-values. Conditional probabilities p_{re}, q_{re}, u_{re} of all regulators and p_{ge}, q_{ge}, u_{ge} of genes already in the regulated gene set are calculated from empirical data. Conditional probabilities p_{ge}, q_{ge}, u_{ge} of the newly incorporated gene in each experiment are uniformly sampled from the simplex $p_{ge}+q_{ge}+u_{ge} = 1, 0 \leq p_{ge}, q_{ge}, u_{ge} \leq 1$. This scenario is desirable because it only considers the fitness of the newly added gene. The simplex constraint is for mathematical convenience and can be achieved by normalizing the conditional probabilities.

Let G_0 be the current set of regulated genes and g be the newly added gene. The contribution of the expression likelihood score in equation 6.13 from each experiment is

$$\log(\sum_{v=\{-1,0,+1\}} P_v(e) \cdot [\prod_{g_0 \in G_0} \sum_{c_{g_0 e}} P(c_{g_0 e}|v) P(x_{g_0 e}|c_{g_0 e})] \cdot \sum_{c_{ge}} P(c_{g_e}|v) P(x_{g_e}|c_{g_e})). \quad \text{(C.9)}$$

The term from H_0 vanishes due to the simplex constraint. This quantity depends on the score of the current regulated gene set thus is less desirable. To simplify the calculation, we define a new test statistic for the newly added gene g on experiment e:

$$T_e = \log(\sum_{v=\{-1,0,+1\}} P_v(e) \cdot \sum_{c_{ge}} P(c_{g_e}|v) P(x_{g_e}|c_{g_e})). \quad \text{(C.10)}$$

By applying $P(c_{ge}|f((c_{re})))$ in Chapter Six,

$$\begin{aligned} T_e &= \log(P_1(e)(\tfrac{2}{3}p_{ge} + \tfrac{1}{3}u_{ge}) + P_{-1}(e)(\tfrac{2}{3}q_{ge} + \tfrac{1}{3}u_{ge}) + P_0(e)(\tfrac{1}{3}p_{ge} + \tfrac{1}{3}q_{ge} + \tfrac{1}{3}u_{ge})) \\ &\equiv \log(a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge}). \end{aligned} \tag{C.11}$$

The overall test statistic is

$$T = \sum_e T_e. \tag{C.12}$$

Let $\hat{T}$ be the empirical test statistic. The p-value of adding gene g to the model is

$$p = Pr(T \geq \hat{T}|(p_{ge}, q_{ge}, u_{ge}) \text{ sampled from simplex } p_{ge}+q_{ge}+u_{ge} = 1, 0 \leq p_{ge}, q_{ge}, u_{ge} \leq 1). \tag{C.13}$$

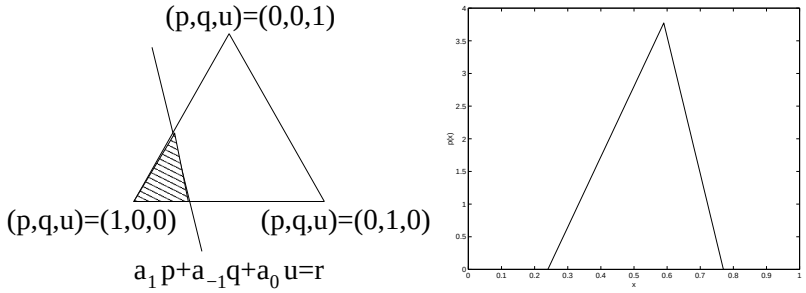
The distribution of each T_e can be analytically calculated. (p_{ge}, q_{ge}, u_{ge}) is uniformly sampled from the simplex $p_{ge} + q_{ge} + u_{ge} = 1, 0 \leq p_{ge}, q_{ge}, u_{ge} \leq 1$ embedded in a three-dimensional space. The test statistic $exp(T_e)$ is a linear function of (p_{ge}, q_{ge}, u_{ge}). Without loss of generality, assume $a_{1e} > a_{0e} > a_{-1e}$. T_e reaches the maximum when $(p_{ge}, q_{ge}, u_{ge}) = (1, 0, 0)$ and reaches the minimum when $(p_{ge}, q_{ge}, u_{ge}) = (0, 1, 0)$. The cumulative distribution of $a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge}$ is then the area of the intersection of the simplex triangle and the band between $a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge} = a_{-1e}$ and $a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge} = r$ as shown in the shaded region in Figure C-1.1. Using geometry,

$$Pr(a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge} \leq r) = \begin{cases} \frac{(r-a_{-1})^2}{(a_1-a_{-1})(a_0-a_{-1})} & a_{-1} \leq r \leq a_0, \\ 1 - \frac{(a-r)^2}{(a_1-a_{-1})(a_1-a_0)} & a_0 \leq r \leq a_1. \end{cases} \tag{C.14}$$

And the probability density function of $a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge}$ is the saw-tooth function as shown in Figure C-1.2.

$$Pr(r \leq a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge} \leq r + dr) = \begin{cases} \frac{2(r-a_{-1})dr}{(a_1-a_{-1})(a_0-a_{-1})} & a_{-1} \leq r \leq a_0, \\ \frac{2(a_1-r)dr}{(a_1-a_{-1})(a_1-a_0)} & a_0 \leq r \leq a_1. \end{cases} \tag{C.15}$$

Figure C-1: The restricted region within a simplex and its density function



The density of $\log(a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge})$ becomes

$$Pr(y \leq \log(a_{1e}p_{ge}+a_{-1e}q_{ge}+a_{0e}u_{ge}) \leq y+dy) = \begin{cases} \frac{2(e^y-a_{-1})e^y dy}{(a_1-a_{-1})(a_0-a_{-1})} & \log(a_{-1}) \leq y \leq \log(a_0), \\ \frac{2(a_1-e^t)e^y dy}{(a_1-a_{-1})(a_1-a_0)} & \log(a_0) \leq y \leq \log(a_1). \end{cases} \tag{C.16}$$

Denote $r_e \equiv a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge}$ and $y_e \equiv \log(a_{1e}p_{ge} + a_{-1e}q_{ge} + a_{0e}u_{ge})$. The p-value of the test statistic is

$$p = Pr(\sum_e y_e \geq \hat{T}). \tag{C.17}$$

The sum of y_e can be in principle calculated by convolution but has a complex distribution. We apply the central limit theorem to approximate $\sum_e y_e$ by a Gaussian distribution with mean $\mu = \sum_e E(y_e)$ and variance $\sigma^2 = \sum_e V(y_e)$. The p-value can be calculated by the Gaussian cumulative distribution function:

$$p = 1 - \Phi(\frac{\hat{T} - \mu}{\sigma}). \tag{C.18}$$

C.4 Fitting a regulatory model to expression data

To filter out the regulatory models which do not fit the data, we have to define the confidence measure of a regulatory model on data. Here we consider only gene expression data because the physical interactions in the model already pass a relatively stringent p-value threshold ($p \leq 0.005$).

We measure the fitness the expression data by permuting rows (genes in the regulated gene set) and columns (experiment indices) of the matrix of gene expression data restricted to the regulated gene set. We measure the fitness by normalizing the log likelihood score with respect to the means and variances of the scores from permuted data.

We rewrite the expression likelihood ratio function in equation 6.13 of Chapter Six:

$$\begin{aligned} & L^e(R,G,f) = \log P((x_{re}),(x_{ge})|H_1) - \log P((x_{re}),(x_{ge})|H_0) \\ & = -|E||R|\log 3 + \textstyle\sum_{e\in E}[\log(\sum_{v=\{-1,0,+1\}} P_v(e)\cdot\prod_{g\in G}\sum_{c_{ge}} P(c_{ge}|v)P(x_{ge}|c_{ge}))] \\ & +|E|(|R|+|G|)\log 3 + \textstyle\sum_{e\in E}[\sum_{r\in R}\log(p_{re}+q_{re}+u_{re}) + \sum_{g\in G}\log(p_{ge}+q_{ge}+u_{ge})]. \end{aligned} \tag{C.19}$$

We discard the contribution from H_0 since it is independent of data. We can calculate $L^e(R,G,f)$ according to empirical data and each permuted data. Denote $\bar{L}^e(R,G,f)$ and $V(L^e(R,G,f))$ the mean and variance of the expression log likelihood score evaluated over a population of permuted data. We then normalize $L^e(R,G,f)$ in terms of the mean and variance:

$$N(L^e(R,G,f)) = \frac{(L^e(R,G,f) - \bar{L}^e(R,G,f))}{\sqrt{V(L^e(R,G,f))}}. \tag{C.20}$$

We use the normalized score as the confidence measure about the fitness to expression data. In the empirical analysis in Chapter Six we report the models whose normalized scores ≥ 10.0.

C.5 Confidence measures of regulatory models

We apply permutation tests to evaluate the significance of the likelihood score of a regulatory model and the significance of the combinatorial property of a regulator. The significance of the likelihood score of a regulatory model is evaluated with the following procedures. We fix the members of regulated genes in the model and randomly permute their gene expression data. The binding data and gene expression

data of regulators remain unchanged. The optimal regulatory program is obtained for each permuted expression data. We compare the empirical likelihood score of the optimal model and the likelihood scores obtained from randomly permuted data. The p-value is the fraction of random trials which yield the likelihood scores $\geq$ the empirical value. In practice, almost all regulatory models are very significant (no random score exceeds the empirical score over 10000 trials). This result does not help us to choose regulatory models. To degrade the significance from the permutation tests, we introduce an offset value of likelihood scores proportional to the number of experiments times the size of the model (number of regulators + number of genes). The p-value is the fraction of random trials which yield the score $\geq$ empirical score - offset value.

The significance of the direction of effectiveness of each regulator is also calculated by permutation tests. Suppose we want to evaluate the significance whether regulator f is necessary. With members of the regulatory model fixed, we first identify the optimal combinatorial function with f as a necessary regulator and the optimal function with f as not necessary. The gap of the likelihood scores between the two models reflects the power of a combinatorial property to fit the data. We then permute the expression data of regulated genes, re-identify the two optimal functions, and calculate the likelihood gaps of the updated functions. The p-value of the statement that f is necessary is the fraction of random trials which yield the likelihood gap $\geq$ the empirical value.

Bibliography

[1] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, mar 2003.

[2] A. Alizadeh, M. Eisen, R. Davis andC. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown, and L. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, February 2000.

[3] R. Altman and D. Kellogg. Control of mitotic events by Nap1 and the Gin4 kinase. *Journal of Cell Biology*, 138(1):119–130, July 1997.

[4] J. Ambroziak and S. Henry. Ino2 and Ino4 gene products, positive regulators of phospholipid biosynthesis in *Saccharomyces Cerevisiae*, form a complex that binds to the ino1 promoter. *Journal of Biological Chemistry*, 269(21):15344–15349, May 1994.

[5] R. Aramayo, Y. Peleg, R. Addison, and R. Metzenberg. Asm-1+, a neurospora crassa gene related to transcriptional regulators of fungal development. *Genetics*, 144(3):991–1003, November 1996.

[6] A. Arkin, J. Ross, and H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage-lambda infected *Escherichia Coli* cells. *Genetics*, 149(4):1633–1648, 1998.

[7] K. Arndt, C. Styles, and G. Fink. Multiple global regulators control His4 transcription in yeast. *Science*, 237(4817):874–880, Aug 1987.

[8] G. Bader, I. Donaldson, C. Wolting, B. Ouellette, T. Pawson, and C. Hogue. BIND - biomolecular interaction network database. *Nulceic Acids Research*, 29(1):242–245, January 2001.

[9] K. Baetz, J. Moffat, J. Haynes, and M. Chang. Transcriptional coregulation by the cell integrity mitogen-activated protein kinase Slt2 and the cell cycle regulator Swi4. *Molecular and Cellular Biology*, 21(19):6515–6528, October 2001.

[10] T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning Journal*, 21:51–83, 1995.

[11] Z. Bar-Joseph, G. Gerber, D. Gifford, and T. Jaakkola. A new approach to analyzing gene expression time series data. In *RECOMB Proceedings*, pages 39–48, April 2002.

[12] Z. Bar-Joseph, G. Gerber, T. Lee, N. Rinaldi, J. Yoo, F. Robert, B. Gordon, E. Fraenkel, T. Jaakkola, R. Young, and D. Gifford. Computational discovery of gene modules and regulatory networks. *Nature biotechnology*, 21:1337–1342, November 2003.

[13] L. Bardwell, J. Cook, J. Zhu-Shimoni, D. Voora, and J. Thorner. Differential regulation of transcription: repression by unactivated mitogen-activated protein kinase Kss1 requires the Dig1 and Dig2 proteins. *PNAS*, 95(26):15400–15405, December 1998.

[14] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3):281–297, 1999.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *ICC Proceedings*, pages 1064–1070, 1993.

[16] P. Blaiseau and D. Thomas. Multiple transcriptional activation complexes tether the yeast activator Met4 to dna. *EMBO Journal*, 17(21):6327–6336, November 1998.

[17] E. Boy-Marcotte, M. Perrot, F. Bussereau, H. Boucherie, and M. Jacquet. Msn2p and Msn4p control a large number of genes induced at the diauxic transition which are repressed by cyclic amp in *Saccharomyces Cerevisiae*. *Journal of Bacteriology*, 180(5):1044–1052, March 1998.

[18] M. Campbell-Kelly and W. Aspray. *Computer: a history of the information machine*, Chapter One. Basic Books, 1996.

[19] S. Chandarlapaty and B. Errede. Ash1, a daughter cell-specific protein, is required for pseudohyphal growth of *Saccharomyces Cerevisiae*. *Molecular and Cellular Biology*, 18(5):2884–2891, May 1998.

[20] T. Chen, H. He, and G. Church. Modeling gene expression with differential equations. In *PSB Proceedings*, 1999.

[21] Y. Cheng and G. Church. Biclustering of expression data. In *ISMB Proceedings*, 2000.

[22] D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Corp., 1994.

[23] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, July 1998.

[24] B. Cohen, Y. Pilpel, R. Mitra, and G. Church. Discrimination between paralogs using microarray analysis: application to the Yap1p and Yap2p transcriptional networks. *Molecular Biology of the Cell*, 13(5):1608–1614, 2002.

[25] F. Collins, E. Green, A. Guttmacher, and M. Guyer. A vision for the future of genomics research. *Nature*, 422:835–847, April 2003.

[26] G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *UAI Proceedings*, pages 116–125, 1999.

[27] D. Cox and D. Hinkley. *Theoretical statistics*. Chapman and Hall, 1974.

[28] J. Crespo, T. Powers, B. Fowler, and M. Hall. The TOR-controlled transcription activators Gln3, Rtg1, and Rtg3 are regulated in response to intracellular levels of glutamine. *PNAS*, 99(10):6784–6789, May 2002.

[29] E. Davidson, J. Rast, P. Oliveri, A. Ransick, C. Calestani, C.H. Yuh, T. Minokawa, G. Amore, V. Hinman, C. Arenas-Mena, O. Otim, C. Brown, C. Livi, P. Lee, R. Revilla, A. Rust, Z. Pan, M. Schilstra, P. Clarke, M. Arnone, L. Rowen, R. Cameron, D. McClay, L. Hood, and H. Bolouri. A genomic regulatory network for development. *Science*, 295:1669–1678, March 2002.

[30] J. Davie, R. Trumbly, and S. Dent. Histone-dependent association of tup1-ssn6 with repressed genes in vivo. *Molecular and Cellular Biology*, 22(3):693–703, February 2002.

[31] C. Deane, L. Salwinski, I. Xenarios, and D. Eisenber. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Molecular Cell Proteomics*, 1(5):349–356, 2002.

[32] J. Deckert and K. Struhl. Histone acetylation at promoters is differentially affected by specific activators and repressors. *Molecular and Cellular Biology*, 21(8):2726–2735, April 2001.

[33] J. Deckert and K. Struhl. Targeted recruitment of Rpd3 histone deacetylase represses transcription by inhibiting recruitment of SWI/SNF, SAGA, and TATA binding protein. *Molecular and Cellular Biology*, 22(18):6458–6470, September 2002.

[34] V. Denis, H. Boucherie, C. Monribot, and B. Daignan-Fornier. Role of the Myb-like protein Bas1p in *Saccharomyces Cerevisiae*: a proteome analysis. *Molecular Microbiology*, 30(3):557–566, November 1998.

[35] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

[36] C. Devlin, K. Tice-Baldwin, D. Shore, and K. Arndt. Rap1 is required for Bas1/Bas2- and Gcn4-dependent transcription of the yeast His4 gene. *Molecular and Cellular Biology*, 11(7):3642–3651, July 1991.

[37] P. D'haeseleer. *Reconstructing gene networks from large scale gene expression data*. PhD thesis, University of New Mexico, 2000.

[38] B. Dibner. *Leonardo Da Vinci, machines and weaponry*. Norwalk: Burndy Library, 1974.

[39] L. Dirick, T. Bohm, and K. Nasmyth. Roles and regulation of Cln-Cdc28 kinases at the start of the cell cycle of *Saccharomyces Cerevisiae*. *EMBO Journal*, 14(19):4803–4813, October 1995.

[40] E. Dodou and R. Treisman. The *Saccharomyces Cerevisiae* MADS-box transcription factor Rlm1 is a target for the Mpk1 mitogen-activated protein kinase pathway. *Molecular and Cellular Biology*, 17(4):1848–1859, 1997.

[41] B. Drees, E. Grotkopp, and H. Nelson. The Gcn4 leucine zipper can functionally substitute for the heat shock transcription factor's trimerization domain. *Journal of Molecular Biology*, 273(1):61–74, 1997.

[42] R. Dror, J. Murnick, and N. Rinaldi. A Bayesian approach to transcript estimation from gene array data: the beam technique. In *RECOMB Proceedings*, pages 137–143, 2002.

[43] A. Dudley, L. Gansheroff, and F. Winston. Specific components of the SAGA complex are required for Gcn4- and Gcr1-mediated activation of the His4-

912delta promoter in *Saccharomyces Cerevisiae*. *Genetics*, 151(4):1365–1378, April 1999.

[44] B. Durbin, J. Hardin, D. Hawkins, and D. Rocke. A variance-stabilizing transformation for gene-expression microarray data. In *ISMB Proceedings*, 2002.

[45] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analyis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.

[46] E. Elion, B. Satterberg, and J. Kranz. Fus3 phosphorylates multiple components of the mating signal transduction cascade: evidence for Ste12 and Far1. *Molecular Biology of the Cell*, 4(5):495–510, May 1993.

[47] B. Errede. Mcm1 binds to a transcriptional control element in Ty1. *Molecular and Cellular Biology*, 13(1):57–62, January 1993.

[48] V.V. Fedorov. *Theory of optimal experiments*. Academic press, New York, 1972.

[49] L. Fernandes, C. Rodrigues-Pousada, and K. Struhl. Yap, a novel family of eight bzip proteins in *Saccharomyces Cerevisiae* with distinct biological functions. *Molecular and Cellular Biology*, 17(12):6982–6993, 1997.

[50] S. Fodor. Massively parallel genomics. *Science*, 277:393–395, July 1997.

[51] F. Foor, S. Parent, N. Morin, A. Dahl, N. Ramadan, G. Chrebet, K. Bostian, and J. Nielsen. Calcineurin mediates inhibition by FK506 and cyclosporin of recovery from alpha-factor arrest in yeast. *Nature*, 360(6405):682–684, December 1992.

[52] G. Fourel, T. Miyake, P. Defossez, R. Li, and E. Gilson. General regulatory factors (GRFs) as genome partitioners. *Journal of Biological Chemistry*, 277(44):41736–41743, Nov 2002.

[53] C. Francisco and W. Fuller. Quantile estimation with a complex survey design. *Annals of Statistics*, 19(1):454–469, 1991.

[54] Y. Freund, S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[55] B. Frey and D. MacKay. A revolution : belief propagation in graphs with cycles. In *NIPS Proceedings*, pages 479–485, 1997.

[56] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. In *RECOMB Proceedings*, pages 127–135, 2000.

[57] N. Friedman and I. Nachman. Gaussian process networks. In *UAI Proceedings*, 2000.

[58] T. Furuchi, H. Ishikawa, N. Miura, M. Ishizuka, K. Kajiya, S. Kuge, and A. Naganuma. Two nuclear proteins, Cin5 and YDR259C, confer resistance to cisplatin in *Saccharomyces Cerevisiae*. *Molecular Pharmacology*, 59(3):470–474, March 2001.

[59] J. Gancedo. Control of pseudohyphae formation in *Saccharomyces Cerevisiae*. *FEMS Microbiological Review*, 25(1):107–123, January 2001.

[60] T. Gardner, D. di Bernardo, D. Lorenz, and J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, July 2003.

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

[62] A.C. Gavin, M. Bosche, R. Krause, P. Grand, M. Marzioch, A. Bauer, J. Schultz, J. Rick, A. Michon, C. Cruciat, M. Remor, C. Hofert andM. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. Heurtier, P. Bork,

B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, January 2002.

[63] S. Ghaemmaghami, W. Huh, K. Bower, R. Howson, A. Belle, N. Dephoure, E. O'Shea, and J. Weissman. Global analysis of protein expression in yeast. *Nature*, 425:737–741, October 2003.

[64] G. Giaever, A. Chu, L. Ni, C. Connelly, L. Riles, S. Veronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. Andre, A. Arkin, A. Astromoff, M. Bakkoury, R. Bangham, R. Benito, S. Brachat, S. Campanaro, M. Curtis, K. Davis, A. Deutschbauer, K. Entian, P. Flaherty, F. Foury, D. Garfinkel, M. Gerstein, D. Gotte, U. Guldener, J. Hegemann, S. Hempel, Z. Herman, D. Jaramillo, D. Kelly, S. Kelly, P. Kotter, D. LaBonte, D. Lamb, N. Lan, H. Liang, H. Liao, L. Liu, C. Luo, M. Lussier, R. Mao, P. Menard, S. Ooi, J. Revuelta, C. Roberts, M. Rose, P. Ross-Macdonald, B. Scherens, G. Schimmack, B. Schafer, D. Shoemaker, S. Sookhail-Mahadeo, R. Storms, J. Strathern, G. Valle, M. Voet, G. Volckaert, C. Wang, T. Ward, J. Wihelmy, E. Winzeler, Y. Yang, G. Yen, E. Youngman, K. Yu, H. Bussey, J. Boeke, M. Snyder, P. Philippsen, R. Davis, and M. Johnston. Functional profiling of the *Saccharomyces Cerevisiae* genome. *Nature*, 418:387–391, July 2002.

[65] C. Gimeno and G. Fink. Induction of pseudohyphal growth by overexpression of Phd1, a *Saccharomyces Cerevisiae* gene related to transcriptional regulators of fungal development. *Molecular and Cellular Biology*, 14(3):2100–2112, 1994.

[66] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

[67] M. Greenberg and J. Lopes. Genetic regulation of phospholipid biosynthesis in *Saccharomyces Cerevisiae*. *Microbiological Review*, 60(1):1–20, March 1996.

[68] D. Griggs and M. Johnston. Regulated expression of Gal4 activator gene in yeast provides a sensitive genetic switch for glucose repression. *PNAS*, 88:8597–8601, October 1991.

[69] J. Hahn and D. Thiele. Regulation of the *Saccharomyces Cerevisiae* Slt2 kinase pathway by the stress-inducible Sdp1 dual specificity phosphatase. *Journal of Biological Chemistry*, 277(24):21278–21284, June 2002.

[70] A. Hartemink. *Principled computational methods for the validation and discovery of genetic regulatory networks*. PhD thesis, Massachusetts Institutes of Technology, 2001.

[71] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *PSB Proceedings*, 2001.

[72] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Combining location and expression data for principled discovery of genetic regulatory network models. In *PSB Proceedings*, 2002.

[73] D. Heckerman. A tutorial on learning with Bayesian networks. In *Learning in graphical models*. MIT Press, 1999.

[74] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Corp., 1994.

[75] Y. Ho, A. Gruhler, A. Heilbut, G. Bader, L. Moore, S. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskart, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A. Willems, H. Sassi, P. Nielsen, K. Rasmussen, J. Andersen, L. Johansen, L. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crwaford, V. Poulsen, B. Sorensen, J. Matthiesen, R. Hendrickson, F. Gleeson, T. Pawson, M. Moran, D. Durocher, M. Mann, C. Hougue,

D. Figeys, and M. Tyers. Systematic identification of protein complexes in *Saccharomyces Cerevisiae* by mass spectrometry. *Nature*, 415:180–183, January 2002.

[76] I. Holmes and W. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In *ISMB Proceedings*, 2000.

[77] F. Holstege, E. Jennings, J. Wyrick, T. Lee, C. Hengartner, M. Green, T. Golub, E. Lander, and R. Young. Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, 95:717–728, November 1998.

[78] C. Horak, N. Luscombe, J. Quian, P. Bertone, S. Piccirrillo, N. Gerstein, and M. Snyder. Complex transcriptional circuitry at the g1/s transition in *Saccharomyces Cerevisiae*. *Genes & Development*, 16:3017–3033, 2002.

[79] W. Huber, A. Heydebreck, H. Sultmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. In *ISMB Proceedings*, 2002.

[80] T. Hughes, M. Marton, A. Jones, C. Roberts, R. Stoughton, C. Armour, H. Bennett, E. Coffey, H. Dai, Y. He, M. Kidd, A. King, M. Meyer, D. Slade, P. Lum, S. Stepaniants, D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[81] W. Huh, J. Falvo, L. Gerke, A. Carroll, R. Howson, J. Weissman, and E. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425:686–691, October 2003.

[82] T. Ideker, O. Ozier, B. Schwikowski, and A. Siegel. Discoversing regulatory and signaling circuits in molecular interaction networks. In *ISMB Proceedings*, 2002.

[83] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, J. Buhler, J. Eng, R. Bumgarner, D. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic

analyses of a systematically perturbed metabolic network. *Science*, 292:929–934, May 2001.

[84] J. Ihmels, R. Levy, and N. Barkai. Principles of transcriptional control in the metabolic network of *Saccharomyces Cerevisiae*. *Nature Biotechnology*, 22(1):86–92, January 2004.

[85] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. In *PSB Proceedings*, pages 175–186, 2002.

[86] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8):4569–4574, 2001.

[87] T. Jaakkola and H. Siegelmann. Active information retrieval. In *NIPS Proceedings*, 2001.

[88] M. Jia, R. Larossa, J. Lee, A. Rafalski, E. Derose, G. Gonye, and Z. Xue. Global expression profiling of yeast treated with an inhibitor of amino acid biosynthesis, sulfometuron methyl. *Physiological Genomics*, 3(2):83–92, August 2000.

[89] J. Jordan, E. Landau, and R. Iyengar. Signaling networks: the origins of cellular multitasking. *Cell*, 103:193–200, October 2000.

[90] Q. Ju, B. Morrow, and J. Warner. Reb1, a yeast dna-binding protein with many targets, is essential for growth and bears some resemblance to the oncogene myb. *Molecular Cell Biology*, 10(10):5226–5234, Oct 1990.

[91] U. Jung and D. Levin. Genome-wide analysis of gene expression regulated by the yeast cell wall integrity signalling pathway. *Molecular Microbiology*, 34(5):1049–1057, December 1999.

[92] J. Jungmann, H. Reins, J. Lee, A. Romeo, R. Hassett, and D. Kosman. Mac1, a nuclear regulatory protein related to Cu-dependent transcription factors is

involved in Cu/Fe utilization and stress resistance in yeast. *EMBO Journal*, 12(13):5051–5056, December 1993.

[93] M. Kasten, S. Dorland, and D. Stillman. A large protein complex containing the yeast Sin3p and Rpd3p transcriptional regulators. *Molecular and Cellular Biology*, 17(8):4852–4858, August 1997.

[94] S. Kauffman. *The origins of order: self-organization and selection in evolution.* Oxford University Press, 1993.

[95] M. Kellis, N. Patterson, M. Endrizzi, and E. Lander B. Birren. Sequencing and comparison of yeast species to identify genes and regulatory motifs. *Nature*, 423:241–254, May 2003.

[96] R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell, and S. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, January 2004.

[97] F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[98] A. Kumar, S. Agarwal, J. Heyman, S. Matson S, M. Heidtman, S. Piccirillo, L. Umansky, A. Drawid, R. Jansen, Y. Liu, K. Cheung, P. Miller, M. Gerstein, G. Roeder, and M. Snyder. Subcellular localization of the yeast proteome. *Genes Development*, 16(6):707–719, March 2002.

[99] M. Kunzler, C. Springer, and G. Braus. Activation and repression of the yeast Aro3 gene by global transcription factors. *Molecular Microbiology*, 15(1):167–178, Jan 1995.

[100] T. Lee, N. Rinaldi, F. Robert, D. Odom, Z. Bar-Joseph, G. Gerber, N. Hannett, C. Harbison, C. Thimpson, I. Simon, J. Zeitlinger, E. Jennings, H. Murray, B. Gordon, B. Ren, J. Wyrick, J. Tagne, T. Volkert, E. Franekel, D. Gifford, and R. Young. A transcriptional regulatory network map for *Saccharomyces Cerevisiae*. *Science*, 298:799–804, 2002.

[101] B. Lewin. *Genes VI*. Oxford University Press, 1998.

[102] S. Li, A. Ault, C. Malone, D. Raitt, S. Dean, L. Johnston, R. Deschenes, and J. Fassler. The yeast histidine protein kinase, Sln1p, mediates phosphotransfer to two response regulators, Ssk1p and Skn7p. *EMBO Journal*, 17(23):6952–6962, Dec 1998.

[103] D. Lohr, P. Venkov, and J. Zlatanova. Transcriptional regulation in the yeast Gal gene family: a complex genetic network. *The FASEB Journal*, 9:777–787, June 1995.

[104] M. Lorenz and J. Heitman. Regulators of pseudohyphal differentiation in *Saccharomyces Cerevisiae* identified through multicopy suppressor analysis in ammonium permease mutant strains. *Genetics*, 150(4):1443–1457, December 1998.

[105] K. Luo, M. Vega-Palas, and M. Grunstein. Rap1-Sir4 binding independent of other Sir, yKu, or histone interactions initiates the assembly of telomeric heterochromatin in yeast. *Genes Development*, 16(12):1528–1539, June 2002.

[106] D. MacKay. Introduction to monte carlo methods. In *Learning in graphical models*. MIT Press, 1999.

[107] D. MacKay and R. Neal. Good error-correcting codes based on very sparse matrices. In *Cryptography and Coding*. LNCS, 1995.

[108] H. Madhani and G. Fink. The riddle of map kinase signaling specificity. *Trends in Genetics*, 14:151–155, 1998.

[109] D. McNabb, Y. Xing, and L. Guarente. Cloning of yeast Hap5: a novel subunit of a heterotrimeric complex required for CCAAT binding. *Genes Development*, 9(1):47–58, January 1995.

[110] F. Messenguy and E. Dubois. Genetic evidence for a role for Mcm1 in the regulation of arginine metabolism in *Saccharomyces Cerevisiae*. *Molecular and Cellular Biology*, 13(4):2586–2592, April 1993.

[111] F. Messenguy, E. Dubois, and C. Boonchird. Determination of the DNA-binding sequences of ARGR proteins to arginine anabolic and catabolic promoters. *Molecular and Cellular Biology*, 11(5):2852–2863, May 1991.

[112] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[113] H. Mountain, A. Bystrom, and C. Korch. The general amino acid control regulates met4, which encodes a methionine-pathway-specific transcriptional activator of *Saccharomyces Cerevisiae*. *Molecular Microbiology*, 9(1):221–223, July 1993.

[114] K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA., 1999.

[115] K. Natarajan, M. Meyer, B. Jackson, D. Slade, C. Robberts, A. Hinnebusch, and M. Marton. Transcriptional profiling shows that Gcn4p is a master regulator of gene expression during amino acid starvation in yeast. *Molecular and Cellular Biology*, 21(13):4347–4368, July 2001.

[116] J. Needham. *Science and civilisation in China*. Cambridge University Press, 1954.

[117] K. O'Connel, Y. Surdin-Kerjan, and R. Baker. Role of the *Saccharomyces Cerevisiae* general regulatory factor cp1 in methionine biosynthetic gene transcription. *Molecular Cell Biology*, 15:1879–1888, Apr 1995.

[118] E. Packham, I. Graham, and A. Chambers. The multifunctional transcription factors Abf1p, Rap1p and Reb1p are required for full transcriptional activation of the chromosomal pgk gene in *Saccharomyces Cerevisiae*. *Molecular Genetics*, 250(3):348–356, Feb 1996.

[119] X. Pan and J. Heitman. Cyclic amp-dependent protein kinase regulates pseudohyphal differentiation in *Saccharomyces Cerevisiae*. *Molecular Cell Biology*, 19:4874–4887, Jul 1999.

[120] J. Park, H. Kim, S. Han, M. Hwang, Y. Lee, and Y. Kim. In vivo requirement of activator-specific binding targets of mediator. *Molecular and Cellular Biology*, 20(23):8709–8719, December 2000.

[121] S. Park, S. Koh, J. Chun, H. Hwang, and H. Kang. Nrg1 is a transcriptional repressor for glucose repression of sta1 gene expression in *Saccharomyces Cerevisiae*. *Molecular and Cellular Biology*, 19(3):2044–2050, 1999.

[122] J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

[123] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. In *ISMB Proceedings*, pages s215–s224, 2001.

[124] R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.

[125] Y. Pilpel, P. Sudarsanam, and G. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature genetics*, 29:153–159, October 2001.

[126] K. Popper. *The logic of scientific discovery*. Basic Books, Inc., 1959.

[127] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, CH. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, JP. Mesirov, T. Poggio, W. Gerald, M. Loda, ES. Lander, and TR. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 98(26):15149–15154, December 2001.

[128] B. Ren, F. Robert, J. Wyrick, O. Aparicio, E. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. Volkert, C. Wilson, S. Bell, and

R. Young. Genome-wide location and function of dna binding proteins. *Science*, 290:2306–2309, December 2000.

[129] K. Robinson, J. Koepke, M. Kharodawala, and J. Lopes. A network of yeast basic helix-loop-helix interactions. *Nucleic Acids Research*, 28(22):4460–4466, November 2000.

[130] F. Roth, J. Hughes, P. Estep, and G. Church. Finding DNA-regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnology*, 16:949–955, 1998.

[131] M. Ruiz-Echevarria, C. Gonzalez, and S. Peltz. Identifying the right stop: determining how the surveillance complex recognizes and degrades an aberrant mRNA. *EMBO Journal*, 17(2):575–589, January 1998.

[132] W. Sabbagh, L. Flatauer, A. Bardwell, and L. Bardwell. Specificity of MAP kinase signaling in yeast differentiation involves transient versus sustained MAPK activation. *Molecular Cell*, 8(3):683–691, September 2001.

[133] K. Schrick, B. Garvik., and L. Hartwell. Mating in *Saccharomyces Cerevisiae*: the role of the pheromone signal transduction pathway in the chemotropic response to pheromone. *Genetics*, 147(1):19–32, 1997.

[134] G. Schwarz. Estimating the dimension of a model. *Annals of statistics*, 6(2):461–464, 1978.

[135] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: a probabilistic framework. In *RECOMB Proceedings*, 2002.

[136] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, and D. Koller. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, June 2003.

[137] I. Seymour and P. Piper. Stress induction of hsp30, the plasma membrane heat shock protein gene of *Saccharomyces Cerevisiae*, appears not to use known stress-regulated transcription factors. *Microbiology*, 145:231–239, January 1999.

[138] G. Shenhar and Y. Kassir. A positive regulator of mitosis, Sok2, functions as a negative regulator of meiosis in *Saccharomyces Cerevisiae*. *Molecular and Cellular Biology*, 21(5):1603–1612, March 2001.

[139] I. Simon, J. Barnett, N. Hannett, C. Harbison, N. Rinaldi, T. Volkert, J. Wyrick, J. Zeitlinger, D. Gifford, T. Jaakkola, and R. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.

[140] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS Proceedings*, 2000.

[141] A. Smith, M. Ward, and S. Garrett. Yeast PKA represses msn2p/msn4p-dependent gene expression to regulate growth, stress response and glycogen accumulation. *EMBO Journal*, 17(13):3556–3564, July 1998.

[142] M. Spector, A. Raff, H. DeSilva, K. Lee, and M. Osley. Hir1p and Hir2p function as transcriptional corepressors to regulate histone gene transcription in the *Saccharomyces Cerevisiae* cell cycle. *Molecular and Cellular Biology*, 17(2):545–552, February 1997.

[143] P.T. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces Cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[144] H. Steck and T. Jaakkola. Unsupervised active learning in large domains. In *UAI Proceedings*, 2002.

[145] M. Steffen, A. Petti, J. Aach, P. D'haeseleer, and G. Church. Automated modelling of signal transduction networks. *BMC Bioinformatics*, 3(34), 2002.

[146] M. Swanson, H. Qiu, L. Sumibcay, A. Krueger, S. Kim, K. Natarajan, S. Yoon, and A. Hinnebusch. A multiplicity of coactivators is required by Gcn4p at individual promoters in vivo. *Molecular and Cellular Biology*, 23(8):2800–2820, April 2003.

[147] M. Szummer and T. Jaakkola. Clustering and efficient use of unlabeled examples. In *NIPS Proceedings*, 2001.

[148] P. Tamayo, D. Slonin, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoertic differentiation. *PNAS*, 96:2907–2912, March 1999.

[149] A. Tanay and R. Shamir. Computational expansion of genetic networks. In *ISMB Proceedings*, pages s270–s278, 2001.

[150] A. Tanay and R. Shamir. Modeling transcription programs: inferring binding site activity and dose-response model optimization. In *RECOMB Proceedings*, pages 301–310, 2003.

[151] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. In *ISMB Proceedings*, 2002.

[152] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, July 1999.

[153] K. Tedford, S. Kim, D. Sa, K. Stevens, and M. Tyers. Regulation of the mating pheromone and invasive growth responses in yeast by two map kinase substrates. *Current Biology*, 7(4):228–238, April 1997.

[154] M. Thattai and A. van Oudenaarden. Intrinsic noise in gene regulatory networks. *PNAS*, 98(15):8614–8619, July 2001.

[155] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585, 1973.

[156] A. Tong, M. Evangelista, A. Parsons, H. Xu, G. Bader, N. Page, M. Robinson, S. Raghibizadeh, C. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294:2364–2368, December 2001.

[157] A. Tong, G. Lesage, G. Bader, H. Ding, H. Xu, X. Xin, J. Young, G. Berriz, R. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J. Levinson, H. Lu, P. Menard, C. Munyana, A. Parsons, O. Ryan, R. Tonikian, T. Roberts, A. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S. Wong, L. Zhang, H. Zhu, C. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Brescher, G. Bell, F. Roth, G. Brown, B. Andrews, H. Bussey, and C. Boone. Global mapping of the yeast genetic interaction network. *Science*, 303:808–813, February 2004.

[158] S. Tong and D. Koller. Active learning for parameter estimation in Bayesian networks. In *13th NIPS Proceedings*, pages 647–653, 2000.

[159] S. Tong and D. Koller. Active learning for structure in Bayesian networks. In *International joint conference on machine learning*, 2001.

[160] P. Uetz, L. Giot, G. Cagney, T. Mansfield, R. Judson, J. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. Rothberg. A comprehensive analysis of protein-protein interactions in *Saccharomyces Cerevisiae*. *Nature*, 403:623–627, 2000.

[161] T. Upton, S. Wiltshire, S. Francesconi, and S. Eisenberg. Abf1 Ser-720 is a predominant phosphorylation site for casein kinase ii of *Saccharomyces Cerevisiae*. *Biological Chemistry*, 270(27):16153–16159, July 1995.

[162] J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281:827–842, 1998.

[163] M. Wahi and A. Johnson. Identification of genes required for $\alpha 2$ repression in *Saccharomyces Cerevisiae*. *Genetics*, 140(1):79–90, May 1995.

[164] M. Wainwright, T. Jaakkola, and A. Willsky. Exact MAP estimates by (hyper)tree agreement. In *NIPS Proceedings*, 2002.

[165] M. Ward, C. Gimeno, G. Fink, and S. Garrett. Sok2 may regulate cyclic amp-dependent protein kinase-stimulated growth and pseudohyphal development by repressing transcription. *Molecular and Cellular Biology*, 15(12):6854–6863, December 1995.

[166] Y. Watanabe, K. Irie, and K. Matsumoto. Yeast Rlm1 encodes a serum response factor-like protein that may function downstream of the Mpk1 (Slt2) mitogen-activated protein kinase pathway. *Molecular and Cellular Biology*, 15(10):5740–5749, October 1995.

[167] K. White, S. Rifkin, P. Hurban, and D. Hogness. Microarray analysis of *Drosophila* development during metamorphosis. *Science*, 286(5447):2179–2184, December 1999.

[168] K. Williams and M. Cyert. The eukaryotic response regulator Skn7p regulates calcineurin signaling through stabilization of crz1p. *EMBO Journal*, 20(13):3473–3483, 2001.

[169] E. Winzeler, D. Shoemaker, A. Astromoff, H. Liang, K. Anderson, B. Andre, R. Bangham, R. Benito, J. Boeke, H. Bussey, A. Chu, C. Connelly, K. Davis, F. Dietrich, S. Dow, M. Bakkoury, F. Foury, S. Friend, E. Gentalen, G. Giaever, J. Hegemann, T. Jones, M. Laub, H. Liao, N. Liebundguth, D. Lockhart, A. Lucau-Danila, M. Lussier, C. Pai, C. Rebischung, J. Revuelta, L. Riles, C. Roberts, P. Ross-MacDonald, B. Scherens, M. Snyder, S. Sookhai-Mahadeo,

R. Storms, S. Veronneau, M. Voet, G. Volckaert, T. Ward, R. Wysocki, G. Yen, K. Yu, K. Zimmermann, P. Philippsen, M. Johnston, and R. Davis. Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science*, 285:901–906, August 1999.

[170] W. Xiao and G. Rank. Branched chain amino acid regulation of the ilv2 locus in *Saccharomyces Cerevisiae*. *Genome*, 33(4):596–603, August 1990.

[171] S. Xu, D. Falvey, and M. Brandriss. Roles of ure2 and gln3 in the proline utilization pathway in *Saccharomyces Cerevisiae*. *Molecular and Cellular Biology*, 15(4):2321–2330, April 1995.

[172] C.H. Yeang and T. Jaakkola. Physical network models and multi-source data integration. In *RECOMB Proceedings*, 2003.

[173] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, 2000.

[174] R. Young. Biomedical discovery with dna arrays. *Cell*, 102:9–15, July 2000.

[175] C.H. Yuh, H. Bolouri, and E. Davidson. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902, March 1998.

[176] J. Zeitlinger, I. Simon, C. Harbison, N. Hannett, T. Volkert, G. Fink, and R. Young. Program-specific distribution of a transcription factor dependent on partner transcription factor and mapk signaling. *Cell*, 113:395–404, May 2003.

[177] H. Zhu, M. Bilgin, R. Bangham, D. Hall, A. Casamayor, P. Bertone, N. Lan, R. Jansen, S. Bildlingmaier, T. Houfek, T. Mitchell, P. Miller, R. Dean, M. Gerstein, and M. Snyder. Global analysis of protein activities using proteome chips. *Science*, 293(5537):2101–2105, September 2001.