Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks

by

Alexander John Hartemink

B.S., Mathematics, B.S., Physics, A.B., Economics, Duke University (1994)

M.Phil., Economics, Oxford University (1996)

S.M., Electrical Engineering and Computer Science, M.I.T. (1997)

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

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Author .....
Department of Electrical Engineering and Computer Science
ust 2001

Certified by ....
Gifford
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

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

Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks

by

Alexander John Hartemink

Submitted to the Department of Electrical Engineering and Computer Science
on 1 August 2001, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

As molecular biology continues to evolve in the direction of high-throughput collection of data, it has become increasingly necessary to develop computational methods for analyzing observed data that are at once both sophisticated enough to capture essential features of biological phenomena and at the same time approachable in terms of their application. We demonstrate how graphical models, and Bayesian networks in particular, can be used to model genetic regulatory networks. These methods are well-suited to this problem owing to their ability to model more than pair-wise relationships between variables, their ability to guard against over-fitting, and their robustness in the face of noisy data. Moreover, Bayesian network models can be scored in a principled manner in the presence of both genomic expression and location data. We develop methods for extending Bayesian network semantics to include edge annotations that allow us to model statistical dependencies between biological factors with greater refinement. We derive principled methods for scoring these annotated Bayesian networks.

Using these models in the presence of genomic expression data requires suitable methods for the normalization and discretization of this data. We present novel methods appropriate to this context for performing each of these operations. With these elements in place, we are able to apply our scoring framework to both validate models of regulatory networks in comparison with one another and discover networks using heuristic search methods. To demonstrate the utility of this framework for the elucidation of genetic regulatory networks, we apply these methods in the context of the well-understood galactose regulatory system and the less well-understood pheromone response system in yeast. We demonstrate how genomic expression and location data can be combined in a principled manner to enable the induction of models not readily discovered if the data sources are considered in isolation.

Thesis Supervisor: David K. Gifford
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

Rare is the dissertation that is completed without significant assistance from others, and this one is certainly no exception. I owe a significant debt of gratitude to a large number of people whom I would like to recognize for their contributions to my writing, to my thinking, or to my personal development.

First, I would like to acknowledge the strong support of my dissertation committee, Professors David K. Gifford, Tommi S. Jaakkola, and Richard A. Young. Dave has been my mentor for the past five years and it has been an absolute joy to work with him each day. His sense of humor, encouragement, creative thinking, and excitement about this area of research have served as an inspiration to me. I have benefited greatly from his wisdom and experience throughout my time at MIT. It is sad to finally be leaving the academic nest, but it is comforting to go with the knowledge that I have received the best training and supervision possible. More importantly, I have a dear friend and colleague for life.

Tommi has provided me opportunities for tremendous intellectual development. He has unfailingly made time to sit down with me whenever I drop by unannounced, is a well-spring of good ideas, and has an answer for every question I ask. He is patient in explaining difficult concepts with which I am unfamiliar, and I owe him significant thanks for his continual assistance in helping me to grow in the area of machine learning.

Rick has been a tremendous encourager to me. He speaks with a smile and a sincerity that is contagious. Whenever I meet with Rick, I am greeted with a warm welcome, we discuss ideas openly and excitedly, and I leave filled with energy. His enthusiasm and support have provided a wonderfully nurturing environment for my forays into the world of molecular biology and I am indebted to him for that.

In addition to my committee, I would like to thank a number of other people at MIT. First, Jeanne Darling for being such a great friend and helper over the past five years. She has always lent a willing hand or a willing ear, depending on which was more needed at the time. Tarjei Mikkelsen deserves a lot of credit for his contributions to this group's efforts in both DNA computing and computational functional genomics, especially in the early days. It was a pleasure to supervise him and share an office with him. Additional thanks goes to

fellow graduate group members Ziv Bar-Joseph, Reina Riemann, Georg Gerber, and Julia Khodor for helpful discussions and insights along the way.

Other wonderful folks in 200 Technology Square include Alex Snoeren, who has been a tremendous friend for the past four years and with whom I sincerely hope I will be able to work in the future; Bienvenido Velez and Ron Weiss, who showed me the ropes in the early days around LCS; Ron Dror and Tony Eng, who have individually each supported me and my recent work in the form of new ideas, encouraging suggestions, and productive discussions; Kai McBride and now Greg Shomo, who keep our servers humming and try to minimize the beeping; Professors Steve Ward, Srinivas Devedas, and especially Silvio Micali, who advised me academically throughout my MIT years; Professor John Guttag, who was extremely generous with his time in providing supplemental advice during my faculty search process; Professor Tomas Lozano-Perez, who was kind enough to write a letter of recommendation on my behalf; and the Laboratory for Computer Science and the Department of Electrical Engineering and Computer Science for providing me space, resources, and a place for meeting so many tremendously gifted people.

Across the street at the Whitehead Institute, a number of people have given generously of their time to explain things to me or work with me in developing this research. Thanks goes especially to Bing Ren, Julia Zeitlinger, Peter Young, John Barnett, Tony Lee, Ezra Jennings, Nicola Rinaldi, François Robert, John Wyrick, and Nancy Hannett.

Outside MIT, I would like to acknowledge Tomi Silander for providing me the source code for the B-Course software he developed at the University of Helsinki in Finland; the software provided the basis for the model discovery implementations I developed in Chapter 7. Tomi responded quickly and thoroughly to every email I sent, and that was a huge help whenever I was under time pressure. I would also like to thank David Haussler, Kevin Murphy, Rick Lathrop, Pierre Baldi, Daphne Koller, and Russ Altman for their support, suggestions, and ideas. I would be remiss if I failed to mention David Heckerman because although he and I have not yet met, his collection of writings has provided me endless enlightenment on the topic of graphical models. Thanks also goes to the European Bioinformatics Institute, the Society for Industrial Microbiology, the University of Turku, the Pacific Symposium on Biocomputing, the SPIE Biomedical Optics Symposium, Tony Jebara and the behavior

group at the MIT Media Lab, Matt Scott's and Daphne Koller's groups at Stanford, and the computer science departments at Virginia, Duke, Michigan, UC Irvine, UC Santa Cruz, Stanford, Harvard, and Cornell for inviting me to share my work in the form of a talk.

Keeping a hungry graduate student alive for five years is a nontrivial venture and I am sincerely grateful to the National Science Foundation, the National Institutes of Health, the Merck Corporation, and the Hertz Foundation for their fellowship and traineeship support. I am also thankful for Professor David Gifford's supplemental research assistantships during my first four years at MIT.

On a personal note, I would like to thank my various housemates for all their encouragement and understanding throughout this grueling process: my brother Chris Hartemink, Travis Broughton, Paul Bollyky, Christian Ahlin, and David Joyce. Special thanks goes to Chris and Travis for making those early years at 2 Carver Street such a wonderful time, and to Paul for his uncanny gift of encouragement, first revealed to me in our days rowing together in Oxford many moons ago. My Bible studies have also been a tremendous source of support, both my current study and my previous men's study led by the amazing Bill Levin. It may seem odd but these acknowledgements would be incomplete if I didn't thank the good folks at the Duke Basketball Report for the amount of work they put in *on a voluntary basis* to keep that web site up and running; although my continual reading of the site detained me in graduate school an extra year or two, it brought me immeasurable joy.

I have saved the most important acknowledgements for last. I owe everything that I am to my wife Melissa, to my family — parents John and Dorothea, sister Alison, and brothers Chris and David — and to my new family — Sharon and Arnold Clark, James and Robin Thomforde, and Jim, Liz, Katie, and Elizabeth Thomforde. Their support has been unsurpassable and unflagging, their love unbounded and unconditional. Where would I be without all of you? My deepest thanks goes especially to my wife Melissa who patiently encouraged me throughout this process, day in and day out. I would like also to thank my grandparents John and Henrietta Wiechertjes for their courage and example. Finally, I would like to express my gratitude to my Father in heaven for giving me life, for equipping me, and for blessing me each day. May this work, and all that I do, bring Him glory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As much as we have learned about the world in which we live during the last three thousand years, scientific investigation is poised to make even greater significant progress today than at any point in history. Being woven together are multiple different threads that are enabling this rapid progress in scientific understanding.

First, our level of general technological sophistication has reached the point where we are able to develop and manufacture measurement technologies for studying the world around us with extreme efficiency and precision. Not only do these measurement technologies enable us to observe more about our world, but we also have access to ever-increasing amounts of computational power for processing the vast quantity of information we gather during our observation. Furthermore, we live in an age which is vastly more internetworked than ever before and in which an overwhelming abundance of devices and channels are available for communication. Among other things, this internetworking and communication facilitate the effective codification, curation, and exchange of scientific information.

In addition to all these infrastructural factors, we are also beneficiaries of a *network effect* with respect to scientific understanding — each advance in our understanding provides a foundation for an even greater future advance. This network effect is captured colloquially in the old saw, "For every question that is answered, two more can be asked". And while we have profited handsomely from a reductionist approach to scientific inquiry, the answers we have discovered now enable us to consider more integrationist approaches to scientific inquiry. Our success in isolating, identifying, and characterizing the various

pieces of scientific puzzles opens the door for the puzzles to be assembled while the broad infrastructural advances described above enable us to cross the threshold.

In the specific context of molecular and cellular biology, the success of high-throughput genome sequencing efforts (most notably the Human Genome Project) has revealed the nucleotide sequences that comprise the chromosomes of different organisms' genomes [74, 119]. Additionally, developments in our ability to measure gene expression on a genome-wide scale have resulted in the availability of an exponentially expanding quantity of genomic expression data. While the reductionist approach to biology has proven immensely effective over the course of the last century, and the latter half of the last century in particular, our efforts are increasingly focusing on more integrated approaches to understanding complex biological systems. All of these developments point to the need for hypothesis-driven methods for understanding, at an integrated systems level, the complex regulatory networks responsible for controlling gene expression within cells.

The work presented in this dissertation is an attempt to address this need. In this work, we develop principled and hypothesis-driven computational methods for validating and discovering models that describe the function of genetic regulatory networks. The dissertation itself takes a closer look at each of the steps necessary for successful computational elucidation of genetic regulatory networks. In this introductory chapter, we elaborate on some of the points we alluded to above and in so doing, provide a sufficient context for understanding and motivating this work. In later chapters, we explore the various steps that are required in order to move from raw genomic expression data to principled scoring methods that describe how well models explain observed expression data.

It has not escaped our attention that the vast bulk of this work is applicable in a multitude of contexts, and not simply in the domain of cellular biology. The methodology is general and thus could be used to facilitate computational scientific inquiry in a number of domains. However, after a brief discussion of general scientific inquiry, we turn our attention for the remainder of the dissertation exclusively to the domain of genetic regulatory network elucidation for reasons of limited space and time.

1.1 Scientific inquiry

The process of scientific inquiry is a repeated cycle of observation and explanation. The earliest stages of the cycle consist of pure observation: the gathering of systematic data about the natural world. Before questions and answers can be formulated, we must first observe because we must gather the raw material out of which to fashion questions and answers. Before we can ask why the sky is blue, we must observe it to be so.

At various stages in our scientific progress over the last three millennia, we have undertaken periods of intense observation. Sometimes a perceptive eye notices something in its environment, as with the ancient Greeks or Charles Darwin, or sometimes a precise mind decides to carefully record specific phenomena, as with Tycho Brahe capturing the motions of the planetary bodies in their celestial orbits or Gregor Mendel counting varieties and variations of peas. At other times, a flurry of detailed observation is due to the invention of a measurement technology that permits new kinds of observations to be made for the first time. Examples include the telescope, which enabled Galileo to observe the motions of sunspots, to map out the surfaces of the moon, to detect rings encircling Saturn, and to notice moons orbiting Jupiter; the microscope, which allowed Hooke to observe cells for the first time; the X-ray; the radio telescope; the interferometer; the cyclotron and other particle colliders; the DNA sequencer; and the gene expression array, about which we will have more to say later in this chapter.

After a period of observation, curious minds naturally begin to ask questions about the regularities that exist in what they observe, and in some cases, perhaps about the absence of regularities in what they observe. Questions are formulated, and possible explanations, or *hypotheses*, are postulated. Frequently, a number of hypotheses are not consistent with observed phenomena and can be quickly rejected, but on the other hand, for any observed phenomenon, there may be a number of hypotheses that are consistent with the data that have been observed to date and none can yet be rejected. An intriguing example of such an instance is the Mpemba effect, named after the Tanzanian ice-cream maker who rediscovered the phenomenon in 1969, although it has been observed empirically at least as far back as Aristotle. The Mpemba effect describes the phenomenon whereby under certain conditions,

a certain volume of warm water freezes faster than an equal volume of cooler water. Current hypotheses for explaining this phenomenon include a difference in dissolved gas content, a reduction of mass through evaporation, an establishment of convection currents that modify the distribution of heat, and a differential amount of supercooling between the warmer and cooler water. To date, none of these has been shown to explain the phenomenon conclusively and so multiple hypotheses coexist for now [10].

After the postulation of a number of hypotheses that are consistent with the data, usually more data is gathered. However, in contrast to the initial set of observational data, the data gathered at this stage is usually the result of a carefully planned collection of experiments, designed to force the system of interest to reveal critical information regarding the tenability of the various alternative hypotheses. We distinguish between two cases. The first is that the data remain observational but are gathered with increased precision, as may be the case, *e.g.*, in experiments designed to confirm the predicted mass of various subatomic particles. The second is that the data are interventional in the sense that specific variables of interest are forced to obey certain sets of constraints, as may be the case, *e.g.*, when recombinant DNA methods are used to over- or under-express a gene of interest. These interventional types of experiments should be designed with enough constraints so as to be able to reject as many hypotheses as possible that are inconsistent with the underlying truth about the operation of the system in question.

In the event that multiple hypotheses remain, more experimentation and observation are necessary to distinguish between alternative explanations of phenomena. In the event that no hypotheses remain, either new hypotheses need to be generated, new data needs to be gathered, or perhaps a revolution waits to be undertaken, a time when not only are new hypotheses postulated but these new hypotheses in some ways contradict previously held hypotheses about the working of the natural world [73]. Examples of such times might be the introduction of the Copernican view of the solar system, Darwin's elaboration of the theory of natural selection, or the advent of quantum mechanics.

1.2 Genomic biology

Although the principles of scientific inquiry outlined above are general, we focus in this dissertation on how they play out in the domain of genomic biology. With the recent announcements regarding the first tentative assemblies of the human genome [74, 119], the field of genomics is abuzz with excitement, and the bulk of this excitement is not with respect to what has already been learned but with respect to what will soon be able to be learned. The sense of eager anticipation in terms of the potential impact genomic biology will have on human health and well-being is palpable.

The vast quantity of data being generated throughout genomic biology affords researchers a significant opportunity to use systematic computational methods to transform our understanding of the cellular processes governing life. The availability of genomic (and eventually proteomic) expression and location data will have a profound impact on the understanding of cell biology, the diagnosis and treatment of disease, and the efficacy of designing and delivering targeted therapeutics. Particularly relevant to these objectives is the development of a deeper understanding of the various mechanisms by which cells control and regulate the transcription of their genes. In the following subsections, we lead up to a discussion of this topic by first considering the various different subfields within genomic biology for context.

1.2.1 Sequential genomics

The principle aim of sequential genomics is the determination of the sequences of nucleotides that comprise the genomes of various living organisms. In addition to sequencing an organism's chromosomes, sequential genomics is also concerned with identifying the subsequences of the genome that correspond to the organism's genes, introns and exons, protein coding sequences, and regulatory elements (like promoters). The genomes of a large number of simple organisms have already been sequenced, including fruit-fly, roundworm, yeast, and dozens of species of bacteria, and the fully sequenced human genome should be completed in the next year or two.

The completion of the Human Genome Project is one of the most significant scientific

accomplishments in recent history. As of early 2001, about 94% of the genome has been sequenced in preliminary draft form and about 33% of the genome has been sequenced to the point of being declared finished. It has taken more than ten years of coordinated effort among thousands of individuals around the globe, armed with hundreds of millions of dollars of funding from governments and corporations, to decipher this sequence. Although the determination of this three billion nucleotide sequence represents a tremendous scientific achievement in its own right, its primary impact is in terms of the potential associated with its being known. Genes are blueprints for proteins, the molecular workhorses with roles in cellular structure, motility, metabolism, homeostasis, signaling, signal transduction, reproduction, repair, and regulation. Knowledge of these genes provides not only basic biological insights into the intricate functioning of the cells that make up living organisms, but also the ability to understand, diagnose, and hopefully ultimately treat a wide range of diseases like Alzheimer's, cystic fibrosis, and cancer.

The process of locating genes within sequenced chromosomes can be a challenge. One helpful method is to have access to sequenced proteins that are produced by genes in the genome under consideration. Since the mapping between the nucleotide sequences of genes and the amino acid sequences of proteins is known, knowledge of protein sequence is of great assistance in identifying the coding regions of genes in the genome. Sequencing mRNA transcripts (or equivalently cDNA) from the organism under study is similarly helpful. In the absence of such corroborating sequence information, algorithmic methods have been developed for locating likely gene regions purely on the basis of various statistical properties associated with these regions over variables such as gene length, initiation and termination sequence conservation, presence of common upstream regulatory elements, relative abundance of GC nucleotides versus AT nucleotides, and sequence similarity with genes in the genomes of other organisms [17, 18].

1.2.2 Structural genomics

The principle aim of structural genomics is the determination of the three dimensional structures of the various proteins that are encoded for by the genes in living organisms' genomes. As such, the name is perhaps slightly misleading since it might better be called

structural proteomics, while structural genomics could be reserved for the study of the three dimensional structure of organisms' chromosomes *in vivo*. In the latter case, the processes under investigation might include the packing of DNA into chromatin, the association of DNA with histones and other such structures, and the subtle dance of alternately hiding and revealing to transcriptional and replication machinery the parts of the genome that need to be transcribed and replicated, respectively.

However, with respect to structural genomics *qua* determination of the three dimensional structures of proteins, the problem arises because proteins do not function inside cells as linear molecules. While sequential genomics is useful in determining the linear sequences of amino acids comprising proteins, it does not reveal how these proteins fold into the specific structures necessary for the execution of their roles within the cell. The only sure way of assessing the structure of proteins currently is to carefully isolate the protein of interest and then employ either X-ray crystallography or NMR to measure the positions of various atoms in the molecule, all of which takes a tremendous amount of time and laboratory finesse. Alternatively, if a number of protein structures have been determined experimentally, it is possible to predict protein structures for sequences that are homologous to those whose structures have already been determined, a process known as *protein threading*. Moreover, a number of people have proposed methods for determining protein structures *ab initio*, without the aid of already known protein structures, but most of these methods remain in their infancy due to modeling and computational limitations.

Other interesting questions about protein structure also fall within the domain of structural genomics/proteomics, including categorization of the various types of proteins folds or motifs, characterization of the various classes of protein families, determination of active sites within proteins, and construction of artificial proteins or ligands for enabling or disabling various protein functions. While the next ten years will certainly witness significant advances in these areas, and in protein structure determination itself, the primary purpose for identification of protein structure is the extent to which it helps us understand protein function. In other words, the ultimate motivation behind most of this investigation remains the determination of the roles that these various proteins play inside the cell.

1.2.3 Functional genomics

The principle aim of functional genomics is the determination of the functions of the various proteins that are encoded for by the genes in the genomes of living organisms. What do all these genes, and by extension, what do all these proteins do? Elucidating the functions of the diverse collection of proteins within cells is the basis of functional genomics and will be the fundamental driving question of biology for at least the next ten to twenty years, and probably much longer.

Although proteins play a number of different roles within cells, one of the most intriguing such roles is that of genetic regulation: control of precisely which genes are expressed and which remain unexpressed at any given time in the cell. Through these genetic regulatory mechanisms, proteins are responsible for controlling their own existence, and yet very little is known about the sets of signals and controls that activate and repress the expression of specific genes. We apply our methods for principled computational inquiry in this domain in order to help reverse-engineer these genetic regulatory networks from observations of the levels of gene expression under varying environmental and genotypic conditions.

1.3 Collecting genomic expression data

A number of technologies exist for gathering data characterizing the levels of gene expression of cells on a genome-wide scale. The most prominent such methods are array-based, but other methods are also used in certain contexts. We discuss a number of these technologies in turn.

1.3.1 Lithographic oligonucleotide arrays

In this dissertation, we consider genomic expression data gathered using Affymetrix GeneChips, which are high-density oligonucleotide arrays printed using a lithographic masking process. GeneChip arrays consist of tens of thousands of *features*, each containing a unique set of short DNA strands that act as probes for binding specific target nucleic acid molecules. To quantify the genome-wide levels of gene expression for a population of cells, mRNA transcripts are extracted from the cells, labeled with fluorescent tags, and hybridized

to arrays containing features designed to collectively probe for all the various genes in the genome.

The oligonucleotide DNA probes on Affymetrix arrays are frequently 25-mers, and each target species has 40 different probe features associated with it on the array, half of which are designated as perfect match (PM) probes and half of which are designated as mismatch (MM) probes.[1] The perfect match probes for a target species are complementary to 25-nucleotide subsequences of the target. The 25-nucleotide subsequences corresponding to the multiple PM probes for a given target are carefully selected so as to optimize sequence specificity for the target (*i.e.*, contain as little redundancy with other parts of the genome as possible) as well as minimize probe secondary structure and ensure roughly uniform hybridization energies. For each of these PM probes, the corresponding mismatch probe is created by making a single-nucleotide substitution in the central position of the 25-mer. The MM probes are used as a control, in order to be able to quantify the amount of non-specific hybridization that may be taking place at the PM probe.

Affymetrix reports the expression level of a target mRNA species using a method it calls *average difference score*, which is exactly as it sounds:

$$\mathsf{AverageDifferenceScore} = \frac{1}{N}\sum_{i=1}^{N}(PM_i - MM_i) \tag{1.1}$$

where PM_i and MM_i are the fluorescent intensities measured at the perfect match and mismatch probes, respectively, and i indexes the multiple probe pairs associated with the target mRNA species. Although N is bounded by the total number of probe pairs associated with a single target, Affymetrix applies an additional method known as *superscoring* which may eliminate certain probe pairs that have high variability. For this reason, the value of N can vary from target to target even when all targets have the same number of probe pairs on the array. More detailed descriptions of the algorithms can be found in [1]. Li and Wong have recently proposed improvements to this method [77] but those improvements are not yet available in practice.

[1]Original designs had 20 probe pairs (40 total probes) in the probe set for each target species, but recent designs sometimes use 16 probe pairs (32 total probes) in each probe set.

To carefully engineer the probe sequences on the array requires that we know the sequences to be detected. In particular, since these arrays can only be built using probes of known sequence, genome-wide quantification of gene expression can only be done for organisms whose genome has been sequenced. The sequences within each feature on the array must be known in advance because the 25-mer probes are constructed one nucleotide at a time using a lithographic masking technique similar to that in use in the semiconductor industry.

The first step in this lithographic deposition process is to apply uniformly to the array linker molecules that serve as a bridge between the array substrate and the eventual oligonucleotide probes. The linker molecules have at their free ends a photo-labile group that acts as protection against further binding. This linker attachment process is followed by 100 rounds of lithographic masking: 4 rounds for each of the nucleotides that make up the 25-mer probes.

In the first round, all probes on the array whose first nucleotide to be added is adenosine (A) are deprotected by shining light on those probes and detaching the photo-labile protective group. The non-adenosine probes on the array are "masked" from the light, causing their protective groups to remain attached. After this deprotection, adenosine groups (themselves containing a photo-labile protective group) are washed across the array and bind to the linkers for the probes that are deprotected. After this is complete, all probes should once again have protective groups at their free ends.

In the second round, all probes on the array whose first nucleotide to be added is cytosine (C) are deprotected by shining light on those probes and detaching the photo-labile protector. Similar to before, the non-cytosine probes on the array are "masked" from the light, causing their protective groups to remain attached. After this deprotection, cytosine groups that themselves contain a photo-labile protective group are washed across the array and bind to the linkers for the probes that are deprotected. In the third round, a similar process binds guanosine groups where they should be bound and in the fourth round, thymosine groups are added. These four rounds are then repeated 24 more times until each probe in the array is a 25-mer of exactly the correct sequence. Of course, there can be infidelities in the manufacturing process, but the presence of $\sim 10^6$ probe 25-mers

per feature and the use of 20 probe pairs per target species are intended to make the technology robust in the face of these infidelities. The lithographic deposition process is patented and the patent is licensed exclusively to Affymetrix, but the method is described in various papers [41, 95, 81, 87]. Incyte produces similar *in situ* oligonucleotide arrays using piezoelectric (ink-jet) deposition technology [64].

1.3.2 Printed cDNA arrays

Compared with oligonucleotide arrays, printed cDNA gene expression arrays use much longer strands of cDNA to probe for target mRNA extracted from the sample of interest. In this case, probes are typically a few hundred nucleotides in length and their sequence may not be known — since the probes need not be constructed one base at a time, it is possible to print arrays from cDNA libraries containing unsequenced polynucleotides. The manufacturing process for these arrays essentially consists of depositing small quantities of probe in solution onto a coated glass slide or nylon membrane. The droplets are then dried in place and the probes are cross-linked to the slide or membrane surface. The deposition technology ranges widely. Some common methods are piezoelectric (ink-jet) deposition [64], advanced glass matrix deposition tips [30], or simply a physical print tip similar to a fountain pen [34] or pin and ring [2].

In the earliest implementations of this array technology, the probe deposition process was fairly erratic. Consequently, the intensity of bound target at each spot on the array could vary widely simply because the amount of probe deposited at each spot varied widely. Because of this shortcoming, samples were (and still are) hybridized to these arrays in a competitive hybridization context wherein two samples, each labeled with a different fluorophor, are bound to the array at the same time. Rather than reporting absolute intensities of each species, only relative intensities are reported because only relative intensities have a meaningful interpretation in this setting. The presence of two different dyes leads to interesting analysis complications such as the fact that cross-talk emerges between the dyes in that each responds somewhat to the intensity of light used to probe the other [69], the fact that the two dyes have different response ranges leading to skew results at extreme ends of the dynamic range [70, 62], and the fact that the two scan images corresponding to the two

wavelengths of light used to scan the array need to be aligned with one another. The latter problem is often solved by scanning at both wavelengths at the same time, but that adds some complication to the measurement technology and enhances the cross-talk.

The progression of this technology from scattered laboratory benches to large-volume manufacturing processes at corporations like Incyte, Corning, and Affymetrix promises to make available at low cost reliable arrays with minimal spot-size variation. If that can be achieved, not only will reproducible commodity expression arrays be available to the general public, but it may become possible to use printed arrays to measure absolute target abundances.

1.3.3 SAGE and RT-PCR

While array-based technologies for high-throughput quantification of gene expression dominate the landscape today, it should be noted that other methodologies are also used with success. In particular, two such methods that deserve mention are Serial Analysis of Gene Expression (SAGE) and reverse transcriptase polymerase chain reaction (RT-PCR). However, for the remainder of this dissertation, we only discuss expression data gathered with Affymetrix oligonucleotide arrays (GeneChips) and location data gathered with printed arrays. Nevertheless, with the possible exception of some of the normalization methods of Chapter 2, the results presented in this dissertation are fairly general and should be readily applicable to data gathered on other technology platforms. In addition, a significant amount of interesting work remains to be done in combining expression data collected using different measurement technologies so that they are inter-comparable, a brief discussion of which can be found in Section 8.2.1 of the concluding chapter.

1.4 Paradigms for the analysis of genomic expression data

With the recent invention of these powerful new technologies for measuring the genome-wide expression of genes, we are in the midst of a period of wonderment and fascination, much as when Galileo first constructed his telescope. There is so much to see and to observe and consequently so much energy has been devoted to collecting data with these technologies

that comparatively little effort has been put into formulating hypotheses describing the operation of the underlying biological systems responsible for controlling gene expression. The production and distribution of these measurement technologies and the collection of data have successfully been pursued, but putting the data together into a coherent picture of the operation of genes within the cell has been less forthcoming.

1.4.1 Data-driven analysis of genomic expression data

The first forays into analysis of genomic expression data can be characterized primarily as data-driven, in the sense that they have focused on the discovery of patterns within the observed data itself. Within this analysis paradigm, data gathered from expression arrays is first preprocessed to make it comparable with other such data, and then the resultant data matrix (genes $\times$ experiments) is mined for interesting patterns. Common methodologies for data analysis include smoothing data, extracting trends from data, correlating data vectors, clustering data, ordering clustered data, labeling clustered data, categorizing data, using principled component analysis or singular value decomposition to extract patterns in data, and representing suitably analyzed data in suggestive visual forms. Extensions to this basic idea include identifying clusters with common *cis*-acting sequence motifs [114, 104] and computing regulatory dependencies by correlating lagged time-series data [21].

This paradigm has proven quite successful in identifying a number of striking patterns within gene expression data. For example, various genes of similar function often cluster together [33, 39, 122, 65, 113, 62, 103], especially when the topological clusters are optimally ordered [11], certain genes from cell-cycle synchronized cells are noted to behave cyclically with periodicity related to the underlying period of the cell-cycle [26, 109, 5], and various genes have been identified that seem to offer some predictive power in terms of categorizing types of cancers [53, 4], and even subtypes of cancers based not only on morphology but on other phenotypic variables like mortality or response to treatment [4, 20].

Unfortunately, these data-driven techniques for analyzing genomic expression data generally do not permit the rigorous statistical testing of hypotheses about the function of the complex regulatory networks responsible for transcriptional control. Moreover, although we know that cells regulate transcription through combinatorial multi-variate con-

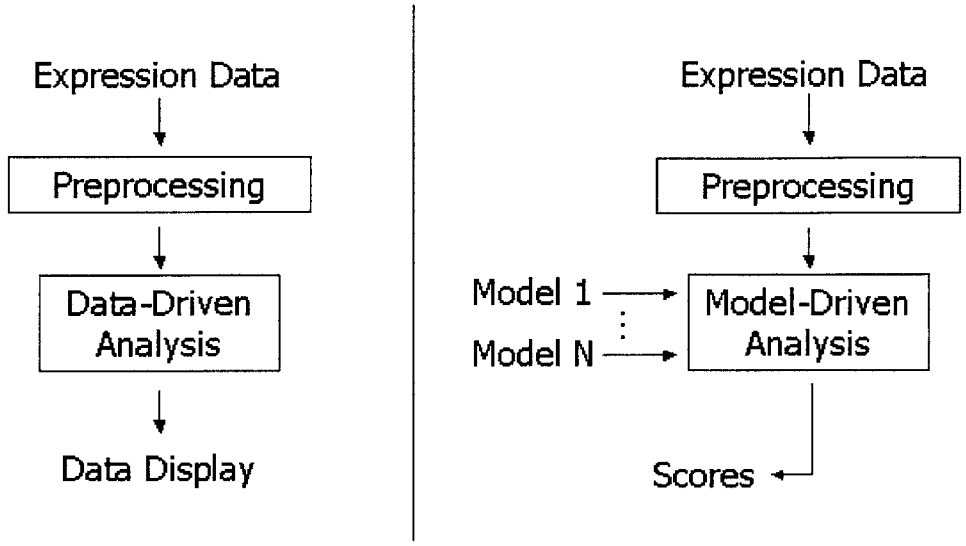

Figure 1-1: *Comparison of two different frameworks for the analysis of genomic expression data. The prevailing data-driven analysis paradigm is depicted in schematic form on the left, while the proposed model-driven analysis paradigm is depicted in schematic form on the right. The essential difference between the two paradigms is that in the former, the results are the data, shuffled, rearranged, summarized, and visualized for the user in suggestive ways. In the latter, the results are principled scores that provide a direct measure for comparing the posterior likelihood of the models in the presence of observed data.*

trol processes [61, 83], most of these methods rely on pair-wise measures such as correlation [39, 109, 65, 4], Euclidean distance [122, 113, 114], or (pair-wise) mutual information [35, 19] to calculate gene expression similarity. As noise in expression array data is typically not analyzed in detail, the significance of alternative conclusions from these studies cannot be quantitatively compared. Finally, a single framework currently does not exist that permits models to describe latent variables (such as protein levels) and make predictions that can be verified later as data becomes available.

1.4.2 Model-driven analysis of genomic expression data

These data-driven analysis methodologies have been useful in uncovering interesting patterns or regularities in the expression data that need to be explained. To explain these patterns, however, we would like to be able to postulate models describing the underlying

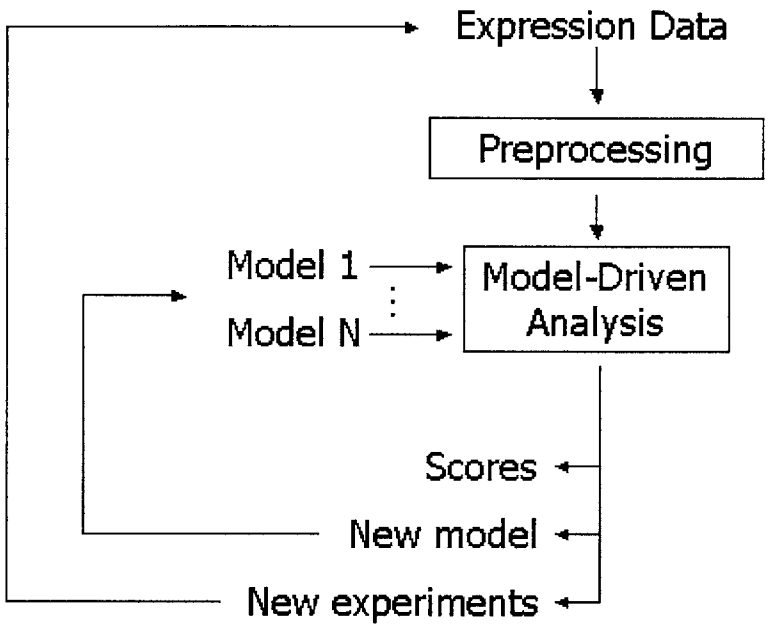

Figure 1-2: *The model-driven analysis paradigm lends itself to further extensions. In particular, a principled scoring method for comparing models paves the way for model discovery, which we address later in this dissertation. Feedback in the form of experimental suggestions for guiding the data-collection process is also possible.*

biological mechanisms that give rise to them and then score these models in order to determine which are most consistent with the observed data. With the large amounts of available data and the complex and frequently combinatorial biological systems that are responsible for generating this data, automated computational methods for discovering and validating hypotheses would be quite useful. We therefore propose a model-driven framework for the analysis of gene expression data in which we represent hypotheses about the function of underlying genetic regulatory networks in a compact probabilistic form and develop principled methods for scoring these hypotheses in comparison with one another in terms of their relative ability to explain noisy expression data. This model-driven framework is shown contrasted with the data-driven framework in Figure 1-1 and is presented in greater detail in Chapter 4.

In addition to providing principled ways to validate models against one another in terms of their ability to explain observed data, a model-driven analysis paradigm lends

itself to further extensions as shown in Figure 1-2. In particular, a principled scoring method for comparing models paves the way for model discovery, which we address later in this dissertation. Feedback in the form of experimental suggestions for guiding the data-collection process is also possible but the complete methodology is not developed here. A brief discussion of the topic is presented in the concluding chapter, however.

1.5 Collecting genomic location data

It should be noted that genomic expression data is not the only source of data available to us in the process of trying to reverse-engineer genetic regulatory networks. For instance, whole-genome arrays can be used not only for gathering information about genome-wide levels of gene expression but also for gathering information about genome-wide locations of DNA binding factors. Recent work by Ren, *et al.* [99] describes how chromosomal DNA bound by DNA binding factors can be selectively and differentially amplified using a chromatin immunoprecipitation protocol and then, using printed arrays whose probes consist only of intergenic regions of DNA, quantified in comparison with chromosomal DNA taken from whole cell extracts.

This assay enables the various intergenic binding locations of transcription factors, for example, to be identified. In particular, this information can be useful in helping identify potential targets of transcription factors. A number of subtleties must be considered, such as the possibility that a protein binds to certain regions of DNA as a transcription factor and to other regions of DNA as an inert factor, but the technology is already providing significant insights into the function of numerous DNA binding factors and is exploited in Chapter 7 of this dissertation to help guide the discovery of models of the regulation of genes involved in yeast pheromone response.

1.6 Dissertation roadmap

In this dissertation, we propose a computational framework based on graphical models, and Bayesian networks in particular, for formulating hypotheses about the function of complex genetic regulatory networks. With this framework for the computational rep-

resentation of regulatory network hypotheses in place, we develop a methodology for the model-driven investigation of genomic expression and location data and demonstrate the end-to-end application of this methodology in the context of both model validation and model discovery using data from *Saccharomyces cerevisiae*.

The following two chapters of the dissertation address the issue of suitable data preparation. In order to make the data from different arrays comparable, we begin in Chapter 2 by developing principled maximum a posteriori methods for normalizing genomic expression data collected on Affymetrix GeneChip arrays. We demonstrate how these methods reduce undesirable variation more effectively and under a broader range of conditions than other methods proposed to date. In Chapter 3, we examine various methods for the discretization of genomic expression data in order to reduce the dimensionality of our modeling problem. We show how expression data can be discretized in such a way as to minimize the loss of information regarding the relationships between genes and conclude that very little predictive information is lost in the discretization process until a relatively small number of discretization levels remain. These two chapters are near the beginning of the dissertation because they do not depend on the following chapters and are interesting in their own right, independent of the hypothesis validation and discovery methodology presentation that follows.

The next two chapters of the dissertation address the issue of suitable computational representations of hypotheses regarding the function of complex genetic regulatory networks. In Chapter 4, we discuss the applicability of graphical models, and Bayesian networks in particular, to our modeling task. We present Bayesian networks as a compromise between on the one hand, representations that are too simplistic to capture important biological regulatory phenomena and on the other hand, representations that are too complicated to be specified or discovered from relatively sparse amounts of data. We also discuss how these Bayesian network models can be scored in a principled way using the Bayesian scoring metric and how this metric enables both model validation and model discovery. We then turn in Chapter 5 to extend the semantics of Bayesian networks by adding the ability to represent refined knowledge about the relationships between variables using graph edge annotations. These annotations allow us to model more refined regulatory relationships like activation

and repression that are not easily expressed in the language of conditional dependence and independence associated with standard Bayesian networks. We develop principled methods for scoring annotated Bayesian networks that preserve the inherent penalty for complexity that is present in the traditional Bayesian scoring metric.

We consider the application of these normalization, discretization, model formulation, model annotation, and model scoring methods in the next two chapters. In Chapter 6, we apply these methods in the context of the galactose regulatory network in *S. cerevisiae*. The function of this network has been worked out extensively in the yeast community and thus the system offers us the opportunity to compare the effectiveness of our model validation and annotation methodologies against what is already known about this system. In contrast to this well-understood galactose system, we consider in Chapter 7 a collection of genes that are all related to pheromone response in *S. cerevisiae* but about which significantly less is known in terms of the structure and function of the associated genetic regulatory network. Working in this context offers us the opportunity to exploit our model discovery methodologies to elucidate and discover pieces of this regulatory network that have not been known before. In this chapter, we consider and apply various mechanisms for the automatic elucidation of Bayesian network structures.

Chapter 8 closes the dissertation with a discussion of the lessons learned, as well as the specific contributions this work makes to the existing body of literature. The chapter also includes a fairly comprehensive itemization of various ways in which this work could be extended in the future. In this light, the concluding chapter puts the entire body of work in its proper context, both with respect to what has been done in the past and with respect to what can be done in the future.

Chapter 2

Normalization of genomic expression data

Expression arrays provide a powerful mechanism for measuring genomic expression levels within populations of cells. As discussed in the introductory chapter, these arrays permit the simultaneous detection and measurement of tens of thousands of species of mRNA in a single experiment. If this vast quantity of genomic expression data can be suitably utilized, the impact of this data on the understanding of basic cellular processes, the diagnosis and treatment of disease, and the efficacy of targeted therapeutics will be profound.

The effective analysis of large amounts of genomic expression data presupposes that the data is both available and comparable. Ensuring the availability of genomic expression data requires that we gather currently dispersed stores of data into large, publicly available databases and exchange data using standardized interchange formats. A number of database schemas have been proposed [8, 48, 50, 49] and various data interchange formats are currently under development [51, 47]. However, all these efforts aimed at achieving data availability are of little value unless accompanied by data comparability. Making genomic expression data from different experiments comparable is the issue undergirding the topic of normalization we consider in this chapter.

In particular, while genomic expression data contains significant variation of interest (biologically meaningful variation), this information is often obscured by other sources of

variation which appear as random error in reported expression levels. The motivation for normalization is to remove, or otherwise account for, random error in order to make the remaining interesting variation as comparable as possible.

Section 2.1 of this chapter characterizes the different sources of variation in expression array data in order to provide a suitable background for tackling the normalization problem. Section 2.2 discusses some previously proposed methods for data normalization while Section 2.3 raises the issue of multiple sources of normalization information. We then formulate the normalization problem in Section 2.4 and present maximum likelihood (ML) and maximum a posteriori (MAP) estimates for parameters used in modeling the reported expression levels. In Sections 2.5 and 2.6, we use these estimates to calculate optimal array scaling factors for a data set consisting of 1280 Affymetrix GeneChip arrays and compare the results to those derived using other normalization methods. We close in Section 2.7 by discussing these results and offering some directions for further investigation.

2.1 Variation in genomic expression data

When measuring genome-wide levels of gene expression, we seek to learn how cells variously express their different genes in response to the diverse genetic and environmental environments they encounter. We define these sources of variation collectively as *interesting variation.* Unfortunately, reported expression levels also include other sources of variation that obscure the variation of interest. Sources of *obscuring variation* include variation introduced during the process of sample preparation, during the manufacture of the array, during the hybridization of the sample on the array, and during the scanning and analysis of fluorescent intensity after hybridization. Before we can separate obscuring variation from interesting variation within reported levels of expression, we first need to characterize how this variation arises. To this end, we discuss each of these sources of variation below.

2.1.1 Sources of interesting variation

Variation in the expression of genes arises at many different levels. At the lowest level, even if we consider a specific gene in a specific cell under a specific environmental condition,

there may be variation in the level of gene expression since mRNA transcription and decay are discrete stochastic processes. In practice, most of the variation at this level is hidden by the limitations of current array technology since we cannot measure gene expression for a single cell but are constrained to measure an ensemble average over a population of cells.

At the next level, if we examine the expression of multiple genes, we observe more variation in the data because different genes are expressed in cells at different levels. This variation in the levels of expression across the various genes in a cell's genome gives rise to an *expression profile* for a cell population (conditional on the population genotype and the specific environmental milieu in which the population is observed), acting as a genetic signature of sorts.

At yet the next level, if we consider the expression of genes under a diversity of conditions, we observe even more data variation because the expression profiles for populations of cells depend dramatically on the genetic and environmental conditions that prevail when the cells are observed. For example, knocking out the activity of a protein, altering the temperature, modifying the nutritive environment, or exposing cells to agents that induce infection, mutation, cellular stress, or signaling cascades can all have a significant influence on the expression profile of the population.

2.1.2 Sources of obscuring variation

In addition to these sources of interesting variation, a number of sources of obscuring variation also exist. Sources of variation introduced during the preparation of sample include variation during mRNA extraction and isolation, variation in mRNA amplification, variation in the introduction of fluorescent tags, and variation in the rate of fluorescent tag incorporation. These are influenced by pipette error, temperature fluctuations, reagent quality, and human error.

Sources of variation introduced during the manufacture of the array include variation in the amount of probe present at each feature or spot and variation in the hybridization efficiency of the probes for their mRNA targets. The factors that influence these sources of variation depend upon the type of array being used. In the case of Affymetrix GeneChip oligonucleotide arrays [95, 87], probe concentration and efficiency are influenced by substrate

surface characteristics, linker effects, probe design and density, and hybridization kinetics and thermodynamics [86, 42]. In the case of printed cDNA arrays, probe concentration and efficiency are influenced by substrate surface characteristics, cross-linking effects, cDNA library selection and amplification, hybridization kinetics and thermodynamics, and probe deposition technology.

Sources of variation introduced during hybridization of the sample on the array include variation in the amount of sample applied to the array and variation in the amount of target hybridized to each particular probe. The amount of target-probe hybridization is influenced by the nature and concentrations of the buffers being used, the temperature and duration of the competitive hybridization reaction, the amount of cross-hybridization interference, and the possibility of probe saturation.

Sources of variation introduced after array hybridization include variation in optical measurements, variation in the fluorescent intensity computed from the scan image, and, in the case of printed arrays, variation in the optical response of the different dyes present in the sample [70]. These can be influenced by spot misalignment, discretization and pixelation effects, edge detection errors, poor intensity calculation algorithms, and scanner lens and laser irregularities.

2.1.3 Separating interesting variation from obscuring variation

The process of producing good estimates of true expression level can be divided into two stages. In the first stage, the pixelated image is processed, features (spots) on the array are located and bounded, background intensity is calculated, and then intensity measures at various pixels within the feature boundary are combined in some manner to produce a single numerical value indicating the total intensity associated with each feature in the array. In the case of Affymetrix arrays, these estimated feature intensities are further processed using the average difference and superscoring techniques mentioned in the introductory chapter to produce a total intensity associated with the target species being probed. Regardless, the image data has been converted to numerical data containing reported numerical expression values. In the second stage, these reported expression values need to be normalized against one another to compensate for biases or systematic errors across features, genes, or

arrays. It should be mentioned that further estimation problems arise even after suitable normalization has been performed. For example, estimating the ratio of two measurements or combining replicate measurements into a single expression value necessitate careful analysis. In this context, Dror, *et al.* [37] develop a Bayesian method for computing these estimates under a particular noise model for Affymetrix arrays.

While many interesting statistical and computational data processing issues arise in the first stage [15, 12, 69, 107], we concentrate here on the second stage. In other words, we assume that we begin not with image data but with numerical values of reported gene expression.

As reported expression levels are a combination of interesting variation and obscuring variation, we need a suitable method for separating the two, where possible. Ideally, given reported levels of expression for a collection of genes across a number of experiments, we would like to develop statistically sound estimates for the levels of gene expression that include interesting variation but exclude, or otherwise account for, obscuring variation.

We should clarify that this obscuring variation can occur at either the array level or the feature level (or both). Moreover, the obscuring variation can be either additive or multiplicative in nature (or both). From the characterization of obscuring variation discussed in the previous section, we see that most of the variation is array-level multiplicative error. That is, the error affects every target in the sample applied to the array and does so in a manner proportional to the amount of target in the sample. We defer discussion of this assumption until Section 2.7.

In this chapter of the dissertation, we develop a new model for deriving estimates for underlying levels of gene expression that account for array-level multiplicative error in the specific context of Affymetrix GeneChip arrays. While there remain sources of obscuring variation that cannot be accounted for in the model, we seek to present a simple model that adequately explains a substantial amount of this variation. We also demonstrate how this method compares with other previously proposed methods. We begin by discussing these existing methods for array normalization.

2.2 Existing methods for array normalization

There have been three proposed classes of GeneChip normalization methods proposed to date, differing primarily in terms of the sources of information that they use to normalize each array. The first class relies on normalizing reported expression values based on information gathered from the expression intensities of all genes on the array. The second class relies on normalizing reported expression values based on information gathered from the expression intensity of certain specific genes on the array. The third class relies on normalizing reported expression values based on information gathered from the expression intensity of exogenous genes that are spiked into the mRNA extract samples applied to the array. We discuss each of these three classes of methods in turn.

2.2.1 Methods based on expression of all genes

A number of previous efforts at array normalization are based on information from all of the genes in the genome. We consider in this section three of these methods. The first is the method of *average intensity normalization.* In this method, the total amount of mRNA produced by a population of cells is assumed to be constant. Under this assumption, the total fluorescent intensity associated with the genomic expression of a population of cells ought to be constant. Unfortunately, it is often the case (and indeed the case for the Affymetrix Ye6100 GeneChips used throughout this dissertation) that a single array is not sufficient for measuring genome-wide levels of gene expression. In these cases, multiple arrays are required to make even a single genome-wide observation and we thus need an additional assumption. Therefore, we make the further assumption that the average intensity of reported expression values should be constant across each of the arrays necessary to measure genomic expression. This assumption relies on the fact that the genes of the genome are distributed roughly evenly according to average intensity across each of the arrays necessary to measure genomic expression. In particular, we use average intensity rather than the previously equivalent total intensity because different arrays may measure expression levels for different numbers of genes. We discuss the suitability of these assumptions below.

If all the error is assumed to be array-level multiplicative, then normalization is equiva-

lent to computing a suitable scaling factor for each array. Under the assumptions of constant average intensity outlined in the previous paragraph, the appropriate scaling factor for the normalization of array j can be computed by averaging the reported expression values of the M genes indexed by i in array j and then using this mean, $\frac{1}{M}\sum_{i=1}^{M} x_{ij}$, to renormalize the expression values on array j. The ratio of the scaling factors between two arrays is thus equal to the ratio of the arrays' average intensities. A simple alternative to this method is to normalize on the median intensity for each array rather than the mean intensity. Although the median is less sensitive to outliers and thus theoretically more robust than the mean, its empirical performance does not seem to be significantly different from that of the mean. Henceforth, we shall consider only normalization based on mean intensity because the assumptions on which it is based are stated explicitly.

This average intensity normalization method (as well as the median variant) is problematic, however, for two reasons. First, the total level of mRNA being produced by a population of cells under many conditions cannot be assumed to be constant. For example, the total level of mRNA may not be constant if a large portion of the genome is being remodeled as a consequence of an extreme shock (like heat shock), if a large portion of the genome is being shutdown as a consequence of a knockout in the function of a gene product essential for transcription (like Rpb1), or under specific environmental conditions (like synchronized cells used in cell cycle experiments[1]). If the total expression intensity is not constant, then the assumptions on which the method is based are invalid and valuable information is lost during the rescaling process.

Compounding the problem is the fact that it is rare for a single GeneChip to contain an array of probes for measuring the expression of an entire genome. Although the latest Affymetrix manufacturing processes do permit the entire yeast genome to be probed using a single GeneChip, the vast majority of yeast expression profiles determined to date have been measured using the Affymetrix Ye6100 GeneChip set, which require four arrays to measure the entire yeast genome. The problem is magnified when we consider the genomes of more advanced organisms such as humans that feature a much larger genome and a significant

[1]There is no *a priori* reason to believe that the total amount of mRNA present during the growth phases of the cell cycle are the same as the total amount of mRNA present during the mitotic or DNA replication phases of the cell cycle, *e.g.*

amount of alternative splicing [74, 119], circumstances under which it is unlikely a single array will be sufficient to measure a complete genome-wide expression profile. In the context of multiple arrays being used to measure the expression profile of the entire genome, there is no guarantee that the average intensity for the genes on each array can safely be assumed to be equal as this method requires. In most cases, however, this assumption is reasonable and in cases where it is not, it may be possible to develop more sophisticated methods for taking any disparity into account. Compared to the first assumption, this second assumption is relatively benign.

A second method for normalizing arrays based on all the genes in the genome is that of linear regression. Under this method, two arrays are normalized against one another by regressing the reported expressions of the genes measured on one array against the reported expressions of the same genes measured on another array. As with the average intensity normalization method above, the average intensity of expression across multiple arrays is equalized when normalizing the arrays to one another because the best fit line under linear regression is constrained to pass through the mean. The difference here is that the method does not assume that the array-level error is purely multiplicative — an additive array-level error can be modeled as well. If we were to add an extra constraint that the best fit line have zero intercept, we would get the previous result that arrays can be normalized using simple scaling factors and that the ratio of the scaling factors between two arrays is equal to the ratio of the arrays' average intensities, as before.

Nevertheless, this method suffers from maladies similar to those of the average intensity method, namely that there is no particular reason to believe that when normalizing two arrays against one another, the average intensity across arrays remains constant under widely varying experimental conditions. Moreover, when multiple arrays are required to measure genome-wide levels of expression, each array contains probes for an entirely different set of genes so it is not clear how this method proposes to normalize the multiple arrays required for a single observation of genome-wide expression.

A third, and significantly more sophisticated, method for normalizing arrays using information from all of the genes in the genome is the method proposed by Schadt, *et al.* [107]. This method is essentially a nonlinear extension of the linear regression method in which

arrays are normalized by fitting splines rather than lines to the scatterplots of reported expression levels. Though it does not constrain the average intensity to remain constant, this method assumes that a sizable portion of the genome is expressed at a comparable level across different observations under widely varying experimental conditions. Thus the suitability of this normalization method is directly related to the suitability of the assumption. In cases where the assumption is reasonable, so should the method be. In cases where the assumption is unreasonable, so should the method be. A further extension to address some of this concern is presented by Schadt, *et al.* [107, 106] wherein not all genes are used but only a subset they label *approximately invariant.* This helps to weaken the impact of the assumption when only a moderate number of genes are changing expression levels, but the method still may have some difficulty in the extreme situations cited above, namely when a large portion of the genome is being remodeled, being shut down, or changing with the cell cycle in synchronized cells. As before, the issue of normalizing the multiple arrays required to measure genome-wide levels of expression is not addressed.

2.2.2 Methods based on expression of certain specific genes

The primary effort at normalization based on the expression of certain specific genes in the genome attempts to leverage information from genes nicknamed *housekeeping genes.* For a long time it has been suggested that certain genes in the genome are expressed at constant levels in the cell under all circumstances and all environmental conditions because they are necessary for basic cellular function. In other words, the genes play housekeeping roles in the cell and as a consequence, the cell would never need to alter the expression of these genes, resulting in roughly constant levels of expression under all circumstances. While a number of such genes have been suggested, perhaps the most prominent example of a supposed housekeeping gene is the gene coding for actin, a protein responsible for cellular structure and motility. Under the assumption that actin is expressed at roughly constant levels at all times, we can control for array-level multiplicative error by computing a scaling factor for each array using reported actin levels from each array — in this case, each array is designed to include probe features to measure the abundance of actin, thereby addressing the problem of how to normalize the multiple arrays needed to measure genome-wide levels

of gene expression. The optimal scaling factor necessary for making reported expression levels comparable is simply the factor needed to compensate for any observed differences in the reported level of actin expression.

The problem with this method is that the more we understand about the regulation of gene transcription, the more we are discovering that it is very unlikely that the cell expresses any of its genes at a roughly constant level under all circumstances. In other words, the more we learn about gene expression, the less likely it seems that these supposed housekeeping genes exist as true housekeeping genes. In particular, these genes seem to be significantly down-regulated along with most other genes in the genome under conditions that necessitate massive genome remodeling.

However, if the specific conditions under which observations are made are such that housekeeping genes can reasonably be expected to be expressed at roughly constant levels, these methods can be very useful. The assumption that certain genes are expressed at roughly constant levels puts no restrictions on the levels of expression of the other genes in the genome. In contrast, the methods of the previous section place a restriction on the levels of expression of all genes in the genome, but do so only in aggregate rather than for any specific genes. Which assumption is more valid remains to be seen, though the answer is likely to depend on the particular set of conditions under which observations are made. In general, we would suspect that if housekeeping genes were in fact expressed at constant levels, that they would be more useful for normalizing under a wide range of experimental conditions, but in cases where they are not, that the methods based on information gathered from all the genes in the genome would be more robust as they are based on the measurements of many genes rather than on the measurements of only a small number of genes. We compare these methods directly later in the chapter.

2.2.3 Methods based on expression of exogenous (spiked) genes

The idea of exploiting housekeeping genes that are expressed at constant levels to normalize reported expression profiles measured on different arrays would be excellent if we could guarantee that such genes existed. However, the idea is suggestive because we can artificially introduce genes that serve the same purpose instead of relying on the existence

of such genes *in vivo*. This can be accomplished using exogenous genes, spiked into our samples as controls at constant levels across different arrays. The mRNA associated with the spiked controls are derived from exogenous genes in the sense that the genes exist in different organisms than the one of interest and are selected to have little or no sequence similarity with the genes in the organism of interest. Since the amount of mRNA associated with each of these exogenous genes is carefully prepared before addition to the extracted sample, these spiked controls provide extrinsic information that can be leveraged in the normalization process.

The disadvantage of using exogenous controls, in contrast to methods based on intrinsic measures like average intensity or housekeeping genes, is that they cannot be added early enough in the observation process to control for all sources of variation. In that sense, intrinsic controls like average intensity or housekeeping genes are probably able to provide more thorough estimates of obscuring variation. Nevertheless, since intrinsic measures are not useful under all circumstances, these exogenous spiked controls provide a perfectly sensible alternative, a claim which we justify later in the chapter.

In light of this concern, if exogenous spiked controls are only added at one stage in the sample preparation process, it is most useful if they are added to the sample as soon as possible because that enables them to compensate for as much of the variation as possible. In particular, exogenous controls are never able to compensate for variation introduced in the process before they are added (which is precisely why intrinsic measures can better normalize arrays when the assumptions on which they are based are applicable). More sophisticated addition of spiked controls might entail adding different species of spiked controls at different stages during the sample preparation and measurement processes in order to more carefully characterize how much variation is introduced at each stage along the way, but this is not commonly practiced and we do not say anything more in this dissertation about such methods.

2.2.4 Methods for normalization of non-Affymetrix expression data

It should also be noted that the problem of normalization is not one that is exclusive to Affymetrix GeneChips. Similar issues arise in making comparable the data from different

experimental observations using other measurement technology platforms, though the specific issues in those contexts are quite different. Papers by Kerr, Martin, and Churchill [70], Hughes, *et al.* [62], and Yang, Dudoit, *et al.* [124, 38] present interesting work on the normalization of data gathered on printed cDNA microarrays, for example.

2.3 Handling multiple sources of normalization information

When we use average intensity of gene expression as a source of information for normalization, it is represented by a single value. However, it is generally the case that when individual genes are used for normalization, whether they are housekeeping genes like actin or exogenous genes spiked into the sample of interest, the array contains multiple probe sets for measuring these control genes in order to be robust in the face of small amounts of additive noise in the signals. The situation is complicated further when multiple different housekeeping genes are used or multiple different species of spiked controls are added at different concentrations to the sample in order to gather even more sources of information about how the arrays should be normalized to one another, as is typically the case.

So a simple question remains: how should all this information be combined together to best determine the optimal scaling factor for each array? Early published results [61] used a simple method for doing this in the context of exogenous spiked controls by computing a scaling factor based on the arithmetic mean of the scaling factors that each species of spiked control suggested. For example, if one spiked control suggested that an array should be scaled by a factor of 2 and another spiked control suggested that the array should be scaled by a factor of 3, then the array would be scaled by a factor of 2.5. The problem with this method is apparent when more a more striking example is used: if one spiked control suggested that an array should be scaled by a factor of 2 and another spiked control suggested that the array should be scaled by a factor of 0.5, then the array would be scaled by a factor of 1.25, instead of the more intuitive result that the array should not be scaled at all (or, equivalently, scaled by a factor of 1). Based solely on intuition, it seems some type of geometric mean would be more appropriate. We shall see in Section 2.4.1 that the solution we derive formally is a weighted geometric mean and thus has the property that

in the example above, the array would be scaled by a factor of 1 (provided the two spiked controls were equally trustworthy, a concept to be elucidated in Section 2.4.1).

We now derive a principled method for combining these multiple sources of normalization information into a single estimate for the optimal scaling factor for an array. Throughout this presentation we assume that we are dealing with exogenous spiked controls, but the method is equally applicable when computing the optimal scaling factor in the context of multiple probes for housekeeping genes on each array.

2.4 Mathematical formulation of normalization problem

Let the reported expression levels of M spiked controls from a set of N Affymetrix GeneChips be denoted x_{ij} where i indexes the spiked controls and ranges from 1 to M, while j indexes the arrays and ranges from 1 to N. The reported spiked control expression levels form an $M \times N$ matrix as shown:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \tag{2.1}$$

We assume that a fixed amount of each spiked control is added to all arrays. We denote the true level of expression for each spiked control i to be m_i, for all settings of j. We allow m_i to be different for each i even in the event that multiple probe sets on the array are probing for the same control species because different probes may bind their target with different efficiencies. One could consider modeling this structure in the data explicitly. This is not likely to provide significant advantage as there does not seem to be much corresponding structure in the multiple probe sets used to probe for a single control species, but the final determination should be based on empirical results, which we do not develop here.

In a world without error, x_{ij} would be equal to m_i for all i and all j, but empirically this is not observed to be the case. Figure 2-1 shows clearly that the reported expression levels for the spiked controls vary considerably (Figure 2-2 shows that the situation is similar for

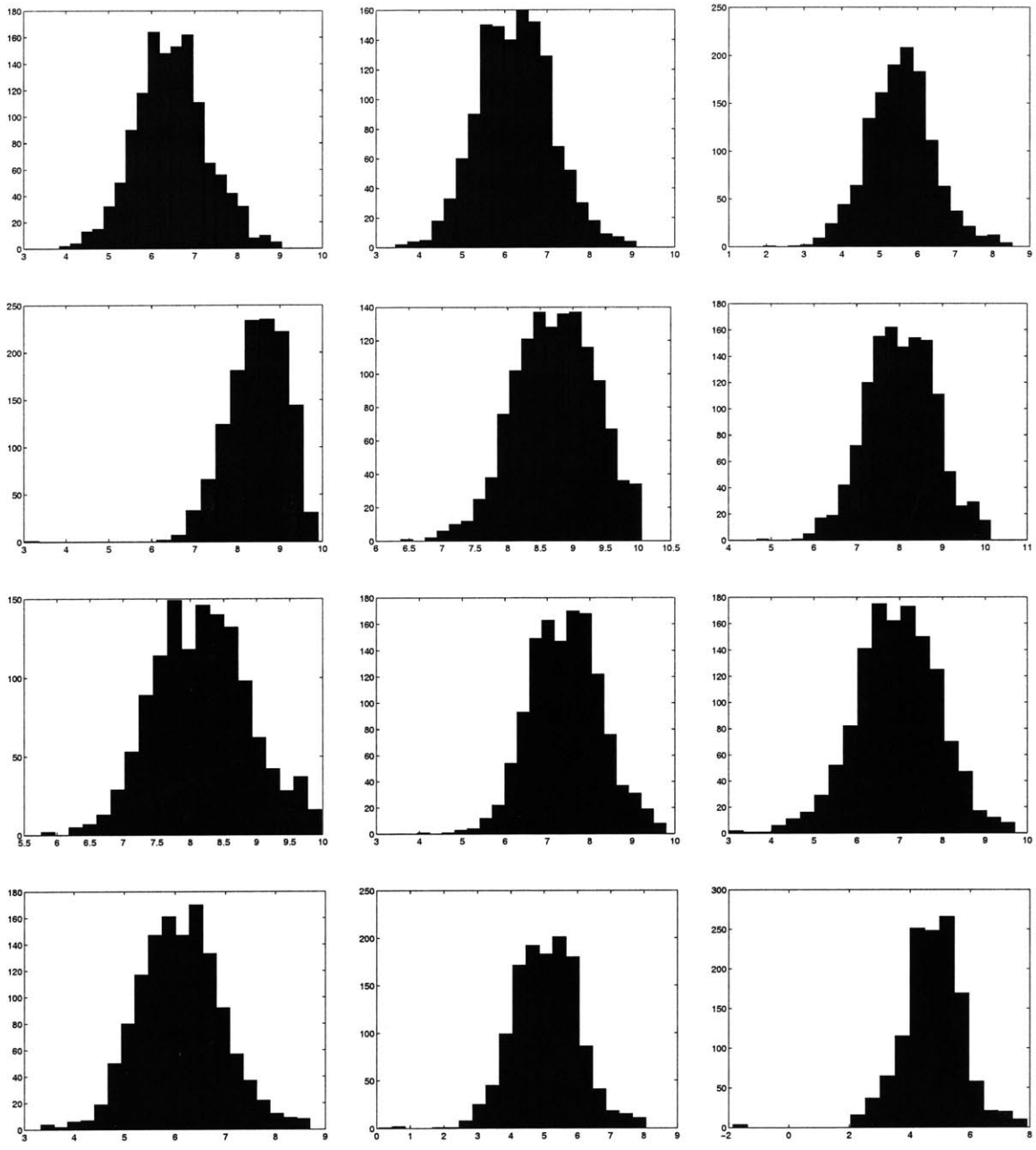

Figure 2-1: *Histograms indicating the level of expression for twelve different spiked control probes over 1280 Affymetrix GeneChip observations. All data is presented on a log scale. While the level of reported expression should be constant under ideal conditions of no noise, a significant range of reported expression is observed in practice.*

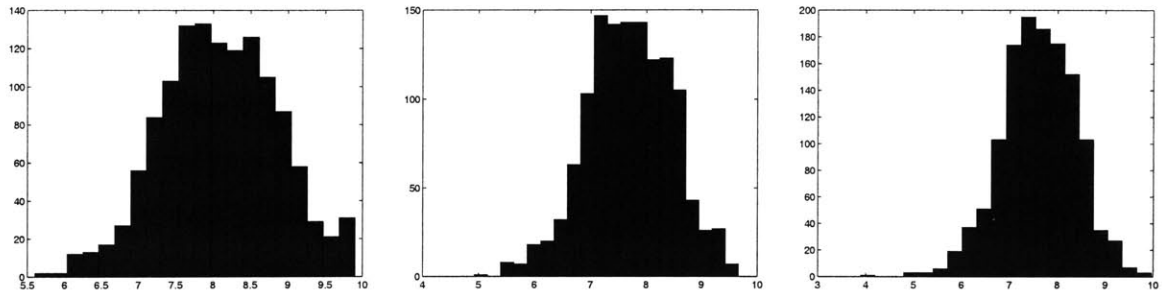

Figure 2-2: *Histograms indicating the level of expression for three different actin probes over 1280 Affymetrix GeneChip observations. All data is presented on a log scale. While the level of reported expression should be constant under ideal conditions of no noise, a significant range of reported expression is observed in practice.*

the actin probe sets on each array). The reported expression level for each spiked control has a number of sources of variation, as discussed in Section 2.1. For example, the level reported depends on the actual quantity of control material pipetted into the sample and the actual amount of sample-control mixture injected into the GeneChip. The manufacture of the array and the density of the probes present on the array introduce more variation. The temperature of hybridization and variations in the buffer makeup also contribute to differences in reported levels. Because each of these sources of error is multiplicative, we assume that the true expression levels are modified by a multiplicative factor r_j which may (or may not) be different for each array j and also by a random multiplicative error e_{ij} for each i and j. That the error is primarily multiplicative can be seen from the histograms in Figure 2-1 and Figure 2-2 which are reasonably symmetric when plotted on a log scale and have similar standard deviations. Under this assumption of purely multiplicative error, we have in formal terms:

$$x_{ij} = m_i \times r_j \times e_{ij} \tag{2.2}$$

where the e_{ij} factors are assumed to be fairly small and close to 1. For convenience, we transform this equation logarithmically so that the multiplicative errors become additive. Let $y_{ij} = \log(x_{ij})$, $\mu_i = \log(m_i)$, $\rho_j = \log(r_j)$, and $\epsilon_{ij} = \log(e_{ij})$ for all i and all j. The

matrix of reported spiked controls after transformation becomes:

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{M1} & y_{M2} & \cdots & y_{MN} \end{bmatrix} \tag{2.3}$$

and the equation describing the error model becomes:

$$y_{ij} = \mu_i + \rho_j + \epsilon_{ij} \tag{2.4}$$

We assume that ϵ_{ij} is randomly distributed and is drawn from a central normal distribution with variance σ_i^2. We permit the variance σ_i^2 to be different for each spiked control i to account for the fact that different probes on Affymetrix arrays may have different underlying variances in terms of their response to targets. With these assumptions in place, we have a model describing how the log reported expression levels for the spiked controls are distributed:

$$y_{ij} \sim N(\mu_i + \rho_j, \sigma_i^2) \tag{2.5}$$

2.4.1 Maximum likelihood estimation of normalization parameters

With a model describing how the log reported expression levels for the spiked controls are distributed, we can use maximum likelihood estimation to derive optimal values for the scaling factors necessary to properly normalize each Affymetrix GeneChip. First, we form the log-likelihood $\mathcal{L}$ for observing the data y_{ij} under the assumption of normality outlined in the previous section:

$$\mathcal{L} = \log\left(\prod_{i=1}^{M}\prod_{j=1}^{N} P(y_{ij}|\mu_i, \rho_j, \sigma_i^2)\right) \tag{2.6}$$

$$= -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\log(2\pi\sigma_i^2) + \frac{(y_{ij} - \mu_i - \rho_j)^2}{\sigma_i^2}\right) \tag{2.7}$$

Then, we solve for the values of μ_i, ρ_j, and σ_i^2 that maximize the log-likelihood of observing the data:

$$(\hat{\mu}_i, \hat{\rho}_j, \hat{\sigma_i^2}) = \underset{\mu_i, \rho_j, \sigma_i^2}{\arg\max}\, \mathcal{L} \tag{2.8}$$

Setting $\frac{\partial \mathcal{L}}{\partial \mu_i} = 0$, $\frac{\partial \mathcal{L}}{\partial \rho_j} = 0$, and $\frac{\partial \mathcal{L}}{\partial \sigma_i^2} = 0$ in turn yields estimates for the values of the parameters in question:

$$\hat{\mu}_i = \frac{1}{N} \sum_{j=1}^{N} (y_{ij} - \hat{\rho}_j) \tag{2.9}$$

$$\hat{\rho}_j = \frac{\sum_{i=1}^{M} (\hat{\sigma_i^2})^{-1} (y_{ij} - \hat{\mu}_i)}{\sum_{i=1}^{M} (\hat{\sigma_i^2})^{-1}} \tag{2.10}$$

$$\hat{\sigma_i^2} = \frac{1}{N} \sum_{j=1}^{N} (y_{ij} - \hat{\mu}_i - \hat{\rho}_j)^2 \tag{2.11}$$

As the estimates of the $2M+N$ unknown parameters are all coupled, it is necessary to iterate this solution until convergence, which can be done in rounds. Each round monotonically increases the likelihood of the observed values y_{ij} under the model. Since the likelihood is bounded from above by one, this series converges.

Once the iteration of estimates has converged, the N estimates for ρ_j that emerge can be used to derive optimal scaling factors for the N arrays. We define the optimal scaling factor for array j to be $\hat{s_j}$ and compute it as shown:

$$\hat{s_j} \equiv \frac{1}{\hat{r_j}} = \mathrm{e}^{-\hat{\rho}_j} = \prod_{i=1}^{M} \left(\frac{\hat{m}_i}{x_{ij}} \right)^{\hat{w}_i} \tag{2.12}$$

where $\hat{m}_i = \mathrm{e}^{\hat{\mu}_i}$ and we have defined the weights $\hat{w}_i$ to be:

$$\hat{w}_i \equiv \frac{(\hat{\sigma_i^2})^{-1}}{\sum_{i=1}^{M} (\hat{\sigma_i^2})^{-1}} \tag{2.13}$$

The optimal scaling factors are simply weighted geometric means of the ratios between $\hat{m}_i$

and x_{ij}, as might be expected, where the weight associated with each spiked control is inversely proportional to the estimated variance for that spiked control.

It should be noted that the model shown in (2.5) is not technically identifiable as presented because the μ_i's and the ρ_j's are confounded (up to a constant). Adding a constraint such as $\sum \rho_j = 0$ makes the model identifiable, as does the inclusion of a constant fixed effect and an additional constraint that $\sum \mu_i = 0$. The latter is probably less helpful because while it makes sense to constrain the ρ_j's given that we believe they are mean zero, doing so for the μ_i's is probably unnecessary since they are certainly not mean zero.

2.4.2 Maximum a posteriori estimation of normalization parameters

The fact that the estimates of μ_i, ρ_j, and σ_i^2 are computed iteratively can lead to a problem: as the optimal scaling factors are weighted geometric means where the weights are inversely proportional to the estimated variances, the information source with the least variance is weighted increasingly more with each iteration until, in the limit, a single information source can become scaled to its mean while the other sources of information are essentially ignored. This happens because the variance of the dominant control approaches zero as it is rescaled uniformly to its mean.

It is unlikely that the previously mentioned lack of identifiability is the culprit behind the failure of ML estimation. Since the lack of identifiability is only up to a constant, such a constant would have no influence whatsoever on the variance and it would be expected that under the appropriate conditions, perverse trivial normalization would be observed regardless of whether we impose a constraint on the ρ_j's or not. Without a constraint, the actual $\sum \rho_j$ depends on the initial setting of the ρ_j's before applying the coupled updates. For the estimates of ρ_j we generate, for example, the mean of the $\hat{\rho}_j$'s is around 10^{-14}, which is already very close to zero, and yet we still observe the perverse trivial normalization. Therefore, adding such a constraint does not fix the problem.

To avoid this pathological behavior and leverage the information about optimal scaling factors present in each of the multiple sources of information rather than simply one such source of control information, we modify the solution to incorporate a regularization term for the variances. This is accomplished by establishing prior distributions over possible

values of the parameters and then estimating the maximum a posteriori (MAP) values of those parameters. In our context, we need only establish a prior for the variances σ_i^2; we can assume a flat prior over the means μ_i and log ratios ρ_j since we do not need regularization terms for these parameters. The assumption of flat priors for μ_i and ρ_j means that the prior terms for these parameters can be set to unity, and therefore the MAP updates for $\hat{\mu}_i$ and $\hat{\rho}_j$ are identical to the ML updates for these parameters.[2] Formally, we seek to maximize the posterior probability distribution for the parameters given the reported expression levels:

$$P(\mu_i, \rho_j, \sigma_i^2 | y_{ij}) \propto P(y_{ij} | \mu_i, \rho_j, \sigma_i^2) \cdot P(\sigma_i^2) \tag{2.14}$$

The likelihood term of the previous section reappears in this Bayesian formulation but is now accompanied by the prior distribution over the variances, serving as a regularization term.

As the likelihood is normally distributed, we assume a conjugate form for the prior over the variances, namely, a Wishart distribution. If we further assume that our prior belief about the variances is uninformative in the sense that we have no reason to believe, *a priori*, that the value of σ_i^2 should be different for one value of i than for any other, the multidimensional prior takes a relatively simple factorized form:

$$P(\sigma_i^2) = \prod_{i=1}^{M} C(\alpha, t) \left(\frac{1}{\sigma_i^2} \right)^{\frac{\alpha-3}{2}} \mathrm{e}^{-\frac{t}{2\sigma_i^2}} \tag{2.15}$$

where t represents the mode of the prior distribution, α represents the degree of confidence in the prior (relative to the quantity of available data), and $C(\alpha, t)$ is a normalizing constant dependent on α and t. Having defined the likelihood term and the prior term, we can proceed to maximize the *a posteriori* probability by taking partial derivatives with respect to μ_i, ρ_j, and σ_i^2 and setting them equal to zero once again, yielding estimates for the values of the parameters in question. This results in the same equations for $\hat{\mu}_i$ and $\hat{\rho}_j$ as given

[2] Although the form of the updates is the same, the actual values of the estimates may be different as the values of $\hat{\mu}_i$ and $\hat{\rho}_j$ depend on the values of $\hat{\sigma_i^2}$.

above in (2.9) and (2.10), but a new equation for $\hat{\sigma_i^2}$:

$$\hat{\sigma_i^2} = \frac{\sum_{j=1}^{N}(y_{ij} - \hat{\mu}_i - \hat{\rho}_j)^2 + t}{N + \alpha - 3} \tag{2.16}$$

A non-zero prior setting for t prevents the estimates of σ_i^2 from converging to zero for any i during the iteration process (except perhaps in the limit of infinite data).

2.5 Maximum a posteriori normalization results

A set of 320 samples of unsynchronized *Saccharomyces cerevisiae* populations of varying genotype were observed under a diversity of experimental conditions. The set of samples ranges widely but consists primarily of observations of various wild-type and mutant *S. cerevisiae* strains made under a variety of environmental conditions including exposure to different nutritive media as well as exposure to stresses like heat, oxidative species, excessive acidity, and excessive alkalinity.

Whole-genome expression data for each of these 320 observations was collected using Affymetrix Ye6100 GeneChips. These GeneChips are 50-micron Affymetrix chips and consequently, four chips are required to measure the expression of all 6135 genes in the *S. cerevisiae* genome. Thus, a total of 1280 GeneChips were used in collecting this data.

Four different control species (DapX, LysX, PheX, and ThrX) are spiked into the extracted mRNA samples before hybridization. Each Affymetrix Ye6100 GeneChip has a set of three probes for each of the spiked control species. One probe contains features binding near the $3'$ end of the target, one contains features binding near the middle of the target, and one contains features binding near the $5'$ end of the target. Thus, a total of 12 spiked control expression levels are reported for each GeneChip. We use the 12×1280 array of reported spiked control expression levels to produce estimates of the optimal scaling factors for the 1280 GeneChips using the ML and MAP estimation methods shown above. In both cases, results are nearly identical, though we display results below for only the MAP estimates because of their regularization properties. We set $\alpha = 3$ and $t = 1$ in our estimation, but varying these parameters by an order of magnitude has little effect on the results (not shown).

Before proceeding, we should mention how we deal with negative reported levels of expression. Negative levels of expression are biologically nonsensical, and we avoid the computational difficulty associated with taking logarithms of negative values by addressing this problem. Although some researchers have chosen to apply a flooring function to their data, the lack of smoothness associated with such a function is disconcerting. Instead, we employ a *compression* function which smoothly compresses the interval $(-\infty, \infty)$ into the interval $(0, \infty)$. The compression function we use is given by:

$$c(x) = \begin{cases} x, & x \in [40, \infty) \\ \frac{1}{160}x^2 + \frac{1}{2}x + 10, & x \in (-35, 40) \\ 0.15625, & x \in (-\infty, -35] \end{cases} \tag{2.17}$$

The function c is twice differentiable, acting as the identity function for values above 40, a flooring function for values below -35, and a smooth quadratic interpolation in between. This compression function can be viewed as a monotonic approximation to the Bayesian estimate of true (necessarily non-negative) expression level from reported expression level developed by Dror, *et al.* [37].

Figure 2-3 is a scatterplot of the estimated standard deviation of log expression levels σ_i versus the estimated mean of log expression levels μ_i for the 12 spiked controls added to the 1280 Affymetrix Ye6100 GeneChips. The estimated standard deviations are generally relatively low and constant, with the exception of the first point. The greater estimated standard deviation associated with the point corresponding to the lowest average level of expression suggests that additive error may be playing a significant role there. Although additive error tends to be swamped by multiplicative error for large levels of expression, it should be incorporated in a more complicated model in order to adequately capture sources of variation when expression levels are low. We discuss this possibility in Section 2.7.

Figure 2-4 is a normal probability plot of the estimated log ratios ρ_j. The plot reveals that the estimated log ratios are roughly normally distributed. Recall that we made no assumptions about the form of the distribution of ρ_j in our modeling.

Figure 2-5 contains both a histogram and a normal probability plot of the residual errors ϵ_{ij}, which represent variations in y_{ij} that remain unexplained after optimally estimating μ_i

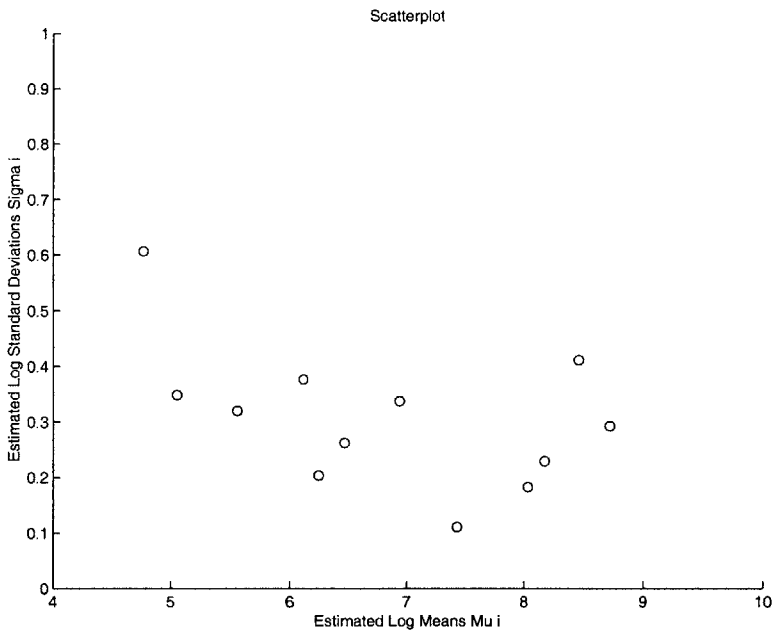

Figure 2-3: *Scatterplot of estimated standard deviation of log expression levels σ_i versus estimated mean of log expression levels μ_i for 12 spiked controls. The estimated standard deviations are generally relatively low and constant, with the exception of the first point. The greater estimated standard deviation associated with the point corresponding to the lowest average level of expression suggests that additive error may be playing a significant role.*

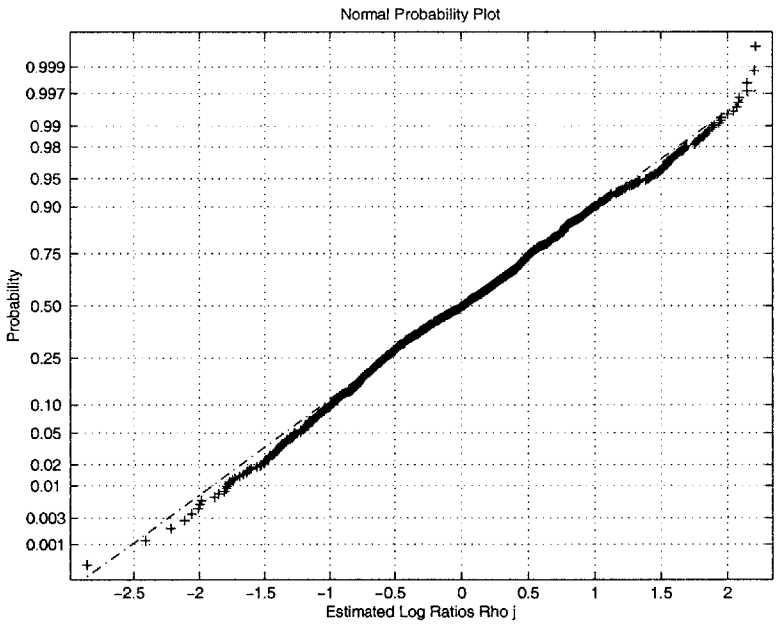

Figure 2-4: *Normal probability plot of estimated log ratios ρ_j. The plot reveals that the estimated log ratios are roughly normally distributed. We made no assumptions about the form of the distribution of ρ_j in our modeling.*

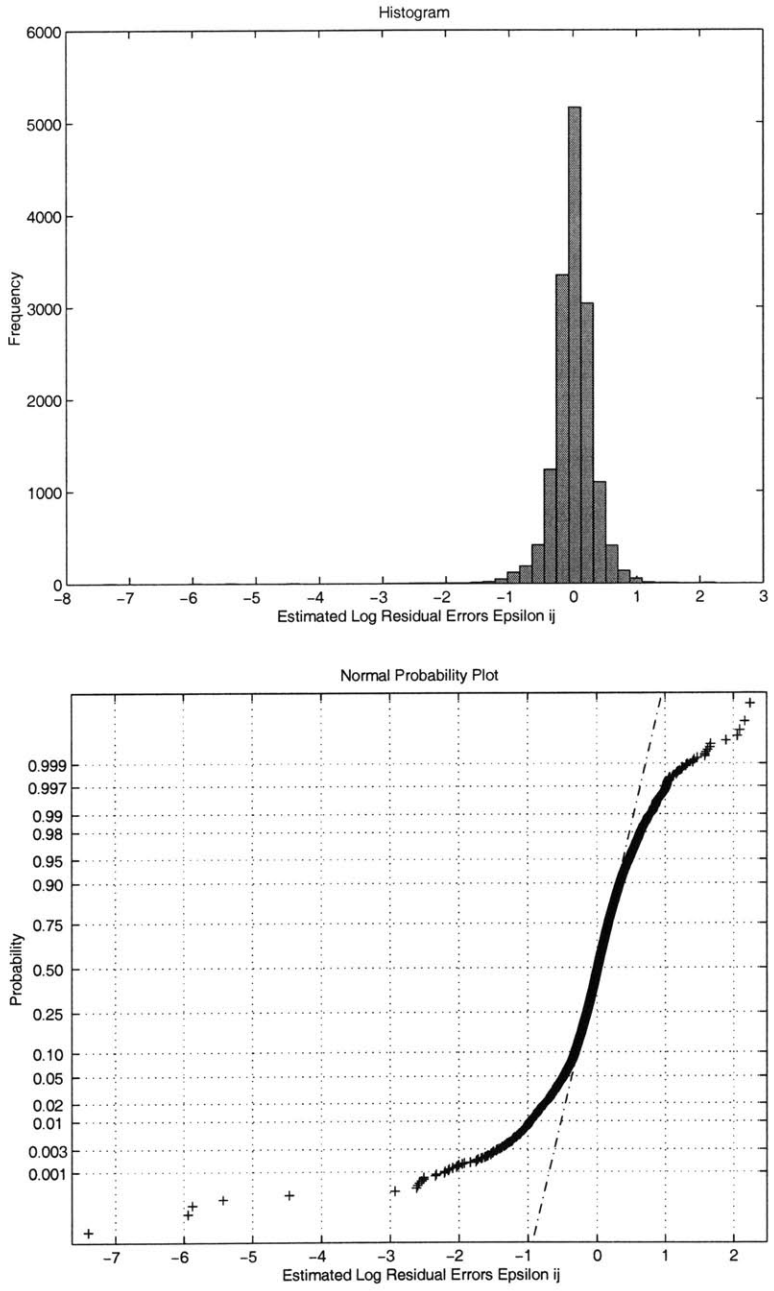

Figure 2-5: *Histogram and normal probability plot of residual errors ϵ_{ij}. The histogram plot on the top appears normal at first glance but the normal probability plot of the same data on the bottom reveals that the distribution is actually fairly heavy-tailed (even excluding the six obvious outliers).*

and ρ_j. The histogram plot on the top appears normal at first glance but the normal probability plot of the same data on the bottom reveals that the distribution is actually fairly heavy-tailed (even excluding the six obvious outliers). We discuss this in greater detail in Section 2.7.

Once we have computed the estimates of μ_i, ρ_j, and σ_i^2, we can use the estimates of ρ_j to compute the optimal scaling factors for the 320 arrays. Figure 2-6 provides scatterplots of the standard deviation of log expression level versus the mean of log expression level for the 6135 yeast genes with probes on the Ye6100 Affymetrix arrays. The plot in the upper left represents unnormalized expression levels from 320 experiments over widely varying experimental conditions. The plot in the upper right represents unnormalized expression levels from 8 wild-type experiments with constant experimental conditions. The lower plots are the same as the corresponding upper plots but are computed from normalized expression levels. Considered column-wise, the plots in Figure 2-6 reveal that the normalization process is successful in reducing the overall variation in the data. In the case of the 320 experiments, the average standard deviation drops from 0.97 to 0.83, while in the case of the 8 wild-type experiments, the average standard deviation drops from 0.73 to 0.54. The fact that points on each plot with low average levels of expression tend to have a much greater standard deviation suggests, consistent with our observations in Figure 2-3, that additive error is playing a significant role at lower levels of expression. The bands that appear in both plots on the right side of Figure 2-6 are integer and flooring effects that arise when genes are expressed at very low levels (frequently reported to be negative, *e.g.*).

The information contained in the plots on the right side of Figure 2-6 can be visualized another way by considering a scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene before and after normalization across the 8 wild-type replicate experiments. This scatterplot is shown in Figure 2-7. Such a scatterplot is essentially the same as plotting the two sets of standard deviations against one another and then rotating the figure by $-\pi/4$ so that the 45°-line becomes the x-axis. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments increased after normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments decreased after nor-

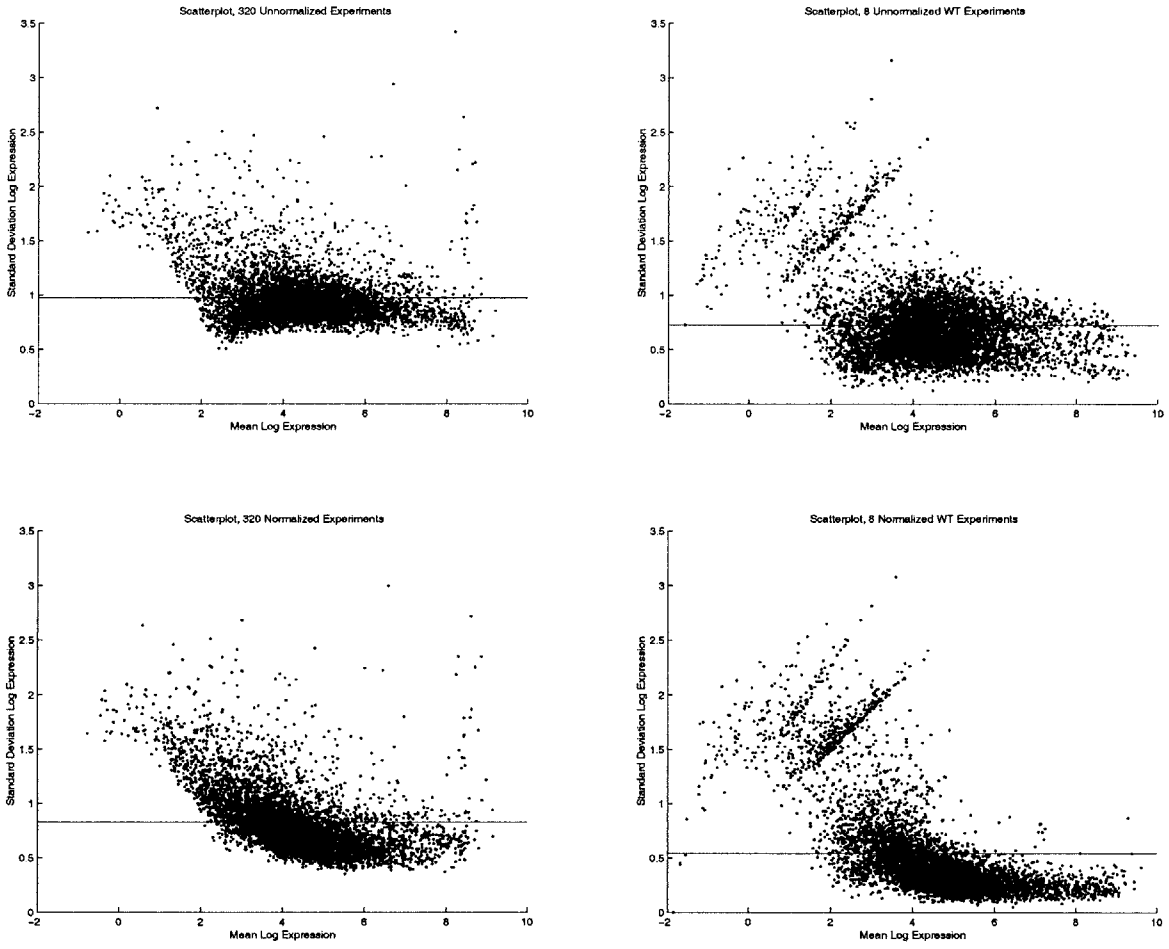

Figure 2-6: *Scatterplots of standard deviation of log expression level versus mean of log expression level for 6135 yeast genes. The panel in the upper left represents unnormalized expression levels from 320 experiments over widely varying experimental conditions. The panel in the upper right represents unnormalized expression levels from 8 wild type experiments with constant experimental conditions. The lower panels are the same as the corresponding upper panels but are computed from normalized expression levels.*

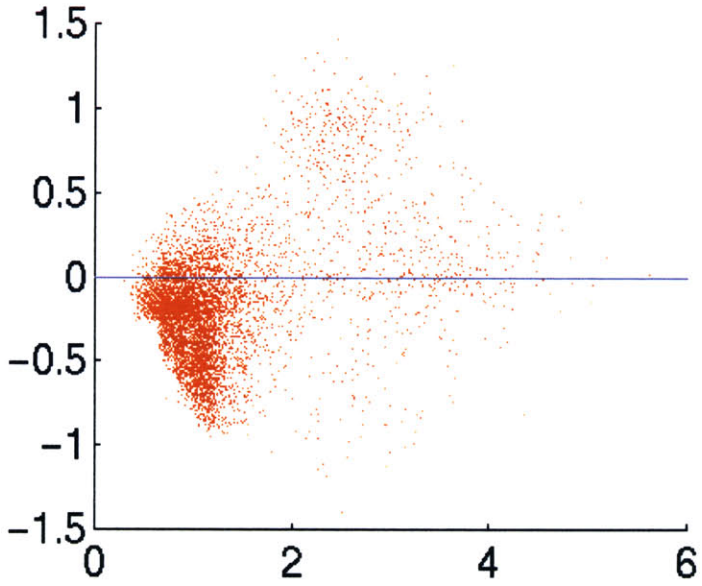

Figure 2-7: *Scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene before and after MAP spiked control normalization across the 8 wild-type replicate experiments. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments increased after normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments decreased after normalization.*

malization. Points farther to the right indicate more highly variable genes and as such, tend to correspond to genes with very low levels of expression being corrupted by additive noise (*cf.* Figure 2-6). This kind of plot indicates how uniformly genes' standard deviations are changed as a result of normalization. We see that the vast bulk of genes have their standard deviations reduced during normalization, especially when highly variable genes are excluded from consideration.

2.6 Comparison of normalization methods

In addition to this normalization method based on MAP estimation of optimal scaling factors from spiked control expression, a number of other methods were also implemented. Of these, results from twelve of the most effective are shown in Figure 2-8 (including the MAP spiked control method). All plots depict a scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene before and after normalization across the 8 wild-type replicate experiments, as in Figure 2-7 of the previous section.

The plots, from left to right and then top to bottom, display the following normalization methods: spiked controls (arithmetic mean of scaling factors), spiked controls (ML estimation), spiked controls (MAP estimation), actin controls (arithmetic mean of scaling factors), actin controls (ML estimation), actin controls (MAP estimation), arithmetic mean of actin controls, geometric mean of actin controls, spiked controls plus actin controls (MAP estimation), arithmetic mean of intensity, median of intensity, spiked controls plus actin controls plus average intensity (MAP estimation). It is apparent from the figures that at a high-level, they all seem to do approximately the same thing. That is, each of the methods is in general agreement as to how to best normalize these 8 wild-type experiments. We shall see later in this section that this is not always the case when we consider experiments in which large portions of the genome are being remodeled.

In order to examine the differences between these methods more closely, we can produce scatterplots comparing different types of normalization with one another rather than with unnormalized data. It is important when doing so to ensure that each normalization method

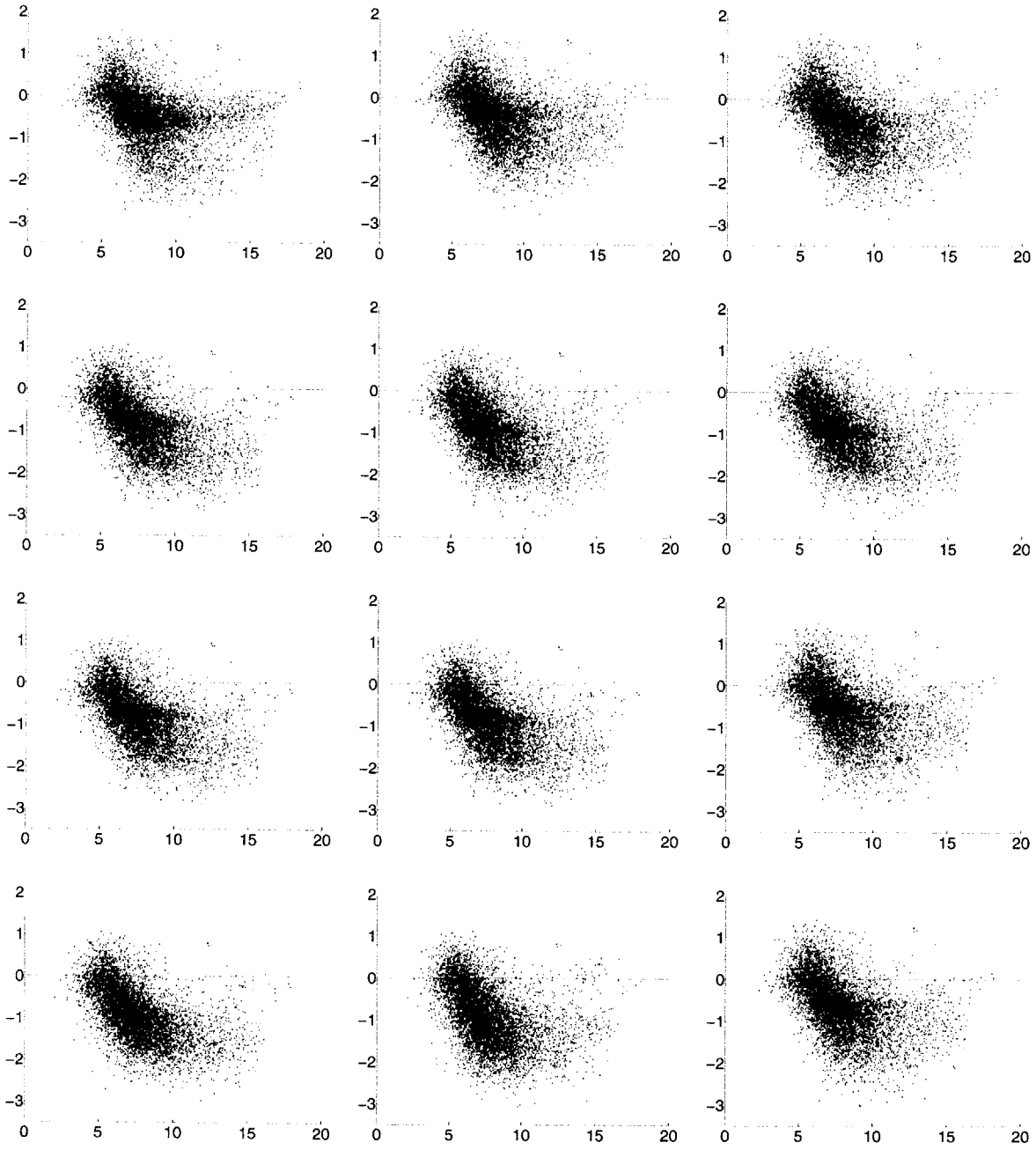

Figure 2-8: *Scatterplots of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene before and after normalization across the 8 wild-type replicate experiments. Each plot presents a different normalization method as described in the text. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments increased after normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments decreased after normalization.*

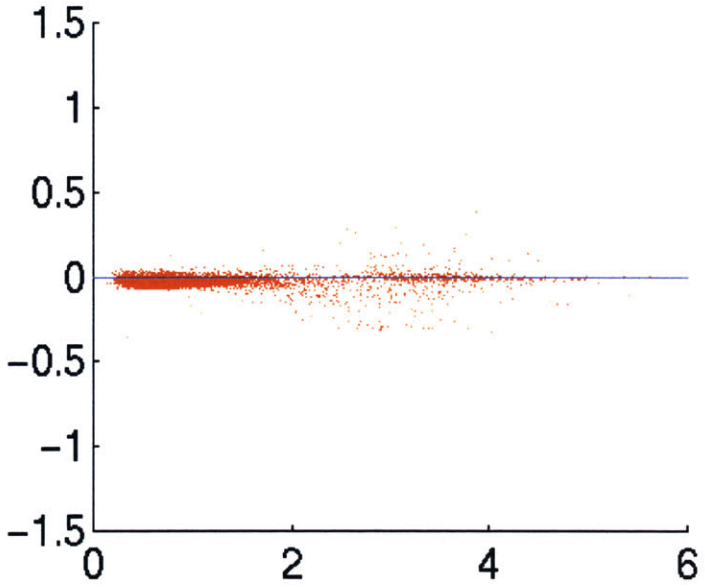

Figure 2-9: *Scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene after MAP spiked control and after ML spiked control normalization across the 8 wild-type replicate experiments. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was higher after MAP spiked control normalization than after ML spiked control normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was lower after MAP spiked control normalization than after ML spiked control normalization.*

is comparable; for instance, if one normalization method simply halved every observation then the computed standard deviations would uniformly be half of what they were before normalization, a misleading result. We could use the coefficient of variance (CV) as a substitute for the standard deviation in order to compensate for this bias, but a better solution is to prevent the bias in the first place. We do this by requiring every normalization method to scale all observations by one on average. In other words, the geometric mean of all scaling factors across all experiments should be unity.[3]

As an example illustrating how different normalization methods can be compared directly, consider Figure 2-9 which shows the difference between using MAP estimation and

[3]This is equivalent to requiring that the total log expression across all experiments remain unchanged as a consequence of normalization and is also equivalent to requiring that $\sum \rho_j = 0$, as discussed above.

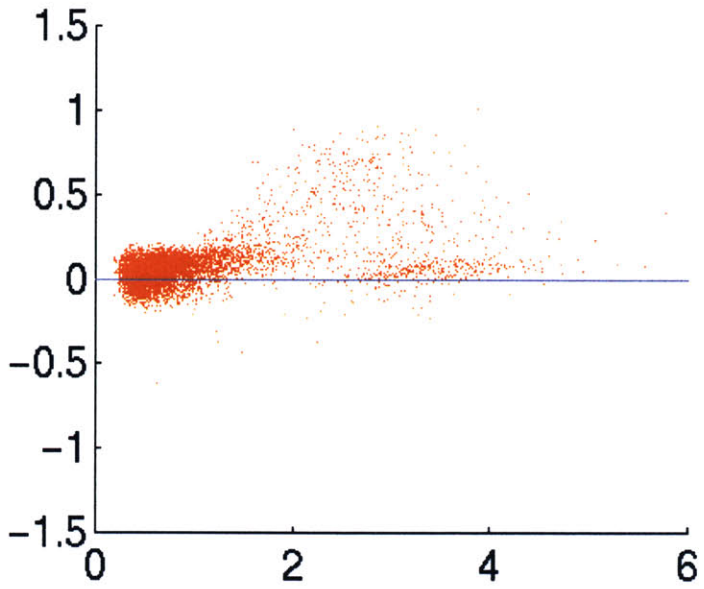

Figure 2-10: *Scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene after MAP spiked control and after MAP actin normalization across the 8 wild-type replicate experiments. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was higher after MAP spiked control normalization than after MAP actin normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was lower after MAP spiked control normalization than after MAP actin normalization.*

ML estimation of scaling factors from spiked control expression. The figure demonstrates that the difference between the methods is slight, but that MAP tends to produce lower standard deviations on the whole than ML for these 8 wild-type experiments.

Two other interesting comparisons are between MAP estimated scaling factors computed from spiked controls on the one hand, and MAP estimated scaling factors computed from actin controls or scaling factors computed from average intensity on the other hand. Figures 2-10 and 2-11 present these comparisons. The figures demonstrate that while the three methods perform roughly comparably at a high-level (as shown in Figure 2-8), when we look closely the methods based on actin controls and average intensity perform better than those based on spiked controls for these 8 wild-type replicate experiments. Figure 2-12 shows that the method based on average intensity seems to slightly outperform the method

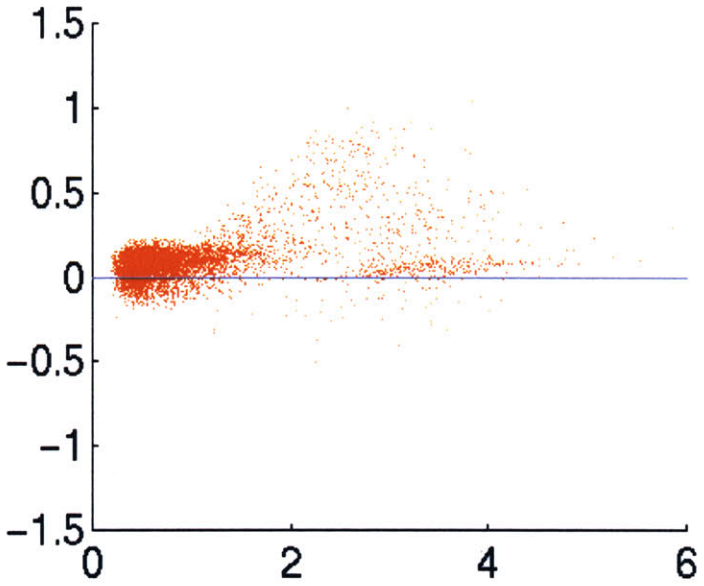

Figure 2-11: *Scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene after MAP spiked control and after average intensity normalization across the 8 wild-type replicate experiments. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was higher after MAP spiked control normalization than after average intensity normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was lower after MAP spiked control normalization than after average intensity normalization.*

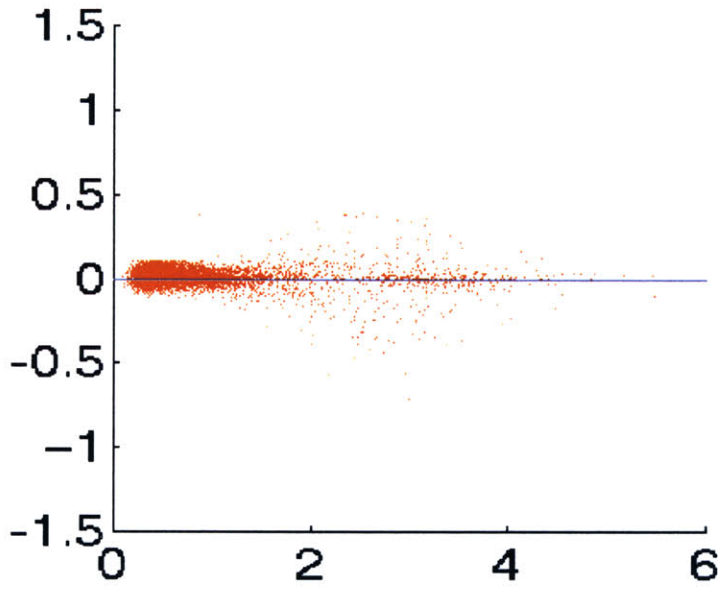

Figure 2-12: *Scatterplot of the difference in standard deviations for each gene as a function of the sum of standard deviations for each gene after MAP actin and after average intensity normalization across the 8 wild-type replicate experiments. Points above the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was higher after MAP actin normalization than after average intensity normalization. Points below the line indicate genes whose standard deviation across the 8 wild-type replicate experiments was lower after MAP actin normalization than after average intensity normalization.*

based on actin controls for these 8 wild-type replicate experiments.

That this is true is to be expected from our earlier discussion in Section 2.2.3 in which we argued that intrinsic measures would be able to control for more noise than extrinsic measures, when applicable. And in comparing the method based on average intensity to the method based on actin controls, it makes sense that the former would be a little more robust because it is based on the expression of about 1500 genes on each array rather than the expression of a single gene (measured with three different probe sets). Using more genes as average intensity normalization does tends to reduce the additive noise that sometimes clouds the picture. So why not always normalize based on intrinsic measures like actin or, even better, average intensity?

The problem with intrinsic measures, as hinted in Section 2.2.3, is that they are suspect when large portions of the genome are being remodeled. If the assumptions on which these normalization methods are based do not hold, then normalizing expression values with these methods severely corrupts the data. As evidence of this, consider what happens when we look no longer only at 8 wild-type replicate experiments (in which nothing interesting is happening between different experiments) but instead consider a collection of experiments in which a large portion of the genome is down-regulated as a consequence of either Rpb1, Srb4, or Kin28 functional deletions. In Figure 2-13, we plot the estimated log scaling factors for the 32 arrays associated with 8 wild-type replicates and then the estimated log scaling factors for the 20 arrays associated with 5 experiments with Rpb1, Srb4, or Kin28 functional deletions. As the plots show, although all three methods (MAP spiked control, MAP actin, and average intensity) generally agree as to the best scaling factors to use in the wild-type replicate experiments as indicated by the cluster of points along each of the 45° lines, the methods based on intrinsic controls both suddenly suggest scaling factors that are $e^{1.5} \approx 4.5$ times higher when most of the genome has been down-regulated. That the average intensity and level of actin expression have decreased by this factor relative to spiked control expression in these experiments is evidence of the severe corruption that can be introduced if we rely on purely intrinsic measures for normalizing arrays across a wide range of experimental conditions. That the level of actin expression is down-regulated as severely as the average level of gene expression in these experiments is evidence that actin

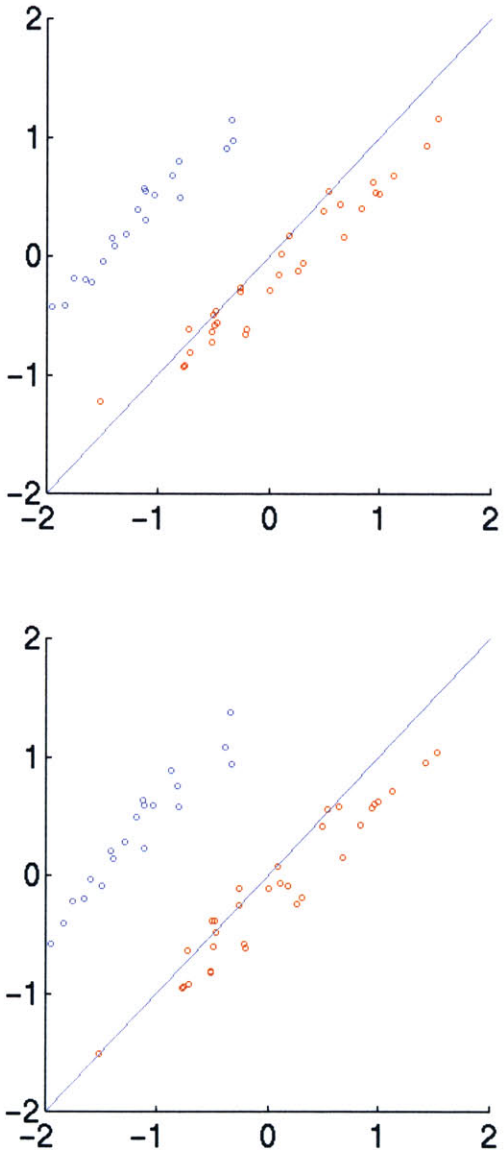

Figure 2-13: *Scatterplots of estimated log scaling factors for the 32 arrays associated with 8 wild-type replicate experiments and for the 20 arrays associated with 5 Rpb1, Srb4, or Kin28 functional deletion experiments. The plot on the top compares the estimated log scaling factors using average intensity normalization versus MAP spiked control normalization. The plot on the bottom compares the estimated log scaling factors using MAP actin normalization versus MAP spiked control normalization. In both plots, the points in red clustered along the* 45° *line come from the wild-type replicate experiments and the points in blue clustered well above the* 45° *line come from the Rpb1, Srb4, or Kin28 functional deletion experiments.*

is not immune to the general down-regulation that most of the genome is experiencing and thus is not a true housekeeping gene.

In conclusion, if only a small number of experiments need to be made comparable to one another and very little of the genome is expected to be changing, then methods based on intrinsic measures such as average intensity or actin controls should be considered. However, under general circumstances and when a number of experiments from different conditions need to all be made comparable with one another, MAP estimates of normalization factors computed from spiked controls are likely to perform best.

2.7 Discussion

In order for data from genomic expression arrays to be comparable, it is necessary that we understand the different sources of variation present in reported gene expression levels. To effectively separate the interesting variation in reported expression levels from the obscuring variation, we need statistically sound methods for deriving estimates under a variety of experimental conditions for the levels of gene expression that include interesting variation but exclude, or otherwise account for, obscuring variation.

In this chapter of the dissertation, we carefully characterize the different sources of variation present in reported gene expression levels and conclude that most error is array-level and multiplicative in nature. In the context of Affymetrix GeneChips with spiked control probes, we present a model for explaining observed expression levels under the assumption of array-level multiplicative error. We make no assumptions regarding the distributions of the scaling factors applied to each array, but assume that the log residual errors are normally distributed with a possibly different variance for each spiked control. Under these assumptions, we develop maximum likelihood (ML) and maximum a posteriori (MAP) estimates of the unknown parameters and use these estimates to compute optimal scaling factors for subsequent array normalization.

A number of interesting directions exist for extending this work. First, the formulation of our initial model is fairly simple in that it is entirely multiplicative and does not incorporate enough terms to adequately model all the sources of variation present in reported expression

levels. A more sophisticated model would consider both additive and multiplicative effects, as well as more complicated interaction terms. The problem, of course, is that most of the additive error is not likely to be additive at the array level but additive at the feature level, in which case it becomes inestimable without repeated observations. One possible way to circumvent this would be to use feature-level information from the 20 perfect match and 20 mismatch features for each gene [77] though Affymetrix currently forbids this practice. A less than ideal solution would be to estimate the general additive uncertainty associated with all probes. In this scenario, we could approximate the range of the underlying expression values, but we would have little way of estimating the best compensatory offset for the specific additive noise associated with each of these values.

Second, the error residuals ϵ_{ij} are clearly not normal as postulated in the context of our initial model. We could consider alternative descriptions of the distribution of these residuals, but the non-normality may be another indication of the simplicity of the model mentioned above. It is possible that a more sophisticated model would result in error residuals that are distributed roughly normally, thus correcting this problem simultaneously with the first.

Third, although the characterization of different sources of variation presented in Section 2.1 is applicable to all array technologies, the specific model postulated in this chapter of the dissertation is intended only for data gathered on Affymetrix GeneChips and not printed cDNA arrays that employ competitive hybridization. However, the methodology is general and the ideas should be useful in other settings with suitable modification. Moreover, we are in the process of developing methods for making data from Affymetrix GeneChips comparable with data from printed cDNA arrays, enabling the comparison of data across technology platforms. We discuss some of this work in greater detail in Section 8.2.1 of Chapter 8.

Chapter 3

Discretization of genomic expression data

Assume we have normalized genome expression profiles (transcriptome profiles) across a collection of experimental observations and wish to model the relationships between different genes as supported by these experimental data. Given that we wish to permit in our models relationships that are more than pairwise and often non-linear, we need transfer functions that permit these kinds of interactions. The space of continuous multivariate non-linear functions is very large, however. It makes more sense to start by considering the space of discrete functions, which is much smaller (but grows combinatorially with the number of discretization levels). An extreme case of discrete transfer functions would be Boolean logic functions in which variables only take on two possible values but we wish to consider transfer functions that are more flexible than this, not restricting ourselves to logical relationships and not restricting ourselves to only two values.

How do we find the right number of discrete values to use in our modeling and how do we assign continuous observations to these discrete values in the best possible way? These are the questions we address in this chapter. We do not specify how these discrete values will be related to one another in the form of transfer functions — we do this elsewhere in the dissertation. In particular, they could be multi-valued logical, they could be probabilistic and multinomial, or they could be something else altogether. In this chapter, we develop

discretization methods that do not depend on specific assumptions of the transfer function being employed. We choose a discretization based on the information content of the (discretized) observed distributions, rather than any function-specific metric for how effective the discretization is. We simply retain as much information as possible, in a coding-theoretic sense.

In Section 3.1 of this chapter we open by arguing briefly that discretization is a reasonable operation to perform on genomic expression data. We then formalize the discretization problem in Section 3.2 and present some simple discretization methods in Section 3.3. These simple methods, while sometimes useful in their own right, can also be used as input for the discretization level coalescence operators introduced in Section 3.4 that serve as the basis for the information-preserving discretization algorithm we present in Section 3.5. We discuss further issues in the closing section of the chapter.

3.1 Discretization justification

Is it reasonable to consider gene expression as discrete? How can we be certain that we can model continuous values discretely? Inside cells, biochemical reactions are at the lowest level discrete events in which individual molecules and enzymes are brought together for oxidation, reduction, hydrolysis, catalysis, *etc.* Given current measurement technology, however, it is impractical to measure whole-genome expression levels at single-molecule resolution. For this reason, large numbers of cells are pooled together and mRNA removed from the population as a whole. Consequently, the various species of mRNA are typically present in sufficient abundance to be represented as continuous concentration values. Nevertheless, reasoning about continuous concentration values can be problematic given the number of degrees of freedom inherent in arbitrary continuous distributions. Because the amount of data available for reasoning about genetic regulatory networks is comparatively limited, we need to reduce the dimensionality of the modeling.

We can reduce model dimensionality in one of two ways: we can make parametric assumptions about the distributions of continuous variables or we can discretize variables into a small number of levels. There are certainly situations in which one might prefer

one of these methods to another. In our case, however, discretization is preferred for four reasons:

1. We do not have reason to believe that any particular specification of a continuous distribution is especially suited to this problem and perhaps more importantly, it is not clear what kinds of transfer functions should be used to describe the biological relationships between a continuous child variable and multiple impinging continuous parent variables. Discretization offers the benefit, hopefully, of allowing the majority of the qualitative relationships between variables to be modeled (multinomial Bayesian networks allow arbitrary combinatorial transfer functions over discrete variables) while at the same time reducing the dimensionality of the problem.

2. It seems reasonable that for many genes, transcription occurs in one of a small number of states, perhaps low/high, off/low/high, low/medium/high, off/low/medium/high, *etc.*, and that the level of transcription of a gene does not smoothly interpolate between these states but rather that these states approximate transcriptional equilibria maintained by the cell through its genetic regulatory network. We present some evidence later in this chapter that lends credence to this claim.

3. It seems that the mechanisms of cellular regulatory networks we are capable of understanding today can be reasonably approximated by primarily qualitative statements describing the relationships between the states of genes. In this light, it is thus natural to consider statements describing relationships such as "in the presence of galactose, expression of Gal2 is turned on, unless Gal4 is absent". Certainly, it is likely the case that such a statement is an over-simplification and that a non-linear transfer function over continuous expression levels would be able to distinguish finer degrees of interaction between these factors. Nevertheless, a discretized representation is likely sufficient to capture most of the qualitative sense of the relationship between the factors. As discretization can always be extended to represent more levels of expression, it is always possible to later refine our understanding by increasing the number of discretization levels or eventually considering continuous levels of expression. The modeling framework we propose elsewhere in this dissertation is agnostic with respect

to this choice.

4. Discretization, as a general rule, introduces a measure of robustness against error, including error that arises during measurement and normalization of genomic expression data.

Discretized values are an approximation of the reported continuous expression values, which are themselves an approximation of the truly discrete processes taking place within the cells. Each approximation is valid only under certain assumptions so it is important to remember and test these assumptions throughout the process.

3.2 Mathematical formulation of discretization problem

A *discretization* of a real-valued vector $\boldsymbol{x}$ of length N is simply an integer-valued vector $\boldsymbol{d}$ of identical length that satisfies the following properties:

- each element of $\boldsymbol{d}$ is in the set $\{0, \ldots, D-1\}$ for some positive integer D
- for all i, j, we have $d_i \leq d_j$ if and only if $x_i \leq x_j$

We assume henceforth, without loss of generality, that $\boldsymbol{x}$ is sorted. Thus, for all $i < j$, $x_i \leq x_j$. From the definition of discretization, it follows therefore that for all $i < j$, $d_i \leq d_j$.

We can define a *spanning discretization of degree* D to be a discretization that satisfies the additional property that the smallest (first) element of $\boldsymbol{d}$ is equal to 0 and that the largest (last) element of $\boldsymbol{d}$ is equal to $D-1$. Henceforth, we assume that all discretizations are spanning discretizations.

We can also define a *discretization policy of degree* D, which is simply a real-valued vector $\boldsymbol{\Lambda}$ of length $D+1$ that satisfies the following properties:

- for all $i < j$, we have $\boldsymbol{\Lambda}_i < \boldsymbol{\Lambda}_j$
- $\boldsymbol{\Lambda}_0 = -\infty$
- $\boldsymbol{\Lambda}_D = \infty$

The discretization policy vector $\mathbf{\Lambda}$ simply delineates the boundaries of a set of D intervals that define the following mapping from real-valued vectors $\boldsymbol{x}$ to integer-valued vectors $\boldsymbol{d}$:

$$\mathbf{\Lambda}_j < x_i \leq \mathbf{\Lambda}_{j+1} \iff d_i = j \qquad \forall i \in \{0, \ldots, N-1\}, j \in \{0, \ldots, D-1\} \tag{3.1}$$

If we use such a policy to discretize $\boldsymbol{x}$, then we can write the resultant discretization $\boldsymbol{d}$ as being equal to $\boldsymbol{x}^{\mathbf{\Lambda}}$.

3.3 Simple discretization methods

In this section, we present two simple discretization methods. Although each of these methods could be used for discretization in its own right, in this dissertation the methods are used as possible initializations for more complicated discretization methods developed later in this chapter.

3.3.1 Quantile discretization

One simple method for discretization is *quantile discretization*. In this method, the N (sorted) observations of a variable are divided into D discretization levels by placing an equal number of the observations into each of the D discretization levels (within the limits of rounding). Of course, this is only sensible if $D \leq N$. Formally, the observation with index i is discretized as level j if and only if:

$$\left\lfloor \frac{jN}{D} \right\rfloor < i \leq \left\lfloor \frac{(j+1)N}{D} \right\rfloor \tag{3.2}$$

For example, if the discretization is binary ($D = 2$), then the lower half of the (sorted) observations correspond to 0 elements in the discretization vector, and the upper half of the (sorted) observations correspond to 1 elements in the discretization vector. If the discretization is centenary ($D = 100$), then the observations are discretized according to their percentile rank among the set of observations.

Quantile discretization can also be expressed in terms of an equivalent discretization

```
function qpv = quantilePolicyVector(observationVec,numLevels)

  sortedObserv = sort(observationVec);
  qpv = zeros(1,numLevels-1);
  for j = 1:(numLevels-1),                                          5
    bndryIdx = floor(j*numExp/numLevels);
    qpv(j) = (sortedObserv(bndryIdx)+sortedObserv(bndryIdx+1))/2;
  end;
  qpv = [-inf,qpv,inf];
```

Figure 3-1: Code implementing the creation of a quantile discretization policy vector. The code is written in Matlab and takes as input `observationVec`, which corresponds to $\boldsymbol{x}$ in the text, and `numLevels`, which corresponds to D in the text. The quantile policy vector produced as output is denoted `qpv`.

policy vector. One such choice might be:

$$\mathbf{\Lambda} = (-\infty, \frac{x_{\lfloor \frac{N}{D} \rfloor} + x_{\lfloor \frac{N}{D} \rfloor + 1}}{2}, \frac{x_{\lfloor \frac{2N}{D} \rfloor} + x_{\lfloor \frac{2N}{D} \rfloor + 1}}{2}, \ldots, \frac{x_{\lfloor \frac{(D-1)N}{D} \rfloor} + x_{\lfloor \frac{(D-1)N}{D} \rfloor + 1}}{2}, \infty) \quad (3.3)$$

For enhanced clarity, code implementing the creation of a quantile discretization policy vector is provided in Figure 3-1. Here, as elsewhere in the chapter, although the code seems written for human consumption, it can actually be run directly in Matlab.

3.3.2 Interval discretization

Another simple method for discretization is *interval discretization*. In this method, the N (sorted) observations of a variable are divided into D discretization levels based not on their indices but rather based on their bin number in a histogram with D bins, uniformly spaced across the support of the observational distribution. More precisely, we divide the interval $[x_0, x_{N-1}]$ into D equally-sized subintervals and then discretize the observations according to the index of the subinterval to which they belong. Formally, the observation with index i is discretized as level j if and only if:

$$x_0 + \frac{j(x_{N-1} - x_0)}{D} < x_i \leq x_0 + \frac{(j+1)(x_{N-1} - x_0)}{D} \quad (3.4)$$

To handle the boundary case, we assign observation x_0 to be discretized as 0.

```
function ipv = intervalPolicyVector(observationVec,numLevels)

  sortedObserv = sort(observationVec);
  ipv = zeros(1,numLevels-1);
  for j = 1:(numLevels-1),                                          5
    bndrySpacing = (sortedObserv(end)-sortedObserv(1))/numLevels;
    ipv(j) = sortedObserv(1)+(j*bndrySpacing);
  end;
  ipv = [-inf,ipv,inf];
```

Figure 3-2: Code implementing the creation of an interval discretization policy vector. The code is written in Matlab and takes as input `observationVec`, which corresponds to $\boldsymbol{x}$ in the text, and `numLevels`, which corresponds to D in the text. The interval policy vector produced as output is denoted `ipv`.

Interval discretization can also be expressed simply in terms of the equivalent discretization policy vector:

$$\boldsymbol{\Lambda} = (-\infty, x_0 + \frac{(x_{N-1} - x_0)}{D}, x_0 + \frac{2(x_{N-1} - x_0)}{D}, \ldots, x_0 + \frac{(D-1)(x_{N-1} - x_0)}{D}, \infty) \quad (3.5)$$

For enhanced clarity, code implementing the creation of an interval discretization policy vector is provided in Figure 3-2.

3.3.3 Comparing quantile and interval discretization

Under quantile discretization, the number of observations corresponding to each discretization level is guaranteed to be equal, within the limits of rounding. Under interval discretization, the number of observations corresponding to each discretization level is not guaranteed to be equal, even within the limits of rounding. In contrast to quantile discretization, interval discretization can produce a discretization vector where some discretization levels may be represented quite frequently while others may not be represented at all.

Quantile discretization is preserved under monotonic transformation of the observed values $\boldsymbol{x}$ while interval discretization is preserved only under affine transformation of the observed values $\boldsymbol{x}$. As a result, quantile discretization depends only on the ordering of observed values whereas interval discretization also considers the relative spacing of values.

3.3.4 Deterministic and stochastic discretization

Under the assumption of deterministic discretization which we have implicitly considered thus far, each observation is assigned to a single discretization level. However, this assumption can be loosened to consider stochastic or probabilistic discretization in which each observation is distributed with a certain probability over each of the available discretization levels. This adds a smoothing effect to the otherwise sharp distinctions being made in deterministic discretization. By initializing our discretization policy vector suitably, we can achieve either stochastic quantile or stochastic interval discretization, *inter alia.*

To elaborate, rather than considering each observation to be distributed as a delta function about its value, we can instead imagine it to be normally distributed about its value. Then, rather than assigning each observation to a single discretization level according to its value (or equivalently, according to the delta function about its value), we instead spread it over each of the possible discretization levels according to the total area of the normal that is contained within the intervals of the corresponding discretization policy vector.

For enhanced clarity, code implementing the creation of both deterministic and stochastic discretization matrices is presented in Figures 3-3 and 3-4 respectively. In the former, the discretization matrix contains only ones and zeros whereas in the latter, the matrix contains values in the interval [0,1]. Regardless, the sum of elements in each row of the matrix is one. The procedure `stdDev()` in Figure 3-4 simply returns the standard deviation of the normal centered at a given mean; this allows the standard deviation of the normal to vary as a function of its mean if necessary.

3.4 Discretization level coalescence operators

For a number of reasons, it is useful to consider operators over discretization vectors that reduce the number of discretization levels (or degree) of the discretization. When this is done by coalescing some number of neighboring discretization levels together, we can say that such an operator is a *discretization level coalescence* operator.

```
function dd = deterministicDiscretizationMatrix(observationVec,discPolicyVec)

  numObserv = length(observationVec);
  numLevels = length(discPolicyVec)-1;
  dd = zeros(numObserv,numLevels);
  [ignore,binIdx] = histc(observationVec,discPolicyVec);
  for i = 1:numLevels,
    dd(:,i) = (binIdx==i);
  end;
```

Figure 3-3: Code implementing deterministic discretization. The code is written in Matlab and takes as input `observationVec`, which corresponds to $\boldsymbol{x}$ in the text, and `discPolicyVec`, which corresponds to $\boldsymbol{\Lambda}$ in the text. The deterministic discretization matrix produced as output is denoted `dd`.

```
function sd = stochasticDiscretizationMatrix(observationVec,discPolicyVec,stdDev())

  numObserv = length(observationVec);
  numLevels = length(discPolicyVec)-1;
  dd = zeros(numObserv,numLevels);
  for i = 1:numObserv,
    currObserv = observationVec(i);
    normCumDist = normcdf(discPolicyVector,currObserv,stdDev(currObserv));
    dd(i,:) = normCumDist(2:numLevels+1)-normCumDist(1:numLevels);
  end;
```

Figure 3-4: Code implementing stochastic discretization. The code is written in Matlab and takes as input `observationVec`, which corresponds to $\boldsymbol{x}$ in the text, `discPolicyVec`, which corresponds to $\boldsymbol{\Lambda}$ in the text, and `stdDev()`, which returns the standard deviation of the normal centered at a given mean. The stochastic discretization matrix produced as output is denoted `sd`.

Without loss of generality, we can define discretization level coalescence (DLC) operators as reducing the number of discretization levels by coalescing a neighboring pair of levels together into a single discretization level. In this light, we can view the repeated application of DLC operators on an initial discretization as producing a dendogram as shown in Figure 3-5. The initial discretization levels can be viewed as a set of ordered leaves in this dendogram, and each application of a DLC operator merges two nodes together into a single new node one level higher in the dendogram. If the initial discretization is of degree D, then after $D-1$ applications of DLC operators, only a single discretization level remains and the

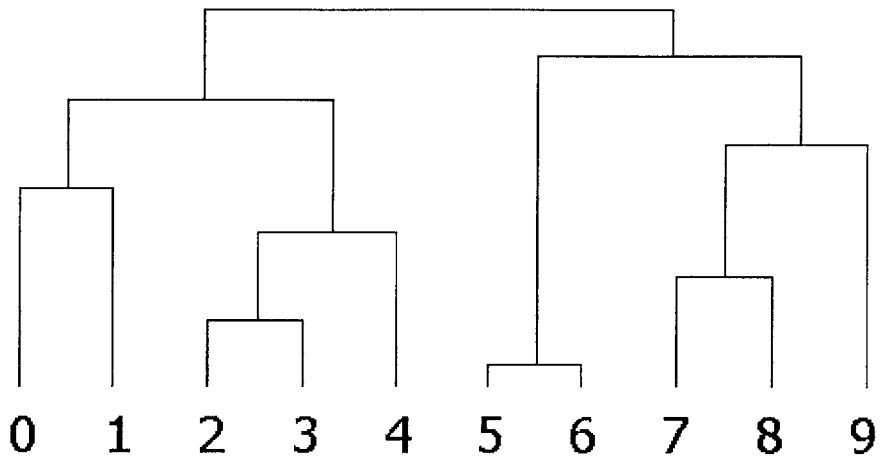

Figure 3-5: *Dendogram showing the sample application of discretization level coalescence operators on an initial discretization of degree ten. The initial discretization levels can be viewed as a set of ordered leaves in the dendogram, and each application of a DLC operator merges two nodes together into a single new node one level higher in the dendogram. By focusing on different depths from the root node of the dendogram, we can extract a coalesced discretization of any degree between one and ten, in this example.*

dendogram reaches a single root node. By focusing on different depths from the root node of the dendogram, we can extract a coalesced discretization of any degree between 1 and D.

It is clear that these DLC operators are lossy in the sense that their application is not invertible (*i.e.*, the underlying mapping is not isomorphic but is instead many-to-one). The method by which a given operator identifies the neighboring pair of levels to coalesce determines the character of the operator as well as the amount of information that is lost upon application of the operator. Because we seek in this dissertation to model statistically informative relationships between sets of variables, we consider DLC operators that seek to minimize the total mutual information that is lost from a set of variables upon operator application.

3.5 Information-preserving discretization

Since DLC operators are lossy, the salient question is: given that we are going to be losing information, which is the most important information to retain and which is the least important? Naturally, the answer to this question depends on the application, but in our context, we seek to model the conditional dependencies and independencies between variables in the domain. So it is important to retain as much information as possible regarding the relationships between variables in the domain. In particular, it is important to discretize variables in such a way that they are not considered in isolation, but rather in relation to one another. For this reason, although it is possible to consider each variable independently during the coalescence process, we choose to coalesce discretization levels for each variable in terms of the mutual information between pairs of variables. More precisely, given a set of variables initially discretized into D discretization levels, how can we apply DLC operators to the *set* of discretized variables to reduce each variable to a smaller number of discretization levels C in such a way as to minimize the total pairwise mutual information lost at each application of the operators?

To amplify this point, consider the scenario depicted in Figure 3-6. Imagine that we have a collection of observations of two hypothetical variables across a number of different experimental conditions and that the observed levels of expression are distributed as shown in Figure 3-6. If we were to discretize the first hypothetical variable in isolation, we might surmise that there are three discretization levels and that the lowest two observations would fall into the low level, the middle three observations would fall into the medium level and the highest observation would fall into the high level, as shown in the top right panel of Figure 3-6. If we were to do the same thing for a second hypothetical variable in isolation, we might perform a similar discretization where each of the three levels has two observations, as shown in the bottom right panel of Figure 3-6. However, when these two variables are considered together as in Figure 3-7, it becomes evident that knowing the level of one of the variables helps to predict the level of the other. When we discretize the two variables in isolation, we do so in such a way that we lose some of the predictive information of one variable with respect to the other in the sense that knowing the discretization level of one

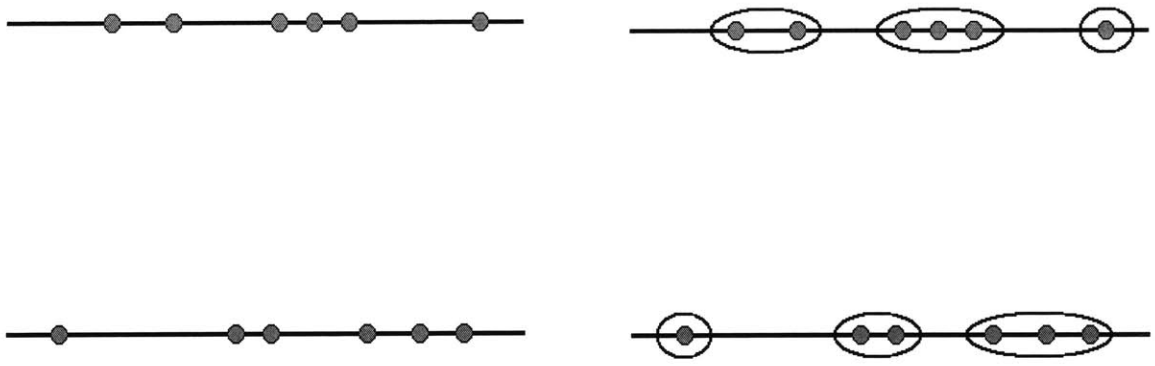

Figure 3-6: *Figure illustrating discretization of variables considered in isolation. The top panels represent the observations of one hypothetical variable and the bottom panels represent the observations of a second hypothetical variable. In the two panels on the left, six observations of the two variables are depicted, scattered along the axis according to their level of expression. In the two panels on the right, these observations have been independently discretized into three discretization levels.*

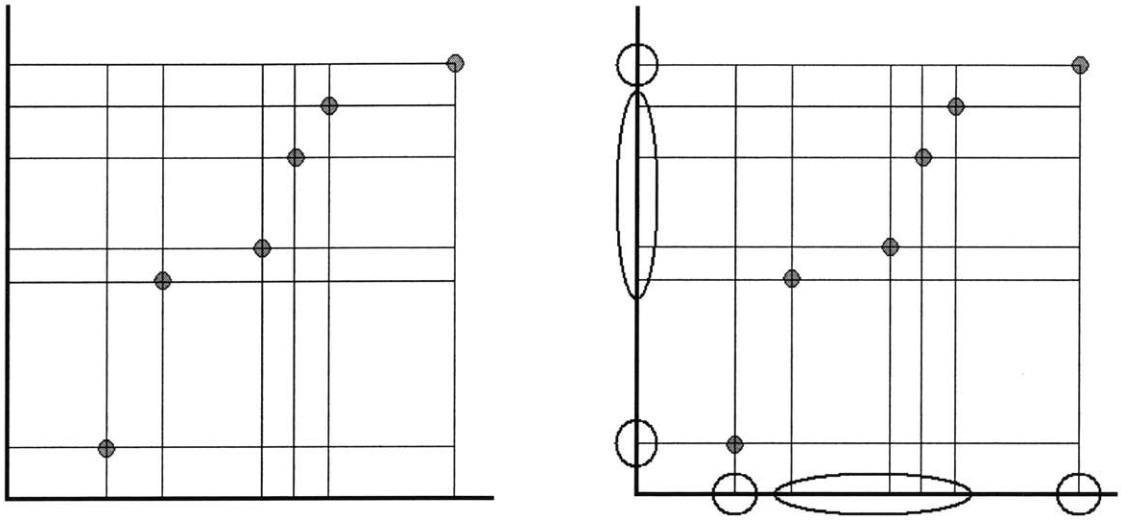

Figure 3-7: *Figure illustrating the principle behind relationship discretization. In the panel on the left, the same two variables from Figure 3-6 are represented, only now their co-expression has been depicted. In the panel on the right, the relationship information between these two variables leads their observations to be discretized into three discretization levels differently than before, preserving more predictive information between one variable's discretization level and the other's.*

variable is no longer a perfect predictor of the discretization level of the other (although they were before discretization). Therefore it is important that we discretize these variables together so as to maximize the amount of mutual information that we retain in the (lossy) discretization process, as shown in the right panel of Figure 3-7. We do so by coalescing discretization levels incrementally to retain as much of the total mutual information between pairs of variables as possible.

3.5.1 Information-preserving discretization algorithm

The input to the discretization coalescence algorithm is an initial discretization. Although any of the simple discretizations discussed above could be used as the initial discretization for application of information-preserving DLC operators, we consider here only initial discretizations using stochastic quantile discretization. Quantile discretization retains more information (in an information theoretic sense) than interval discretization, and stochastic discretization increases this retained information even more through its smooth-

ing effect.

The algorithm itself consists of two loops. The outer loop simply counts from the degree of the initial discretization down to one, with the loop index indicating the number of discretization levels remaining. The inner loop iterates over each of the variables in the set to determine for each variable which single coalescence of neighboring discretization levels reduces the total mutual information score the least. We define the total mutual information score for n discretization levels, $\mathsf{TMI}(n)$, as the sum of the pairwise mutual information between all pairs of variables when each has been discretized into n discretization levels. The algorithm is an extension to pairwise mutual information of an agglomerative algorithm by Slonim and Tishby [108].

For example, if M variables need to be discretized and each has N observations, we can create an initial discretization simply by applying stochastic quantile discretization of degree $D = N$. Other values for D are discussed later, but regardless of what value is chosen, with an initial discretization of degree D for each of the M variables, the total mutual information between variables before any application of DLC operators is simply $\mathsf{TMI}(D)$. At each step of the algorithm, for each variable, the algorithm coalesces some neighboring pair of discretization levels into a single level, reducing the total number of discretization levels by one. Thus, the algorithm runs for $D - 1$ steps — it is finished when it has discretized all observations into a single level, at which time the total mutual information score is zero trivially: $\mathsf{TMI}(1) \equiv 0$. At this point, an entire dendogram similar to the one in Figure 3-5 has been created. If the ultimate number of desired discretization levels C is known in advance, the algorithm can be aborted after computing the necessary number of coalescences rather than proceeding for all $D - 1$ steps of the outer loop. If C is not known in advance, we suggest methods for choosing a suitable value later in this chapter.

As the discretization algorithm is intended to retain as much relationship information between variables as possible, we coalesce a neighboring pair of levels into a single level by choosing the neighboring pair of levels that, when coalesced, loses as little mutual information as possible between that variable and the other variables to be discretized. Before step i of the algorithm, each of the M variables has been discretized into $D - i + 1$ discretization

levels. So there are $D - i$ neighboring pairs of levels to consider coalescing for each of the M variables. For each such neighboring pair, we compute the resultant mutual information that would be lost if the pair were coalesced into a single level and select the pair that has the least such information loss. We identify the M such pairs associated with the M such variables, and once these have been identified, we coalesce them all at once. In other words, the actual coalescing is only implemented at the end of each round (so the order in which the variables are considered during the round is moot).

The time to complete the full discretization process is proportional to:

$$M \sum_{i=1}^{D-1} (D - i) \approx MD^2/2 \tag{3.6}$$

Although this algorithm produces a dendogram like many hierarchical clustering algorithms do, because the variables are ordered and only neighboring levels can be coalesced, we need not consider all pairs of levels at each step of the algorithm but only neighboring ones, so at step i we pay a cost only linear in $D - i$ rather than quadratic in $D - i$. Moreover, once the dendogram is constructed, only a single orientation needs to be considered rather than 2^D such orderings; we can think of the leaves as being anchored in a fixed order before the dendogram is built.

For enhanced clarity, code for performing information-preserving discretization level coalescence based on total mutual information is given in Figure 3-8. The input is an array of discretization matrices, one for each variable in the set. Discretization matrices are produced as output by either the deterministic or stochastic discretization routines presented in Section 3.3.4.

3.5.2 Handling large numbers of observations

If one is dealing with a very large number of observations for each variable (large N), starting with an initial discretization of degree $D = N$ may require an unnecessarily large amount of time to construct the entire dendogram (since the running time is $O(MD^2)$). In such cases, it is possible to reduce the running time of the algorithm by selecting a value for D that is large enough for discretization purposes but much less than N. For example,

```
function coalescenceInfo = tmiDiscretizationLevelCoalescence(arrayOfDiscMatrices)

  numVar = length(arrayOfDiscMatrices);
  [numObserv,numLevels] = size(arrayOfDiscMatrices{1});
  numDLC = numLevels-1;                                                    5
  coalescenceInfo = zeros(numDLC,numVar,2);
  for i = 1:numDLC,
    for j = 1:numVar,
      currNumLevels = numLevels - i;
      sumI = zeros(1,currNumLevels);                                       10
      for k = setdiff(1:numVar,j),
        Px = sum(arrayOfDiscMatrices{j})/numObserv;
        Py = sum(arrayOfDiscMatrices{k})/numObserv;
        Pxy = (arrayOfDiscMatrices{j}'*arrayOfDiscMatrices{k})/numObserv;
        Hx = -Px.*log2(Px);                                                15
        Hy = -Py.*log2(Py);
        Hxy = -Pxy.*log2(Pxy);
        rowIx = Hx-nansum(Hxy');
        I = nansum(rowIx)+nansum(Hy);
        tPx = Px(1:currNumLevels)+Px(2:(currNumLevels+1));                 20
        tPxy = Pxy(1:currNumLevels,:)+Pxy(2:(currNumLevels+1),:);
        tHx = -tPx.*log2(tPx);
        tHxy = -tPxy.*log2(tPxy);
        tI = nansum([repmat(I,[1,currNumLevels]); ...
                     -rowIx(1:currNumLevels); ...                          25
                     -rowIx(2:(currNumLevels+1)); ...
                     (tHx-nansum(tHxy'))]);
        sumI = sumI+tI;
      end;
      [maxMutualInfo,maxIdx] = max(sumI);                                  30
      coalescenceInfo(i,j,:) = [maxIdx,maxMutualInfo];
    end;
    for j = 1:numVar,
      coalesceIdx = coalescenceInfo(i,j,1);
      currDiscMatrix = arrayOfDiscMatrices{j};                             35
      currDiscMatrix(:,coalesceIdx) = currDiscMatrix(:,coalesceIdx)+ ...
                      currDiscMatrix(:,coalesceIdx+1);
      currDiscMatrix(:,coalesceIdx+1) = [];
      arrayOfDiscMatrices{j} = currDiscMatrix;
    end;                                                                   40
  end;
```

Figure 3-8: Code implementing the information-preserving discretization level coalescence algorithm. The code is written in Matlab and takes as input the data structure `arrayOfDiscMatrices`, which is simply an array of discretization matrices produced by either deterministic or stochastic discretization code. The output `coalescenceInfo` records how much mutual information was lost and which levels were merged in each round.

regardless of the number of observations N, it may be possible to initially discretize the observations into percentiles (quantile discretization with 100 discretization levels) and use this as input to the discretization coalescence algorithm. As long as the final number of discretization levels C in this example is significantly less than 100, doing this should have little effect on the end result, except perhaps if, for example, significantly fewer than $\lfloor \frac{N}{100} \rfloor$ observations among the N are associated with a certain critical cellular phenotype. Making this tradeoff between running time and possible information loss can be left to the discretion of the user in the context of the specific application.

3.5.3 Determining the optimal number of discretization levels

How exactly is the final number of discretization levels C determined? One attractive feature of this algorithm is that it measures the amount of information being lost at each step of the algorithm and thus we can determine the optimal number of discretization levels after running the algorithm. For example, Figure 3-9 shows the total mutual information among 36 variables over 320 observations as a function of the number of discretization levels remaining during the discretization level coalescence algorithm. Clearly, after each round of coalescing the total mutual information cannot increase. In other words, for all $i < j$, we have $\mathsf{TMI}(i) \leq \mathsf{TMI}(j)$. Therefore, we know that $\mathsf{TMI}(i)$ is a nondecreasing function of i with $\mathsf{TMI}(1) \equiv 0$. As is immediately evident from the graph, the total mutual information retained at each stage of the discretization level coalescence process is relatively unchanged until a small number of discretization levels are reached. In particular, the amount of mutual information at just five levels of discretization per variable is 99.5% of the total mutual information at the original 100 levels of discretization per variable. An enlargement of the region of the curve between 1 and 10 discretization levels per variable is shown in Figure 3-10 for closer inspection.

The information shown in Figure 3-9 enables us to determine a suitable number of discretization levels for each variable after coalescence is completed, trading off the total amount of retained mutual information with the extra complexity associated with a larger number of discretization levels. What additional complexity is associated with a larger number of discretization levels? Recall that in our modeling framework, the expression

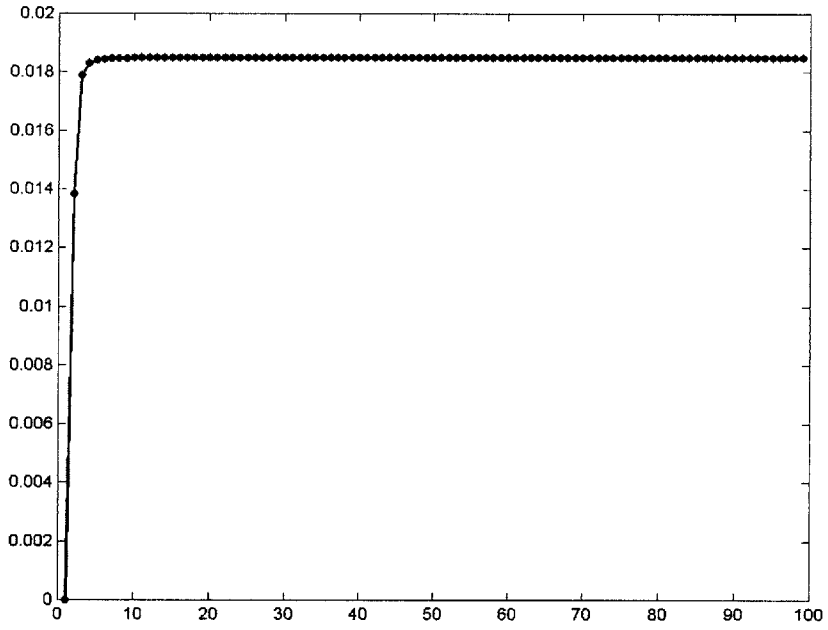

Figure 3-9: *Graph showing the total mutual information preserved during the discretization level coalescence process as a function of the number of discretization levels remaining. When all variables share a single discretization level, the mutual information between them is zero. The long plateau before the total mutual information decreases indicates that even a small number of levels of gene expression may be sufficient to capture statistically predictive relationships between these 36 variables.*

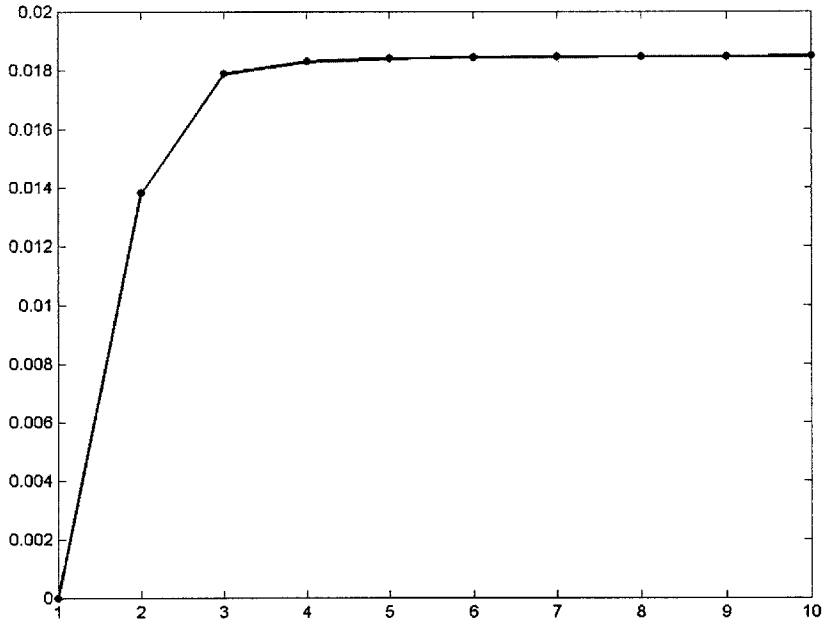

Figure 3-10: *Enlargement of the section of Figure 3-9 corresponding to low numbers of discretization levels remaining. This enlargement is included for clarity.*

level of a given variable can depend combinatorially on the parents of that variable in the graph representing the underlying regulatory network. In particular, for each of the possible discretization levels for a given variable, we must specify the probability of observing the variable at that level of discretization given the (discrete) configuration of its parents. In general, if a variable has n discretization levels and p parents, and the i^{th} parent of the variable has n_i discretization levels where i runs from 1 to p, then the total number of parameters needed to characterize the local conditional probability distribution is:

$$(n-1)\prod_{i=1}^{p} n_i \tag{3.7}$$

If all the variables in the network have the same number of discretization levels, C, then this expression is simply $C^{p+1} - C^p$ which is approximately C^{p+1} for large C. So having a large number of discretization levels requires a large number of parameters to specify the conditional probability distribution for a variable, especially a variable that has a significant number of parents p. If the number of observations of the system is less than the number of parameters necessary to characterize the distribution, then it becomes difficult to determine these parameters with any certainty. For these reasons, it is probably helpful to keep the number of discretization levels at as low a level as is justified by the total mutual information curve produced during the running of the discretization algorithm.

3.6 Discussion

Figures 3-9 and 3-10 lend credence to our earlier claim that genes can likely be modeled with a small number of discretization levels without losing much information in doing so. Moreover, the figures seem to support the claim that genes may operate in only a relatively small number of modes biologically; or if their underlying expression is more continuous than discrete, then the spectrum of continuous expression can be modeled as a relatively small collection of expression regimes for the purposes of modeling the dependence of one gene's expression on another's.

A number of directions exist in which this work can be extended. First, we suggest using as small a number of discretization levels as possible while retaining most of the

mutual information initially present during the discretization process. The total mutual information curve produced during the discretization coalescence algorithm is quite helpful in this regard, but it still leaves the decision to the user. While this flexibility may be a feature in some contexts, it would likely be advantageous for the user to have some guidance as to how to best pick this cutoff. Possible solutions might include theoretical measures based on minimum description length, for example [101, 102].

An additional degree of freedom could also be introduced by allowing each variable to be discretized to a possibly different number of discretization levels. This opens a whole can of worms in our relationship discretization context, but a number of researchers have attempted discretization along these lines, most notably Friedman and Goldszmidt [43] and Monti and Cooper [89]. The work of Monti and Cooper is especially nice as it employs a fully Bayesian method for learning the discretization policy for each variable as it learns the network structure (Friedman and Goldszmidt do something similar but use an MDL approximation), but both methods suffer from computational limitations. An advantage of the methods developed in this chapter, is that discretization needs only be done once. Clearly, there are tradeoffs between the two approaches and context must dictate which is more appropriate.

Finally, although this method allows for stochastic discretization as the output of the discretization coalescence algorithm, we end up discarding this information at the final stage of the process. In other words, although we initialize with stochastic discretization, coalesce with stochastic discretization, compute the residual mutual information at each stage with stochastic discretization, and decide on our final number of discretization levels with stochastic discretization, when we finally apply the corresponding discretization policy learned throughout this process, we apply it as a deterministic discretization to the original data. The reason for this is that the model validation and discovery algorithms further down the analysis pipeline are only implemented to handle deterministically discretized data. A significant extension would be to modify the model validation and discovery algorithms to handle stochastically discretized data and thereby retain this information all the way through the analysis pipeline.

Chapter 4

Bayesian network models

As was discussed in the introductory chapter of this dissertation, with a vast quantity of genomic expression data becoming available, more sophisticated methodologies are needed to advance beyond simple data-driven analysis of this data. In particular, we need methods that are capable of helping us understand the patterns in the data that data-driven analysis reveals but leaves unexplained. To accomplish this, we propose the use of a model-driven analysis paradigm. In addition, model-driven analysis of genomic expression data offers a number of significant benefits including a return to hypothesis-based biological investigation, the ability to capture complex interaction phenomena that are a necessary component of more integrative approaches to scientific inquiry, the ability to refine (or suggest *de novo*) hypotheses consistent with observed data, and the prospect of suggesting experiments that ought to be done to distinguish between possible models. While it therefore seems natural to consider model-driven analysis, there remains a large space of potential modeling frameworks that might be applicable in this particular context.

In this chapter, we first present in Section 4.1 a number of alternatives for modeling genetic regulatory networks so as to explain observed expression data, and eventually settle on Bayesian networks, a class of graphical models that seems a good compromise between various modeling extremes. Section 4.2 provides an introduction to Bayesian networks in terms of the semantics and various modeling characteristics of the networks. In Section 4.3 of this chapter, we discuss the Bayesian nature of these networks, including how models of regulatory networks can be scored rigorously in the presence of noisy expression data using

the Bayesian scoring metric and how these models can incorporate prior information in a principled way. We work through an example in the case of discrete variables with multinomial distributions. In Section 4.4, we present methods for the validation and discovery of Bayesian network models. We close the chapter with a discussion of some subtleties in Section 4.5.

4.1 Consideration of possible modeling frameworks

We can consider a number of possible modeling frameworks for representing the function of genetic regulatory networks. At one end of the spectrum are highly specified models such as those based on differential equations [52, 85, 7] or stochastic Petri nets [90, 54]. These kinds of models seek to explain observed expression levels by capturing the very small-scale dynamics of fundamental interactions taking place within the cell, in some cases simulating not only the temporal evolution of molecular concentrations and reactions, but also the spatial aspect of these phenomena as well. A model at this level of specification would seek to explain the minutiae of how and why various levels of gene expression might be expected to change over time under the influence of various genetic and environmental factors. The difficulty with using such a model for this task is that it is usually so highly specified that it requires not only an exact knowledge of which factors interact with which other ones, which is precisely what we do not know, but also the reaction rates associated with such interactions, in terms of binding affinities, free energies, dissociation rates, equilibrium constants, and the like. While models at this level of specification represent the Holy Grail of our ability to understand what is happening in the cell, with the exception of certain special cases, they are unattainable at this juncture because so little is currently known about which factors in the cell interact with which other factors, let alone the frequencies and rates at which such interactions occur. In contrast, we need models that are more abstracted than these, capable of capturing the kernel phenomena without requiring a burdensome level of specification.

At the other end of the modeling spectrum are highly abstracted models. If the semantics associated with these models are so highly abstracted that they lose the ability to represent

core regulatory phenomena, however, then we can make similarly little progress. One example of such an overly abstracted model might be a Boolean network model [115, 76, 78, 3]. In a Boolean network model, all factors in the genetic regulatory network are represented by Boolean variables, which can only take on two possible values. While in some contexts it may be possible to capture the essential qualities of a particular regulatory network using only two discrete levels of expression for each gene, certainly other contexts arise in which this is not possible and a gene will instead need to be modeled with more than two levels of expression. Moreover, in a Boolean network all relationships between variables are required to be logical, which allows little room for explaining levels of gene expression that have become corrupted by noise during the measurement process or are not the result of clean and logical regulatory processes but rather ones that are inherently stochastic.

Another example of an overly abstracted model might perhaps be a linear regression model of gene expression [36, 118]. While this model avoids some of the limitations of Boolean network models in that gene expression is modeled as being continuous and noise is modeled explicitly, nevertheless it has another limitation of its own in the sense that it can only capture linear relationships between factors in the cell. This can be problematic in a context where, for example, a gene remains unexpressed until the levels of expression of two other genes both exceed some high value. This type of nonlinear `AND` relationship is difficult to explain in the setting of a linear modeling framework.

Between these two extreme ends of the modeling spectrum lie a family of models known as *graphical models*, a family of flexible and interpretable models for compactly representing probabilistic relationships among variables of interest in the form of a graph. While this family of models is fairly large and includes a number of possibly relevant classes of models, we concentrate here on a particular class of models known as *Bayesian networks*. In our modeling framework, Bayesian networks are used to describe relationships between variables in a genetic regulatory network.

Modeling genetic regulatory networks with graphical models like Bayesian networks represents a reasonable compromise between models that are too specified versus models that are too abstracted. Graphical models smoothly interpolate between these two extreme points on the modeling spectrum and thus can be used to operate anywhere along the

spectrum, though at this stage our modeling remains fairly abstracted in the grand scheme of things. For example, in contrast to models employing differential equations to simulate the molecular dynamics of interactions between factors in the cell, determining the precise dynamics of genetic regulation is outside the scope of the Bayesian network techniques we present here. Rather, we seek to develop comprehensive high-level models that are able to suggest which factors in the cell are interacting with which others. Once we have information about which factors interact with which others, this can be used as the basis for constructing more highly specified, low-level models based on differential equations.

Bayesian networks can describe arbitrary combinatorial control of gene expression and thus are not limited to pair-wise or linear interactions between genes. Due to their probabilistic nature, Bayesian networks are robust in the face of both noisy expression data and imperfectly specified hypotheses about the function of genetic regulatory networks. Moreover, Bayesian networks cleanly handle missing data and permit latent variables to represent unobserved factors, and when we extend the semantics of Bayesian networks to allow edge annotations (as described later in Chapter 5), Bayesian networks can specify relationships between variables at increasing levels of refinement. Most importantly, models of genetic regulatory networks that are based on Bayesian networks are biologically interpretable and can be scored rigorously against observed genomic expression data.

We should mention that our work on modeling genetic regulatory networks using Bayesian networks was developed concurrently with but independent of similar work by Murphy and Mian [91] and Friedman, *et al.* [45]. While their research concentrates on different aspects of this domain, all three bodies of work taken together represent a fairly comprehensive treatment of this topic in published literature to date.

4.2 Modeling characteristics of Bayesian network models

Bayesian networks [93, 75, 67] are a member of the family of graphical models, a class of flexible and interpretable models that use graphs for representing probabilistic relationships among variables of interest.

Variables in a Bayesian network can be either discrete or continuous, and can rep-

resent mRNA concentrations, protein concentrations, protein modifications or complexes, metabolites or other small molecules, experimental conditions, genotypic information, or conclusions such as diagnosis or prognosis. A variable that describes an observed value is called an *information variable*, while a variable that describes an unobserved value is called a *latent variable.*

A Bayesian network describes the relationships between variables at both a qualitative and a quantitative level. At a qualitative level, the relationships between variables are simply dependence and conditional independence. These relationships are encoded in the structure of a directed graph, S, to achieve a compact and interpretable representation. Vertices of the graph correspond to variables, and directed edges between vertices represent dependencies between variables. The fewer edges a model has, the more constrained is the model since it makes more independence assertions. In practice, we seek sparse models because they are able to explain away certain "indirect" dependencies through more "direct" dependencies mediated by other variables. Formally, if vertices X and Y are d-separated[1] by a set of vertices $\boldsymbol{Z}$, then X and Y are conditionally independent given $\boldsymbol{Z}$. In particular, a directed edge exists from X to Y, then Y is dependent on X. Since Y can have multiple incoming directed edges, it can depend combinatorially on multiple variables. We call variables that have a directed edge to Y the *parents* of Y, denoted $\mathrm{Pa}(Y)$.

At a quantitative level, relationships between variables are described by a family of joint probability distributions that are consistent with the independence assertions embedded in the graph. Each member of this family is described by the vector, $\boldsymbol{\theta}$, of parameters that characterize it. As this method is Bayesian in nature, we do not consider only a single value for $\boldsymbol{\theta}$, but rather a distribution over all possible values of $\boldsymbol{\theta}$ that are consistent with the structure of the graph, S. This distribution over distributions enables these models to avoid over-fitting, a common problem when parameters are restricted to a single value in the context of small quantities of data.

In a Bayesian network, each joint probability distribution over the space of variables

[1]The notion of d-separation is an extension of the notion of separation to directed graphs. We say that X and Y are *d-connected* if there exists an (undirected) path between X and Y that only visits some combination of (i) vertices that are non-colliders and also non-members of $\boldsymbol{Z}$, or (ii) vertices that are colliders and also either members of $\boldsymbol{Z}$ or have descendants that are members of $\boldsymbol{Z}$. We say that X and Y are *d-separated* if they are not d-connected. For more information on this topic see [93, 120].

can be factored into a product over the variables, where each term is simply the probability distribution for that variable conditioned on the set of parent variables:

$$\mathrm{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathrm{P}(X_i \mid \mathbf{Pa}(X_i)) \tag{4.1}$$

This follows from the conditional Markov assumption which states that each variable is independent of its non-descendants when conditioned on its parents. The parameters that characterize the conditional probability distributions on the right hand side of Equation 4.1 therefore comprise the parameter vector, $\boldsymbol{\theta}$.

While continuous variables are permitted in Bayesian networks, for the remainder of this dissertation we consider only discrete variables to simplify the exposition. Each variable is thus in one of a set of states, and the number of states used to model a variable represents a tradeoff between precision, the ability to intuit what the state of the variable means, and the computational complexity of evaluating a model with a given number of states (for a more detailed discussion of this point in the context of discretizing continuous levels of reported gene expression, see Chapter 3). In the case of multinomial distributions over discretized variables, Bayesian networks do not concern themselves with the relative order of the discretized levels. Consequently, we can no longer leverage information about the fact that discretization level 0 is less than level 1 or that level 1 is between levels 0 and 2, *e.g.* This problem can be addressed by using edge annotations to represent monotonicity constraints, an approach we present in the next chapter.

4.3 Bayesian characteristics of Bayesian network models

While we have addressed the basic modeling aspects of Bayesian networks, we still need to consider the Bayesian aspects of these networks. There has been a newfound appreciation for Bayesian methods in statistical investigation and learning [84] primarily because of the ability of Bayesian methods to naturally incorporate prior information with observed data, to cleanly handle missing data or unobserved variables, to prevent overfitting in the context of low data, and to avoid the semblance of ad-hockery that sometimes accompanies frequentist statistical methods. These characteristics are all relevant in our

modeling context. We now discuss how a Bayesian network can be scored in a principled manner in the presence of observed data regarding the states of its variables, and also consider how this model scoring methodology is able to cope with prior information, missing data, and the risk of over-fitting.

4.3.1 Bayesian scoring metric

When scoring Bayesian networks against observational data, we employ the Bayesian scoring metric, a principled statistical scoring metric that allows us to directly compare the merits of alternative models of genetic regulatory networks.[2] The model scores produced by the Bayesian scoring metric permit us to rank alternative models based on their ability to explain observed data economically. Moreover, the difference between the scores for any two models leads to a direct significance measure for determining how strongly one should be preferred over the other.

According to the Bayesian scoring metric, the score of a model is defined as the logarithm of the probability of the model given the observed data. Formally,

$$\mathsf{BayesianScore}(S) = \log \mathrm{p}(S \mid D) \tag{4.2}$$

$$= \log \mathrm{p}(S) + \log \mathrm{p}(D \mid S) + c \tag{4.3}$$

where the first term on the right hand side of Equation 4.3 is the log *prior* distribution of S, the second term is the log *likelihood* of the observed data D given S, and c is a constant that does not depend on S.[3] The likelihood term can be expanded as follows:

$$\mathrm{p}(D \mid S) = \int_{\boldsymbol{\theta}} \cdots \int \rho(D, \boldsymbol{\theta} \mid S)\, \mathrm{d}\boldsymbol{\theta} \tag{4.4}$$

$$= \int_{\boldsymbol{\theta}} \cdots \int \mathrm{p}(D \mid \boldsymbol{\theta}, S)\rho(\boldsymbol{\theta} \mid S)\, \mathrm{d}\boldsymbol{\theta} \tag{4.5}$$

[2]Due to space limitations, we present here only the basic intuition behind the Bayesian scoring metric; more detailed quantitative treatments are available elsewhere [28, 59]. We note that the entire discussion is equally valid in the case of dynamic Bayesian networks.

[3]We assume henceforth that $c = 0$ to simplify computation. Note that this is the same as defining the Bayesian scoring metric as the logarithm of the joint probability of the model and the data, which is equivalent for the purposes of comparing models in terms of their ability to explain a single set of data D.

From this last expression, we see that the likelihood component of a model's score can be viewed as the average probability of generating the observed data over all possible values of the parameter vector, $\boldsymbol{\theta}$.

Because the Bayesian scoring metric includes an average over a family of probability distributions, it is well suited to our context for a number of reasons. First, it includes an inherent penalty for model complexity, thereby balancing a model's ability to explain observed data with its ability to do so economically. Consequently, it guards against overfitting models to data. Second, regulatory network models are permitted to be incomplete. An incomplete model contains additional degrees of freedom pertaining to the possible ways of completing the model, and is thus penalized by the scoring metric for these additional degrees of freedom. Scores improve as a model converges to properly depict underlying regulatory mechanisms without extraneous degrees of freedom, thereby allowing network elucidation to proceed incrementally. Third, it allows us to represent uncertainty about the precise dependencies between variables since we need not select a single value for $\boldsymbol{\theta}$, but rather can permit all feasible values to exist in the distribution over $\boldsymbol{\theta}$.

4.3.2 Prior specification and incorporation

In a Bayesian setting, we need to establish prior distributions both over the set of parameter vectors, $\boldsymbol{\theta}$, that describe the joint probability distribution, and over the set of model structures, S. In a discrete Bayesian network satisfying the reasonable assumptions of parameter modularity, parameter independence, and likelihood equivalence, Heckerman, *et al.* [59] have shown that the parameters characterizing the local conditional probability distributions in a discrete multinomial Bayesian network are necessarily Dirichlet distributed. If prior information about parameters is available, this information can be captured in the form of an equivalent *prior network* with Dirichlet distributed parameters [59]. However, if no prior information about parameters is available, an uninformative prior is frequently employed. In either case, an *equivalent sample size* needs to be specified. This value is a measure of how confident we are in the prior over model parameters relative to the quantity of data.

With respect to any prior information we might have regarding model structures, this

information can again be naturally incorporated, as was the case with prior information about the model parameters, by using an informative prior distribution. In this case, the prior distribution over structures $\mathrm{p}(S)$ is used to capture this prior information. Although no theoretical requirements govern how this should best be done, various proposals have been offered. For example, we might consider simple nonuniform priors over structures based on either the number of edges present or their degree of divergence from some pre-specified prior structure. If no prior information is available, then the prior over structures is usually assumed to be distributed uniformly over structures for computational convenience ($\forall S, S'$, we have $\mathrm{p}(S) = \mathrm{p}(S')$). However, it should be noted that computational convenience is a relative notion. For example, nonuniform priors over model structures can arise from choices based on computational convenience if we are examining the space of model equivalence classes (PDAGs) or node orderings rather than the space of model structures.

4.3.3 Bayesian scoring metric example

In the case of a discrete Bayesian network with multinomial local conditional probability distributions,[4] the integral shown in Equation 4.5 can be represented exactly in closed form. Borrowing notation from Heckerman [57], we index the n variables in the Bayesian network using the variable i, we index the q_i parent configurations of variable i using the variable j, and we index the r_i states of variable i using the variable k. We designate as θ_{ijk} the probability of observing variable i in state k given parent configuration j. As previously discussed, the prior distribution over these parameters under reasonable assumptions is necessarily Dirichlet:

$$(\theta_{ij1}, \ldots, \theta_{ijr_i}) \sim \mathsf{Dirichlet}(\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \qquad \forall i, j \tag{4.6}$$

If we let N_{ijk} be the number of occurrences in the data set D of variable i in state k given parent configuration j, and then define $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, it can be

[4] In the case of multinomial distributions, the distributions can be represented simply as conditional probability tables.

shown that the Bayesian scoring metric is expressible in closed form as:

$$\text{BayesianScore}(S) = \log \text{p}(S) + \log \left\{ \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right) \right\} \quad (4.7)$$

$$= \log \text{p}(S) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left\{ \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\} \quad (4.8)$$

where $\Gamma(\cdot)$ is the gamma function. A more detailed exposition, as well as a derivation of this closed form expression, can be found in Cooper and Herskovits [28] and Heckerman, *et al.* [59].

4.4 Applying the Bayesian scoring metric

In this dissertation, we apply the Bayesian scoring metric in two different contexts: validation of genetic regulatory network models and discovery of genetic regulatory network models. We discuss these two modeling scenarios in turn.

4.4.1 Genetic regulatory network validation

When a collection of models exists for explaining existing data, we can validate these models against the data by measuring the extent to which each model is able to explain the data relative to the others, as shown in Figure 1-1 of Chapter 1. With the Bayesian scoring metric already in place, this process becomes relatively straightforward. We simply need to compute the score associated with each model and then give preference to the model which scores the highest. The relative likelihood of one model over another is simply the exponential of the difference between the two models. In this sense, the scores computed by the Bayesian scoring metric give us a direct measure of the relative likelihood of one model explaining the data in comparison with another. The inherent penalty against over-complexity and the ability of the Bayesian scoring metric to guard against over-fitting permit this score difference to be interpreted naturally.

The validation context is useful in enabling the kinds of hypothesis-driven investigation of data that is the basis of scientific inquiry as discussed in the introductory chapter of

this dissertation. It forces scientists to go through the exercise of formulating models that represent hypotheses about the function of genetic regulatory networks. However, one limitation of this validation approach is that hypotheses must be postulated in advance before they can be compared against one another. As Figure 1-2 of Chapter 1 indicates, we would like to also consider the possibility of discovering new models either *de novo* or to extend an existing collection of models.

4.4.2 Genetic regulatory network discovery

Methods for the induction (discovery) of Bayesian network models from observational data generally fall into two classes. The first class consists of constructive methods based on the examination of various constraints that must hold over the conditional dependences and independences computable from the empirical probability distributions on the variables represented in the data. Examples include the PC and FCI algorithms developed by Spirtes and Glymour [110, 111] and the IC algorithms developed by Pearl [93]. The second class, which we consider exclusively here, consists of search methods that seek to maximize some scoring function that describes the ability of the network to explain the observed data. The Bayesian scoring metric introduced earlier in this chapter is an especially common choice for the scoring function, although other choices can be made if the Bayesian scoring metric is difficult to compute exactly. Since the discovery of the highest-scoring model under the Bayesian scoring metric for a given set of data is known to be NP-hard [25, 24], we consider heuristic rather than exhaustive search strategies such as best-first, beam search, or A*.

Among heuristic search strategies, we again have a number of possible choices to consider. The first choice we need to make is with regards to the search space we use. Most algorithms search in the space of Bayesian networks (DAGs), but it is also possible to search in the space of node orderings [44], in the space of model equivalence classes (PDAGs) [22], or alternating between the space of Bayesian networks (DAGs) and graph skeletons [112]. Henceforth, we consider only search methods over the space of Bayesian networks as it is especially easy to define local search operators in this space.

Most Bayesian network search strategies are local because the evaluation metric (the Bayesian scoring metric in this case) is typically decomposable. This allows for a significant

savings in the computation of scores for candidates in the search process — a "local" change to the model's structure requires only a "local" change to the model's score. Local search operators in the DAG search space usually traverse the space by employing only edge additions, edge deletions, and edge reversals, but these local operations can be pasted together in numerous ways to assemble a heuristic search algorithm.

One alternative is known as greedy hill-climbing. At any stage in this algorithm, the single local operation which maximally increases the overall score is effected and the process then repeats itself. While $O(n^2)$ such local operations must be considered in the first round, since the evaluation metric is decomposable and the search operators are local, only $O(n)$ scores need to be recomputed in each round after the first.

Another alternative is to select a local operation at random (either an edge addition, edge deletion, or edge reversal) and then select a valid target edge at random (for addition, deletion, or reversal respectively) and implement the operation if and only if the operation increases the score. A more general version of this greedy random algorithm is that of Metropolis [88] wherein the random local operation is implemented if it increases the score, as before, but is also implemented with a certain probability p if it does not (setting $p = 0$ yields the previous greedy random algorithm). The Metropolis search strategy forms the basis of a more complicated search strategy known as simulated annealing, so named because it operates in a manner analogous to the physical process of annealing. During the search process, the Metropolis algorithm is run as a subroutine at various "temperatures" T. The prevailing temperature and the score difference between graphs determine the transition probability p within Metropolis, with higher temperatures indicating more permissive transitions. Initially, the temperature is set very high (allowing almost all changes to be made), but is gradually reduced according to some schedule until it reaches zero, when p is also zero, at which point the Metropolis subroutine is equivalent to the greedy random algorithm. The schedule that the temperature is constrained to follow can be varied to produce different kinds of search algorithms including ones that allow for "reannealing" after the temperature becomes sufficiently low.

The greedy algorithms (hill-climbing and random) settle into local maxima directly since they are guaranteed to increase the scores of the graphs they consider in each round. As

the topography of the space of graphs can be quite irregular, it is usually wise to augment these algorithms with random restarts to consider other regions of the space in order to find better maxima, especially in the case of the deterministic hill-climbing algorithm.

Genetic regulatory network discovery implementations

In this dissertation, we implemented two basic search algorithms that served as the basis for various model discovery strategy extensions. The first is a greedy random search algorithm with random restarts. The second is a simulated annealing search algorithm with a temperature schedule that allows for "reannealing" after the temperature becomes sufficiently low. In each case, we use as our prior distribution over parameters the uninformative Dirichlet prior of Buntine [16] (which Heckerman, *et al.* [59] call BDeu).

Both of these implementations were extended to be able to search for models with constraints specifying which edges are required to be present and which are required to be absent. This allows for the incorporation of prior information about edges in the graph since these kinds of constrained search algorithms are equivalent to search algorithms with a nonuniform prior over structures that gives zero weight to models that either include edges required to be absent or do not include edges required to be present. In this way, data from other sources like location analysis can be easily incorporated by modifying the prior over structures. We use a uniform prior over structures when we perform uninformed search, but employ these nonuniform priors over structures when we wish to incorporate prior information from other sources.

Regardless of one's choice of heuristic search algorithm, it is necessary to specify in advance the initial graph to be submitted to the search algorithm. One can begin with a complete graph, an empty graph, a pre-specified graph (including the prior network), or a random graph. In the absence of a strong preference for one of these alternatives, another choice can be made. Although the construction of an optimal arbitrary DAG is known to be NP-hard, the construction of an optimal tree or forest can be done in polynomial time [27]. It is relatively easy to compute an optimal forest from the observed data and then use this as the initial graph for the heuristic search algorithm. As any of the nodes in the trees that comprise the forest can be chosen as root, a collection of equally good graphs can be

generated for use in initializing the search algorithm or its restarts.

In the case of the greedy random search algorithm, the initial graphs submitted to the search algorithm were the set of orderings of the optimal forest consistent with the observed data. After each of these forest orderings were considered, random restarts were initialized by taking a previously high-scoring graph at random and then randomly removing a number of edges. In the simulated annealing context, we simply use the empty graph as an initial graph — the random edge additions, deletions, and reversals of the Metropolis algorithm at high temperatures quickly randomize the graph structure so the initial choice is irrelevant in this setting.

Our search algorithm implementations are written in C and are capable of searching about 200,000-250,000 (not necessarily unique) graphs per minute on a 400MHz Pentium II Linux workstation. The code is not especially optimized but does keep a small hash table of the scores of recently visited graphs in case those model structures are visited again. Anecdotally, in our experience the greedy random search algorithm frequently identifies its highest scoring graph within about five minutes, even if it is allowed to run up to two hours. In contrast, the simulated annealing algorithm seems to benefit from being run for about half an hour before it visits a region of particularly high scoring graphs and running the algorithm for many hours sometimes produces new regions of even higher scoring graphs. In practice, we observed that the simulated annealing algorithm is able to find higher scoring graphs than the greedy random search algorithm, provided each is allowed to run for a reasonably long period of time (long enough to allow the simulated annealing algorithm to reach sufficiently low temperatures in its temperature schedule).

As these algorithms retain a table of the highest scoring graphs visited thus far, they are anytime algorithms — they can be stopped at any time and report the best graph visited thus far. Allowing the algorithms to run longer can only improve the result.

Genetic regulatory network discovery application contexts

We apply these two search algorithms, and their extensions to edge-constrained search, in two contexts. The first is known as *model selection* and the second as *model averaging.* In the model selection context, the goal is to select a single model that best explains the

observed data. In such a setting, the search algorithms are run for a period of time and then report the single highest-scoring model visited during that time. This single model can then be used to represent the underlying structure of the system generating the data and can be used for making predictions about the function of the system.

The problem with the model selection context is that it tends to over-fit the data by selecting a single model and ignoring completely other models that score nearly as well. Although the Bayesian scoring metric includes an inherent penalty for unnecessary parameter complexity within a given model, it cannot guard against the possibility that when millions of different models are examined during heuristic search, that any one of them may be over-fitting the observed data. Just as the Bayesian scoring metric eliminated the risk of *parameter* over-fitting by averaging over all parameter settings rather than depend on a single maximum a posteriori parameter setting, the Bayesian approach for eliminating the risk of *model structure* over-fitting is to compute probabilities of features of interest as an average over the posterior model distribution rather than depend on a single maximum a posteriori model structure, as occurs in model selection. For example, if we are interested in determining whether the data supports the inclusion of an edge between two variables X and Y, a more principled approach than simply examining the maximum a posteriori model produced by model selection is to compute:

$$\mathrm{p}(E_{XY} \mid D) = \sum_S \mathrm{p}(E_{XY}, S \mid D) \tag{4.9}$$

$$= \sum_S \mathrm{p}(E_{XY} \mid D, S) \cdot \mathrm{p}(S \mid D) \tag{4.10}$$

$$= \sum_S 1_{XY}(S) \cdot \mathrm{e}^{\mathsf{BayesianScore}(S)} \tag{4.11}$$

where E_{XY} represents an edge from variable X to variable Y and $1_{XY}(S)$ is an indicator function that is one if and only if graph S includes E_{XY} as an edge. However, this sum is difficult to compute because the space of graphs S is enormous. Fortunately, it is possible to approximate this sum since the vast bulk of its mass lies among the highest scoring models; the exponential factor in the sum has the effect of drowning out all but the highest scoring models, even though they are relatively infrequent. For example, if we restrict our attention

to the N highest scoring models, and index these by the variable i, then we have:

$$\mathrm{p}(E_{XY} \mid D) \approx \frac{\sum_{i=1}^{N} 1_{XY}(S_i) \cdot \mathrm{e}^{\mathsf{BayesianScore}(S_i)}}{\sum_{i=1}^{N} \mathrm{e}^{\mathsf{BayesianScore}(S_i)}} \tag{4.12}$$

Using model averaging in this way reduces the risk of over-fitting the data by considering a multitude of models when computing the probabilities of features of interest.

4.5 Discussion

We use this section to discuss a number of subtleties regarding the use of Bayesian networks in the context of modeling genetic regulatory networks. The first is the notion of statistical significance. The Bayesian scoring metric is a principled metric for comparing alternative models relative to one another because of its ability to guard against over-fitting, its ability to incorporate prior information where available, and its ability to handle noisy and missing data. Moreover, as we have indicated, the difference between the scores for any two models leads to a direct significance measure for determining how strongly one should be preferred over the other: since the scores are log probabilities of model structures conditioned on the observed data, the exponential of the score difference between two model structures indicates the relative likelihood of one model with respect to the other, conditioned on the observed data. No concept of p-value exists in a Bayesian context. One could easily argue, as others have, that relative likelihood is a more natural measure of significance than p-value but this is not the place for such an argument.

We have mentioned the ability to represent as Bayesian networks models that contain variables that are unobserved or for whom data is occasionally missing. The difficulty with these latent variable models is that the integrals computed as part of the Bayesian scoring metric can no longer be solved exactly once we are faced with incomplete data. One way to score models with latent variables is to instantiate the latent variables by sampling from the distribution of possible values for each such variable (*e.g.*, MCMC methods). Though this is feasible for small networks, it becomes computationally prohibitive as networks become

very large. In such settings, variational approximation methods [66, 9] can be used, either on their own or in conjunction with sampling. In addition, variational methods can also yield upper and lower bounds on the score, often enabling the highest scoring graph to be identified without resorting to sampling. For reasons of computational simplicity, we consider in this dissertation only models with variables for which we have complete data. The extension to the context of incomplete data, while computationally burdensome, is fairly straightforward.

As previously discussed, model scores depend on the available data, which has two implications. First, while Bayesian networks are well-suited to dealing robustly with noisy data, as noise increases, the score difference between correct and incorrect models (and thus the significance) goes down. In the limit of uninformative data, correct models score as poorly as incorrect ones, which is to be expected. Second, the ability of particular data to enhance score difference between models suggests the possibility of performing experimental suggestion in the future. In such a context, existing models and data could be used to generate suggestions for new experiments, yielding data that would optimally elucidate a given regulatory network.

Although we only discuss static models of regulatory networks in this work, Bayesian networks can also be used to model dynamic processes such as feedback [32, 71, 14, 13]. This is accomplished by "unrolling" a static model, creating a series of connected models that contain dependencies spanning across time steps. In a modeling context, dynamic Bayesian networks smoothly interpolate between static graphical models and differential equation models.

We should also point out that care needs to be taken in the context of interventional data (as distinguished from observational data). Variables which have been exogenously fixed by the researcher in the course of an experiment cannot be reasoned about in terms of their causes. Under these circumstances, the edges impinging upon exogenously fixed variables are typically excluded from the model structure when evidence is presented. However, interventional data can help to resolve causal ambiguities of the type that arise from statistical analysis of purely observational data. For this reason, interventional data play an important role in the elucidation of genetic regulatory networks. For a further discussion

of the relationships between causality and intervention, see Pearl [94].

The culture of biological investigation entails making predictions from data and then testing. It should be emphasized that while we are primarily occupied here with formulating models from data, once we have models in hand it is straightforward to make predictions from them. To the extent that the models are correct, they can be used to predict the impact of various types of interventions in the system, as well as generate simulated data that can be compared with observed data. In fact, at some level the Bayesian scoring metric for a particular model can be seen as a quantitative measurement of the proximity of observed data to the simulated data that the model is able to generate [58]. Certainly the practice of testing models by using them to make predictions is appropriate. Moreover, it fits quite naturally into the vision of incremental or interactive elucidation of regulatory networks presented in the concluding chapter of this dissertation. In this sense, the models act as a guide for the planning of future experiments and for our understanding of biological systems rather than established fact. Models should always be thought of as hypotheses rather than absolute truth.

Certain limitations exist when using Bayesian networks for modeling genetic regulatory networks. The most important of these is the caution with which models must be interpreted. While graphs are highly interpretable structures for representing statistical dependencies, they have the potential to be misleading if interpreted incorrectly. It is important to distinguish between statistical interaction and (physical) biological interaction.

In general, multiple biological mechanisms may map to the same set of statistical dependencies and thus be hard to distinguish on the basis of statistical tests alone. Furthermore, if sufficient data does not exist to observe a system in a number of different configurations, we may not be able to uncover certain dependencies. These two limitations are not specific to this methodology, however, but rather are true for statistical methods in general. We defer a more comprehensive discussion of this topic until Section 8.1.1 of the concluding chapter. We note that although multiple biological mechanisms may map to the same set of dependence statements, that frequently the opposite is true so the method is typically likely to be useful. Moreover, as mentioned above, interventional data can help resolve certain ambiguities that can arise in the context of purely observational data.

Another limitation is that Bayesian network models are required to be acyclic, which seems a serious problem in light of the fact that biological systems are known to involve significant amounts of feedback and auto-regulation, *e.g.* This limitation can be mitigated by considering dynamic Bayesian networks, as indicated above. In dynamic Bayesian networks, the acyclicity constraint does not preclude feedback but only precludes a variable at one instant in time from affecting a variable at a previous instant in time, which is a perfectly sensible constraint from the point of view of our temporal understanding of cause. Nevertheless, and especially in the context of standard (static) Bayesian networks, this limitation reinforces the notion that caution must be used in interpreting Bayesian network structures.

As for the cost associated with scoring large models, it should be noted that this cost is to a large extent based on the in-degree (number of parents) of the variables in the models. As we scale up to larger models, the in-degree is likely to remain fairly small whereas the out-degree might be very large, which is fine for our Bayesian network approach.

Chapter 5

Representing and scoring annotated network models

Biologists use many specialized terms to describe the actions and interactions of factors within the cell. A fairly comprehensive listing is presented in Table 5.1. Each of these terms implies a specific kind of relationship between the factors involved in the interaction, a relationship that is specified at a finer degree of granularity than the generic statement about conditional dependence that is implied by the existence of an edge connecting the corresponding vertices in a Bayesian network. We would like to represent in our networks this kind of refined knowledge about the form of the relationship between factors in the cell and in addition, be able to score in a principled way models with these refined relationships present.

In Sections 5.1 and 5.2 of this Chapter, we discuss how relationships of increased refinement between variables in a Bayesian network can be represented and scored through the introduction of a constraint framework. We argue the theoretical merits of such a framework in Section 5.3. In Section 5.4, we present the precise semantics of the monotonicity edge annotations developed in this chapter. Section 5.5 indicates how we modify the Bayesian scoring metric to score these annotated network models in a manner that preserves its inherent penalty for complexity. We close with a discussion in Section 5.6.

Table 5.1: *Terms used to describe various relationships between biological molecules or complexes within the cell.*

turns on	turns off	raises	lowers
activates	deactivates	represses	derepresses
inhibits	deinhibits	expresses	suppresses
methylates	demethylates	phosphorylates	dephosphorylates
protects	deprotects	reduces	oxidizes
transcribes	translates	regulates	controls
catalyzes	metabolizes	ligates	binds
initiates	enhances	silences	stimulates
induces	promotes	requires	elevates
is necessary for	is a component of	is a substitute for	is a factor in

5.1 Modeling increased knowledge refinement

Bayesian network semantics regarding graph structures specify that directed edges either exist in the graph or do not exist. If a directed edge exists between two vertices, then the child depends conditionally on the parent, and if a directed edge does not exist between two vertices, then the two variables are conditionally independent of one another. Bayesian network semantics do not permit, however, the representation of more refined qualitative information about the relationship between variables in the structure of the graph. Nevertheless, it is often useful to consider refinements in the degree of our understanding of the dependence relationships between variables in a joint probability space. A simple example of such a refinement is the case of conditional independence itself. Conditional independence is a refinement of conditional dependence in that conditional dependence includes conditional independence as a special case. Because this is the only form of refinement present within the existing framework of Bayesian network semantics, it is helpful to extend the framework to consider more refinements of the dependencies between variables.

5.1.1 Monotonicity refinement

One example of such a refinement might be *monotonicity refinement.* We may seek to represent a monotonic relationship between two variables in a network. Under such an assumption, the child variable is increasingly likely to be in a higher state as the parent

variable monotonically changes its state. Not only are standard Bayesian networks unable to constrain their consideration of the relationship between two variables to be monotonic, in the case of discrete Bayesian networks with multinomial local conditional probability distributions, no relevance is attached to the particular labels associated with a discrete variable. Recall that the labels themselves can be permuted or interchanged and the score associated with any structure representing the joint probability space remains unchanged.

However, the labels associated with discrete variables may contain important structure. In particular, the labels may have an implicit complete ordering associated with them, as they do in our context. For example, the label 1 comes between the labels 0 and 2, which itself comes between the labels 1 and 3, and so forth. Standard multinomial Bayesian network models have little interest in this information but the modeler likely does, especially in the context of monotonic relationships between variables. For instance, as a particular variable changes its state from one discretization level to another in an increasing fashion, a conditionally dependent variable may also be changing its state in an increasing fashion. This type of statement cannot be made if the labels do not have some inherent ordering that permits us to understand what it means for a variable to be changing state "in an increasing fashion". Moreover, this type of statement is impossible to represent within the framework of standard Bayesian network semantics. This type of statement represents a refinement of generic conditional dependence in the sense that it represents the subset of generic conditional dependence that corresponds to a monotonic form of conditional dependence.

5.1.2 Knowledge refinement tree

Figure 5-1 depicts a tree structure in which the nodes of the tree describe the degree of knowledge refinement regarding the relationship between two variables of interest, here labeled X and Y. Child nodes in the tree are knowledge refinements of their parent nodes, and parent nodes, conversely, represent knowledge coarsenings of their child nodes. Consequently, the root of the tree contains the description of the relationship between X and Y that is the least refined relationship possible: that X and Y are conditionally dependent. One possible refinement of this relationship, as mentioned above, is that X and Y are con-

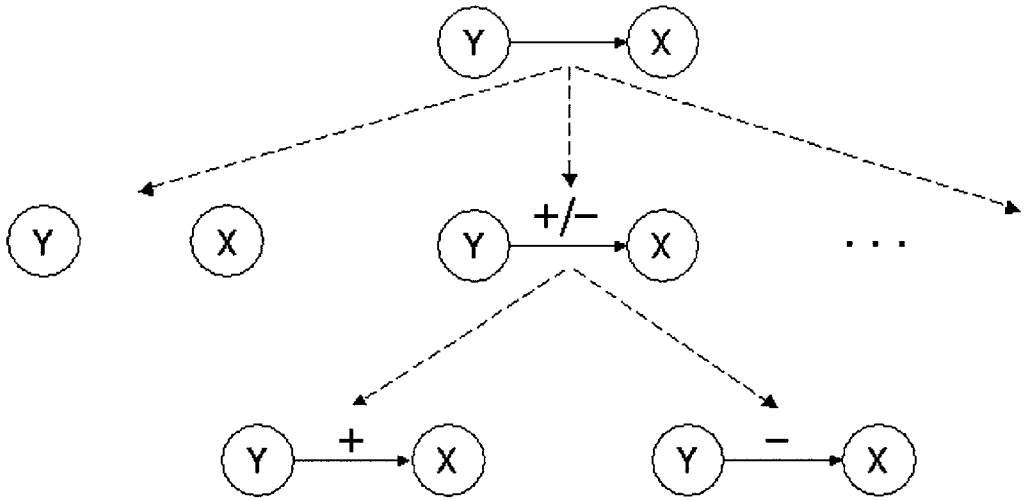

Figure 5-1: *Tree depicting possible refinements of the relationship between variables in an annotated Bayesian network. The root of the tree contains a single unannotated edge between two variables X and Y, indicating conditional dependence between the two but not restricting the kind of dependence. A refinement of this relationship is conditional independence, shown as the leftmost child of the root. Similarly, monotonic dependence is a refinement of generic conditional dependence, and positive and negative monotonic dependence are further refinements in turn. Additional levels and methods of refinement are possible, as suggested by the ellipsis, but are not considered here.*

ditionally independent. By specifying the conditional independence of X and Y, we make a testable claim about the joint probability space described by the two variables. In particular, we narrow the space of parameters that are necessary for characterizing the relationship between X and Y. If X and Y are conditionally dependent as shown at the root of the tree and if X has r_X states and Y has r_Y states, then we require $r_X r_Y - 1$ parameters to characterize the joint probability space. In scoring such a model, we need to integrate over this $(r_X r_Y - 1)$-dimensional simplex characterizing the joint probability space. But if X and Y are conditionally independent, then we need only specify $(r_X - 1) + (r_Y - 1)$ parameters to characterize the joint probability space. In scoring such a model, we only need to integrate over this lower-dimensional subspace within the larger $(r_X r_Y - 1)$-dimensional simplex.

In addition to the refinement relationship that exists between conditional dependence and conditional independence in Figure 5-1, the tree also shows some additional refinement relationships we can consider. Conditional dependence can be refined as monotonic conditional dependence (depicted by use of the $+/-$ annotation on the edge between X and Y); and monotonic conditional dependence can be further refined as either positive monotonic conditional dependence (depicted by use of the $+$ annotation on the edge between X and Y) or negative monotonic conditional dependence (depicted by use of the $-$ annotation on the edge between X and Y). The ellipsis in the tree is included to suggest that other possible refinements of the conditional dependence relationship are certainly possible. As monotonic relationships are the most useful in our setting and the most straightforward to characterize semantically, we consider only these types of annotations henceforth. One possible extension of this work would be to consider and develop the semantics of additional refinements that are perhaps less intuitive but might be useful in different modeling contexts.[1]

Although child nodes in the tree represent refinements of parent nodes, and thus parent nodes can explain any interaction that their children can explain, we would like to be able to score models with annotated edges in such a manner that scores actually improve as refinements are made, provided that those refinements are consistent with the underlying process generating the observed data. This can happen if the scoring metric includes a

[1] For example, alternative refinements of conditional dependence might include concave or convex annotations; further refinements of monotonic relationships might include linear, sigmoid, saturating, or logarithmic annotations.

penalty for complexity.

5.2 Scoring increased knowledge refinement

Fortunately, the Bayesian scoring metric already includes an inherent penalty for complexity. In particular, although data generated from a network with two conditionally independent nodes can be represented by a network with two conditionally dependent nodes, the score of the more general model is penalized for its complexity when scored against independently generated data since the score is an average over a larger space of possible parameters settings and that larger space of parameter settings does not contribute to the model's ability to explain the generated data on average. In the same way, we would like to be able to score annotated models so that when data is generated from a network with an edge relationship of a refined nature, an annotated network with that edge refinement scores better than any corresponding generalized model with a coarser relationship annotation. Consequently, it should be possible for scores to improve even as model relationships are refined.

Of course, just as is true of the standard Bayesian scoring metric, there may be times when the more general model scores better than the more refined model. In such cases, presumably the extra complexity is needed to suitably describe the observed data. In particular, with reference to Figure 5-1, if the network at the root of the tree scored better than any of the other four networks in the figure, then presumably the variables X and Y share a relationship that is non-independent (*i.e.*, dependent) and non-monotonic, for example.

It should be mentioned that the idea of representing monotonic refinements of edge relationships in a Bayesian network was developed independently and previously by Wellman, *et al.* [121, 80, 79] although their work deals only with inference under qualitative annotations; they do not seek to quantify the impact of these annotations on the score of the corresponding network. Also, more recent independent work by Wittig and Jameson [123] presents a quantitative approach to scoring models with monotonic annotations but does so in a non-Bayesian manner. The approach we take in this chapter is to lever-

age the intuition behind the inherent complexity penalty of the Bayesian scoring metric to develop a fully Bayesian solution to the problem of scoring annotated Bayesian networks.

We propose a general method for providing and scoring increased refinement of knowledge in graphical models by introducing a *constraint framework*. In this framework, refined knowledge about the form of a relationship between two variables is represented as a constraint that the relationship must satisfy. The data is not forced to obey these constraints (after all, the data is noisy) but the parameters that characterize the distributions used to model the data are forced to obey these constraints. In this framework, we can leverage the intuition behind the standard Bayesian scoring metric to produce a new scoring metric for models with annotated edges that preserves the inherent complexity penalty of the original metric. Before describing the exact method for exploiting such constraints, we first discuss how the presence of constraints can be beneficial in scoring graphical models in terms of their ability to explain observed data.

5.3 Theoretical motivation for constraint framework

Recall from the previous chapter that in comparing the relative validity of models, we score the models using the Bayesian scoring metric. The Bayesian scoring metric for a particular model is defined as the logarithm of the probability of the model given the observed data. Formally,

$$\mathsf{BayesianScore}(S) = \log \mathrm{p}(S \mid D) \tag{5.1}$$

$$= \log \mathrm{p}(S) + \log \mathrm{p}(D \mid S) + c \tag{5.2}$$

where the first term on the right hand side of Equation 5.2 is the log *prior* distribution of S, the second term is the log *likelihood* of S given the observed data D, and c is a constant

that does not depend on S. The likelihood term can be expanded as follows:

$$\mathrm{p}(D \mid S) = \int_{\boldsymbol{\theta}} \cdots \int \rho(D, \boldsymbol{\theta} \mid S)\, \mathrm{d}\boldsymbol{\theta} \tag{5.3}$$

$$= \int_{\boldsymbol{\theta}} \cdots \int \mathrm{p}(D \mid \boldsymbol{\theta}, S)\rho(\boldsymbol{\theta} \mid S)\, \mathrm{d}\boldsymbol{\theta} \tag{5.4}$$

From this last expression, we see that the likelihood component of a model's score can be viewed as the average probability of generating the observed data over all possible values of the parameter vector, $\boldsymbol{\theta}$.

Equation 5.4 suggests that from a sampling perspective, the contribution of the likelihood term to the score can be viewed as a two-level data generation process whereby a realization of the parameter vector, $\boldsymbol{\theta}$, is selected at random from its prior distribution, and then the probability of generating the observed data is calculated using this realization of $\boldsymbol{\theta}$. The probability of generating the data is then averaged over repeated samplings. This interpretation reveals that a model scores poorly if there does not exist a sufficiently large mass of realizations in the complete distribution over $\boldsymbol{\theta}$ that are capable of generating the data with sufficiently high probability.

On the other hand, if the model is constrained in the sense that its constrained distribution over $\boldsymbol{\theta}$ has a great deal of its mass concentrated on realizations that are capable of generating the data with sufficiently high probability, then the constrained model scores better under the Bayesian scoring metric. For example, one such constraint on the type of relationship between variables that we have already encountered is conditional independence (edge absence), which is merely a special instance of refinement in this constraint framework. Whether the relationship between two variables is constrained as independence or in some other fashion, if the constraint permits the model to avoid unneeded complexity by restricting itself to realizations of $\boldsymbol{\theta}$ that are capable of generating the data with sufficiently high probability, then the model's score increases under the Bayesian scoring metric. This is the basis for the complexity penalty inherent in the Bayesian scoring metric and serves as the basis for the new metric we develop for scoring network models with annotated edges where the annotations imply constraints on the form of the relationship between the

variables and thus on the parameters that describe this relationship.

5.4 Semantics of annotated network models

We now extend Bayesian network model semantics by adding the ability to annotate edges, permitting us to represent additional information about the type of dependence relationship between variables. Although many such annotations are possible, because monotonic relationships are especially useful in a biological setting and the most straightforward to characterize semantically, we consider here only the following four types of edges:

- An *unannotated* edge from X to Y represents a dependence that is unconstrained (the default case). In the presence of unannotated edges from all parents of Y, we can represent arbitrary combinatorial control of Y.

- A *positive (+)* edge from X to Y indicates that higher values of X are constrained to bias the distribution of Y higher. This monotonic influence of X on Y holds for all possible values of the other parents of Y, though the strength of the influence can vary with the setting of the other parents. Formally, for all values y of Y, for all values $x_i < x_j$ of X, and for all instantiations $\mathcal{I}$ of the variables in $\mathrm{Pa}(Y)/X$, we require $\mathrm{P}(Y > y \mid X = x_i, \mathcal{I}) \leq \mathrm{P}(Y > y \mid X = x_j, \mathcal{I})$.

- A *negative (−)* edge from X to Y indicates that higher values of X are constrained to bias the distribution of Y lower. This monotonic influence of X on Y holds for all possible values of the other parents of Y, again with possibly varying strength. Formally, for all values y of Y, for all values $x_i < x_j$ of X, and for all instantiations $\mathcal{I}$ of the variables in $\mathrm{Pa}(Y)/X$, we require $\mathrm{P}(Y > y \mid X = x_i, \mathcal{I}) \geq \mathrm{P}(Y > y \mid X = x_j, \mathcal{I})$.

- A *positive/negative (+/−)* edge from X to Y indicates that Y's dependence on X is either positive or negative but the true relationship is not known. This monotonic influence of X on Y holds for all possible values of the other parents of Y, again with possibly varying strength.

As discussed earlier, other types of relationships may occur in a given network like concavity or, in the case of multiple parents, a non-linear relationship like `XOR`. The annotations

considered above cannot be used to model these kinds of relationships but this poses little concern for two reasons. First, suitable annotations could be easily added to the framework if they are found to be necessary for modeling biological relationships. Second, the fact that the current set of annotations is not capable of describing these relationships means that if such a relationship existed in the cell, the unannotated edge would out-score the other possible refinements in the presence of sufficient data. In this manner, non-monotonic dependence may be identified by finding edges where a lack of annotation scores better than either the presence of any monotonicity annotation or the lack of the edge altogether. For these two reasons, the framework can be said to be both extensible and sensible.

Because edge annotations describe the relationship between a variable and a single parent while Bayesian networks describe the relationship between a variable and all its parents, we have chosen to specify the semantics of annotations by requiring that the implied constraints hold for all possible values of the other parents.

A given Bayesian network can have any combination of edge annotations. This enables us to represent finer degrees of refinement regarding the types of relationships between variables when we desire, but does not force us to do so since unannotated edges are always permitted. It also allows a model to evolve as more knowledge is gained about the types of influences that are present in the biological system under study. For example, all edges can be initially unannotated, with $+/-$ and then $+$ and $-$ annotations being added incrementally as activators and repressors are later identified.

5.5 Scoring annotated network models

The implied constraints on the form of the dependence between variables permit us to score annotated models much as we score unannotated models. We simply modify the scoring metric so that the likelihood term is now the average probability of generating the observed data over all possible values of the parameter vector $\boldsymbol{\theta}$ that satisfy the constraints implied by the annotations.

For example, consider a simple network with two variables and one edge annotated with $+$ as shown at the bottom left of Figure 5-1. In such a case, the parameter vector $\boldsymbol{\theta}$ must

satisfy the following set of constraints:

$$\mathrm{P}(Y > y \mid X = x_i) \leq \mathrm{P}(Y > y \mid X = x_j) \qquad \forall y \text{ and } x_i < x_j \tag{5.5}$$

This set of constraints on the parameters reduces the size of the parameter space and the integral in the likelihood term of the score is modified accordingly:

$$\mathrm{p}(D \mid S) = \int_{\boldsymbol{\theta}_{\text{valid}}} \cdots \int \mathrm{p}(D \mid \boldsymbol{\theta}, S) \rho(\boldsymbol{\theta} \mid S) \, \mathrm{d}\boldsymbol{\theta} \tag{5.6}$$

Alternatively, the integral over a subspace of the full parameter space is easily seen to be equivalent to an integral over the full parameter space but with a modified prior over parameters:

$$\mathrm{p}(D \mid S) = \int_{\boldsymbol{\theta}} \cdots \int \mathrm{p}(D \mid \boldsymbol{\theta}, S) \rho_{\text{constr}}(\boldsymbol{\theta} \mid S) \, \mathrm{d}\boldsymbol{\theta} \tag{5.7}$$

This modified prior, $\rho_{\text{constr}}(\boldsymbol{\theta} \mid S)$, is simply the previous (Dirichlet) prior, $\rho(\boldsymbol{\theta} \mid S)$, but with no support for parameter settings that violate any of the constraints implied by the edge annotations. The modified prior must of course be renormalized:

$$\rho_{\text{constr}}(\boldsymbol{\theta} \mid S) = \frac{\rho(\boldsymbol{\theta} \mid S) \cdot 1_{\text{constr}}(\boldsymbol{\theta})}{\int \cdots \int_{\boldsymbol{\theta}} 1_{\text{constr}}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}} \tag{5.8}$$

where $1_{\text{constr}}(\boldsymbol{\theta})$ is an indicator function that is 1 if and only if $\boldsymbol{\theta}$ satisfies all the constraints implied by the edge annotations.

5.6 Discussion

The scoring method for annotated network models developed here is applied in the following chapter in the context of validation of models of the galactose regulatory network. That the inherent penalty for complexity is preserved is readily apparent from the results presented in Section 6.3.2.

The current implementation of the scoring method relies on straightforward rejection

sampling as suggested by Equations 5.7 and 5.8. This works fine for models with only a limited number of annotations but eventually, more efficient methods would be desirable. Since each of the annotation constraints is simply a plane in parameter space, one possible direction for future work is to consider more sophisticated methods for sampling concave functions over polytopes as suggested by Equation 5.6. The work of Applegate and Kannan [6] offers a possible starting point.

Perhaps the single most important direction for future work is a general theory of how to automatically discover optimal annotations for a given model structure and a given set of experimental data. Even beyond this, the ability to discover annotated structures from a given set of data is of significant interest.

Finally, as alluded to earlier in the chapter, alternative annotations and their corresponding semantics could be considered and eventually integrated into the constraint framework developed here for representing and scoring annotated network models.

Chapter 6

Modeling the yeast galactose regulatory network

As a demonstration of the utility of Bayesian networks for modeling genetic regulatory networks, we analyze and score models of the regulatory network responsible for the control of genes necessary for galactose metabolism in *S. cerevisiae*. Because this is a fairly well-understood model system in yeast, it affords us the opportunity to evaluate our methodology in a setting where we can rely on accepted fact. We are also utilizing our Bayesian network methodology to explore other systems that are less well-understood like pheromone response in yeast and present those results in the next chapter.

Section 6.1 opens the chapter with an examination of the data we use in the elucidation of the galactose regulatory network. In Section 6.2, we formulate the Bayesian network models to be validated in comparison with one another, each representing a hypothesis about the relationship between Gal80, Gal4, and Gal2. We present model validation results, both with standard and annotated Bayesian network models, in Section 6.3 and then close in Section 6.4 with a discussion of these results and the care with which they should be interpreted.

6.1 Data preparation

A set of 52 samples of unsynchronized *Saccharomyces cerevisiae* populations of varying genotype were observed under a diversity of experimental conditions. The set of samples ranges widely but consists primarily of observations of various wild-type and mutant *S. cerevisiae* strains made under a variety of environmental conditions including exposure to different nutritive media as well as exposure to stresses like oxidative species, excessive acidity, and excessive alkalinity.

Whole-genome expression data for each of these 52 observations was collected using Affymetrix Ye6100 GeneChips. These GeneChips are 50-micron Affymetrix chips and consequently, four chips are required to measure the expression of all 6135 genes in the *S. cerevisiae* genome. Thus, a total of 208 GeneChips were used in collecting this data.

6.1.1 Data normalization

The reported *average difference* values from these 208 Affymetrix GeneChips were normalized using the MAP spiked control normalization methods described in Chapter 2 of this dissertation and in Hartemink, *et al.* [55]. The output of this process was a 6135×52 matrix of normalized log expression values, one row for each gene in the yeast genome and one column for each experimental observation.

6.1.2 Data selection

Much is known about the actions of the various genes and proteins involved in the galactose metabolism pathways [68, 82, 125, 98, 100, 97, 60, 31]. Galactose is transported into the cell through the cell membrane by Gal2, a galactose and glucose permease. The sugar is metabolized by the cell through the actions of a small number of proteins including Gal1, a galactokinase that catalyzes the first step in galactose metabolism, Gal7, Gal10, and Pgm2 (also known as Gal5).

The expression of these various genes is regulated by means of an interaction between two regulatory proteins, Gal4 and Gal80. Gal4 is expressed constitutively at very low levels and has a DNA binding domain as well as a transcriptional activation domain, enabling it

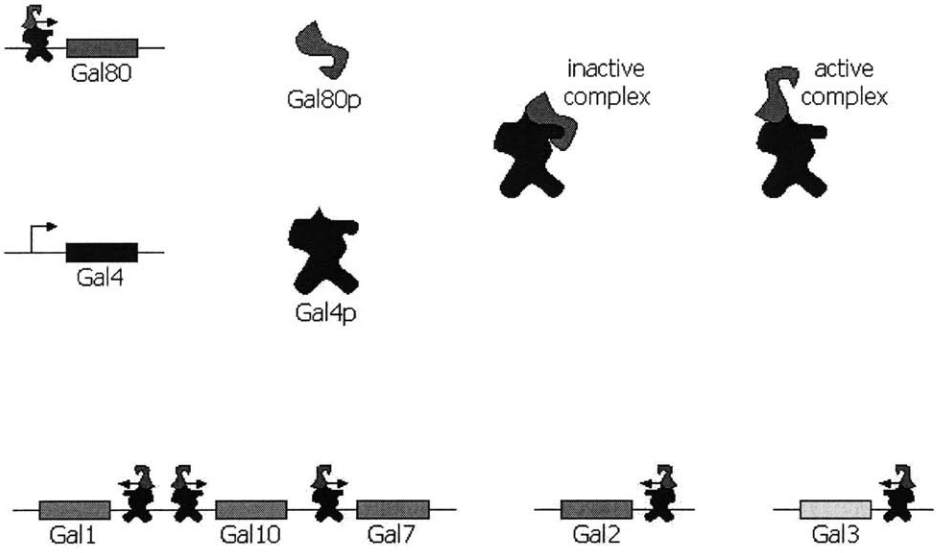

Figure 6-1: *Schematic representation of the factors involved in regulating the levels of expression of the various genes necessary for galactose metabolism in* S. cerevisiae. *Gal4 is a transcriptional activator with a common promoter sequence upstream of most other genes in the network. Gal4 activity can be repressed by Gal80, but in the presence of galactose and absence of glucose, the complex becomes active and the genes are quickly up-regulated.*

to recognize and bind to a specific consensus sequence upstream of its target genes and then recruit or otherwise activate RNA polymerase to initiate transcription of those genes. Gal80 binds directly to Gal4 and inhibits its ability to activate its target genes in the absence of galactose or presence of glucose. However, when the cell is grown in a galactose medium, the repressive role of Gal80 is impeded and the various galactose genes transcriptionally activated by Gal4 are quickly up-regulated. It has been known for some time that Gal3 can bind directly to Gal80, thereby possibly implicating Gal3 in the process of switching on galactose gene expression. However, very recent evidence suggests that Gal3 is localized to the cytoplasm [97] and the exact mechanism by which Gal3 may be modifying the repressive role of Gal80 is still not known. The relationships among these various factors are shown in Figure 6-1.

6.1.3 Data discretization

Bayesian networks are capable of modeling continuous variables using parametric or semi-parametric density estimation, but discretization is more robust in a setting such as this one where only a small number of observations is available. The normalized expression values associated with genes involved in the galactose regulatory network were therefore extracted from the normalization output matrix and binary discretization was performed independently for each gene using a maximum-likelihood separation technique. Other sensible discretization methods could also have been employed; for the particular data set and models in this example, results do not depend on the discretization method and are robust among various different sensible methods. In general, however, the discretization method employed will affect reported scores, and we continue to develop discretization methods that are well suited for genomic expression data, such as those presented in Chapter 3 (which are used in the following chapter when we discretize a larger number of genes).

6.2 Model validation candidates

Examples of genetic regulatory networks represented as Bayesian networks are shown in Figure 6-2. Boxed variables suffixed with "m" describe mRNA levels that can be deter-

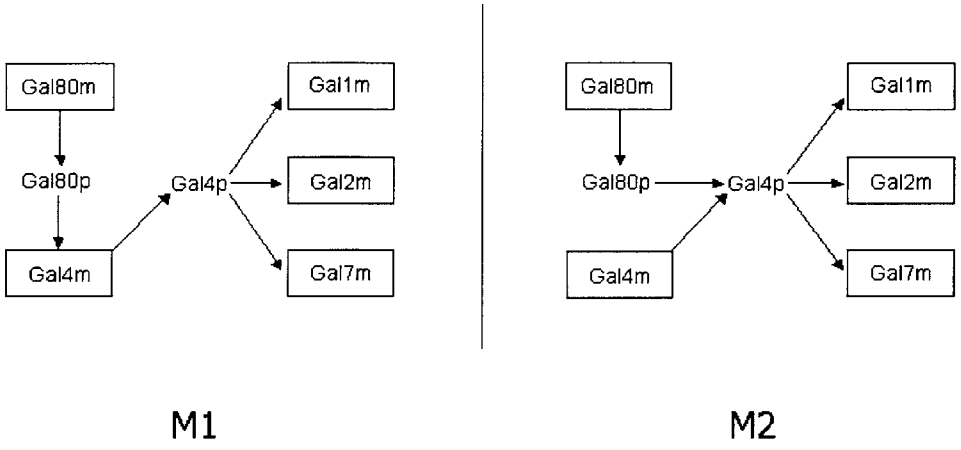

Figure 6-2: *Representative Bayesian networks for describing a portion of the galactose system in yeast. The model M1 on the left represents the claim that Gal80p represses the transcription of Gal4m, while the model M2 on the right represents the claim that Gal80p inhibits Gal4p activity posttranslationally.*

mined from expression array data. Unboxed variables suffixed with "p" describe protein levels; in this model they would be latent variables whose values cannot be measured directly. The two networks in the figure represent two competing models of a portion of the galactose system in yeast, and differ in terms of the dependence relationships they assert hold between the variables Gal80p, Gal4m, and Gal4p. To quote from Johnston, "it was originally proposed that Gal80 protein is a repressor of GAL4 transcription. It is now clear that GAL4 is expressed constitutively and that its activity is inhibited by Gal80 protein posttranslationally" [68]. The network on the left (M1) represents the original proposition, while the network on the right (M2) represents the new assertion.

The models in Figure 6-3 represent the same conditional independence assertions of the models in Figure 6-2, but are simplified to reveal the kernel of the distinction between the two hypotheses in terms of the effects on the observed transcript levels, namely that in M1, Gal2m is independent of Gal80m when conditioned on Gal4m, while in M2, Gal4m is marginally independent of Gal80m.

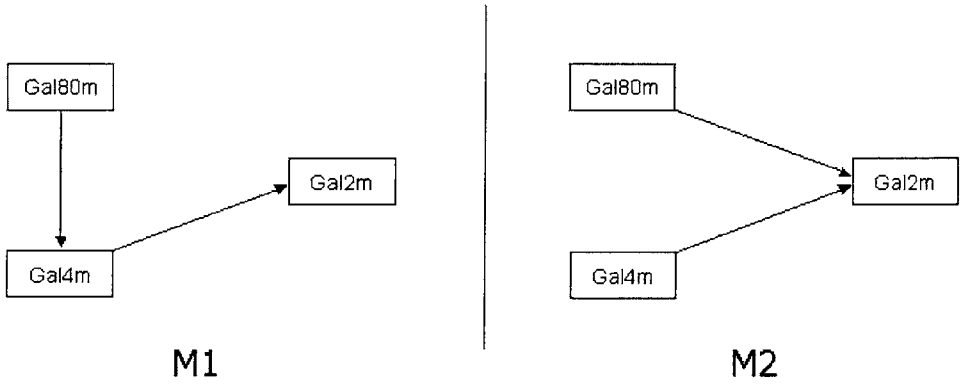

Figure 6-3: *Simplified Bayesian networks for describing a portion of the galactose system in yeast. These simplified versions of M1 and M2 capture the kernel of the conditional independence assertions of the more complex models of Figure 6-2. As above, in M1, Gal2m is independent of Gal80m when conditioned on Gal4m, and in M2, Gal4m is marginally independent of Gal80m.*

6.3 Model validation results

6.3.1 Scoring galactose network models

Using the Bayesian scoring metric, we are able to compare the two models shown in Figure 6-3 in terms of their relative likelihood of explaining the observed (now discretized) data. The model M1 received a score of -44.0, while the model M2 received a score of -34.5. This score difference translates to the data being over 13,000 times more likely to be observed under M2, the currently accepted model. For extra measure, we also scored a more complex model (M1 or M2) that would admit either of the two models as special cases. The data do not persuade us to accept such a model since the score (-35.4) is lower than that of the currently accepted model.

We then broadened our scope to consider not only these three models, but all possible models among these three variables.[1] Results of this analysis are shown in Figure 6-4. As is evident from the figure, the models fall into two primary groupings based on their score: those scoring between -34.1 and -35.4 (unshaded) which all include an edge between Gal80m

[1]Note that some model structure possibilities are equivalent to others in that they describe the same set of conditional independencies; more accurately then, we consider all possible model equivalence classes [23].

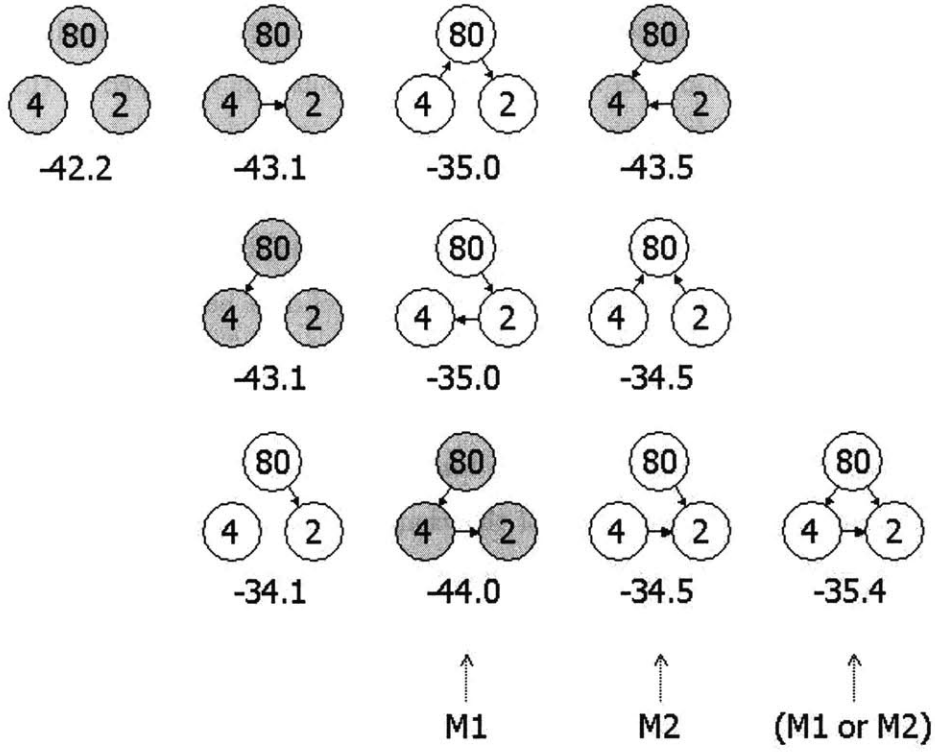

Figure 6-4: *Scores for all model equivalence classes of the three variable galactose system. The classes of models that score poorly are shown shaded while the classes of models that score well are shown unshaded. The feature that perfectly characterizes classes of models scoring well is the presence of an edge between Gal80m and Gal2m, lending support to the claim that the two variables are not conditionally independent. The previously considered models M1, M2, and (M1 or M2) are indicated.*

and Gal2m, and those scoring between -42.2 and -44.0 (shaded) which all do not include an edge between Gal80m and Gal2m. This lends support to the claim that Gal80m and Gal2m are very unlikely to be conditionally independent given Gal4m, again consistent with the currently accepted hypothesis.

It is interesting to note that the best scoring model in Figure 6-4 actually has no edge from Gal4m to Gal2m, indicating little evidence in this particular data set for requiring the edge to be present. This is consistent with the fact that under normal conditions, Gal4m is constitutively expressed and its influence on Gal2m is usually regulated by the action of Gal80 protein, as hypothesis M2 indicates. If the data set instead contained experiments with GAL4 deletion mutants in which the absence of Gal4m resulted in a loss of Gal2m expression, there would be strong support for the inclusion of this edge. We could also learn more about the presence of this edge from other kinds of data besides genomic expression data. For example, data from transcription factor binding location analysis reveals that Gal4 protein indeed binds upstream of the GAL2 gene, indicating that an edge between Gal4m and Gal2m might likely exist when data from other sources is taken into account. We discuss these subtleties at the end of this chapter and demonstrate in Chapter 7 how this kind of binding location data can be integrated into our framework during the model discovery process.

6.3.2 Scoring annotated galactose network models

When we expand the semantics of Bayesian networks to include annotated edges, we are able to score models that describe more refined relationships between variables. For example, when we consider again the two models M1 and M2, and allow the edges in each model to take on all possible combinations of annotations ($-$, $+/-$, or $+$), we are able to score the models as shown in Table 6.1. In model M1, adding different kinds of annotations fails to change the score significantly, as the structure of the graph is fundamentally limited in explaining the observed expression data. The same effect is observed when the edge between Gal4m and Gal2m is considered in model M2, which is consistent with the results of Figure 6-4 indicating little evidence in this particular data set for including the edge between Gal4m and Gal2m at all (as might be expected from the constitutive expression

Table 6.1: *Scores for models M1 and M2 under all possible configurations of annotated edges.*

		Gal4m → Gal2m		
		−	+/−	+
Gal80m	−	-45.3	-44.6	-44.2
↓	+/−	-44.6	-43.8	-43.4
Gal4m	+	-44.2	-43.4	-43.0

M1

		Gal4m → Gal2m		
		−	+/−	+
Gal80m	−	-48.9	-47.3	-46.7
↓	+/−	-35.5	-35.4	-35.4
Gal2m	+	-34.8	-34.8	-34.7

M2

of Gal4m and the lack of GAL4 deletion mutant experiments in the data set). In contrast, adding a + annotation to the edge between Gal80m and Gal2m results in a score comparable with previously achieved scores, but adding a − annotation to the same edge worsens the score dramatically. Such an asymmetric response is to be expected as failure to explain the observed data is more revealing than success. This example illustrates that when the constraints implied by edge annotations cannot be satisfied by the data, scores result that are as poor as when the underlying structure is incorrect. For this reason, annotations serve as a useful discriminator of the kinds of relationships present in the data.

The lowest score (-33.6) is achieved by model M2 when the edge from Gal4m to Gal2m is unannotated and the edge from Gal80m to Gal2m is labeled +. Although Gal80 is known to act in a repressive role in the cell, its level increases as galactose becomes available for metabolism. This increase, however, is more than offset by a rise in the level of a factor that counteracts the effect of Gal80. The identity of this factor is currently unknown, but it is believed to require a byproduct of the metabolism of galactose [68]. Experimental evidence suggests that Gal3 protein plays a role through its ability to bind Gal80, but Gal1 has a similar but diminished ability and galactose seems to be required to be present [60, 31, 125, 98, 100]. A complete model would include the effect of these latent (unmeasured) variables, and in such a model, it would be expected that with sufficient data, the edge between Gal80 and Gal2 would be labeled −, corresponding to the direct repressive role of Gal80. Nevertheless, in the limited model considered here, a + annotation for the edge is indeed correct as the level of Gal80 rises concomitantly with the level of Gal2 in our experimental

data. This example reveals that caution must be used when interpreting results from models that are incomplete. It also reveals the difference between statistical explanations of data and biological ones, a distinction which is elaborated upon in Section 8.1.1.

6.4 Discussion

The galactose example is intended to illustrate that expression array data can be quite useful in elucidating regulatory networks. While nine of the 52 experiments were carbon source time-series experiments, it should be noted that none of the 52 was performed with the goal of distinguishing between these two models. Nevertheless, they were successfully exploited to select the currently accepted model over the one that had previously been postulated to be true, as well as clarifying the degree and sign of the statistical dependencies between the variables in these data sets. As more experiments become available and more complex models are formulated, these methods will be able to distinguish between subtle differences in proposed models in ways that are not possible without computational assistance.

As discussed in detail in Chapter 4, there are certain limitations when using Bayesian networks for modeling genetic regulatory networks, the most important of these being the caution with which models must be interpreted. To review, while graphs are highly interpretable structures for representing statistical dependencies, they have the potential to be misleading if interpreted incorrectly. It is important to distinguish between statistical interaction and physical interaction.

For example, if the data strongly supports the inclusion of an edge between two variables X and Y, that may indicate a physical interaction between these two factors in the cell. Alternatively, it is possible that an unmodeled variable Z actually intermediates between X and Y, such that X and Y exhibit statistical dependence but no physical interaction. As the example in Section 6.3.2 indicates, caution must be used when interpreting models that may be missing critical explanatory variables.

In contrast, if the data strongly supports the exclusion of an edge between two variables X and Y, that may indicate an absence of physical interaction between these two factors

in the cell. Alternatively, we may not have observed the cell under an appropriate set of conditions where this interaction could have been observed. This is the case in Section 6.3.1 where there is not strong support for including an edge between Gal4m and Gal2m, though the two factors are known to interact in the cell.

Chapter 7

Modeling the yeast pheromone response regulatory network

While genomic expression data has proven tremendously useful in providing insights into cellular regulatory networks, other valuable sources of data are increasingly becoming available to aid in this process. The wide range of data modalities presents a significant challenge, but also an opportunity since principled fusion of these diverse information sources helps reveal synergistic insights not readily apparent when sources are examined individually. We attempt to tackle the information fusion challenge by developing principled methods for the automatic discovery of genetic regulatory network models from both genomic location and expression data. We combine genomic location and expression data to guide the model discovery process by permitting the former to influence the model prior and the latter the model likelihood.

The advantage of modeling a well-studied genetic regulatory network such as the yeast galactose network examined in the previous chapter is that our validation framework can be tested in a context where some semblance of ground truth is available, at least to the extent that such a thing is possible given our currently limited understanding of cellular regulatory mechanisms. The disadvantage, however, is that results obtained within our validation framework that are consistent with independently substantiated regulatory mechanisms are, by definition, not novel. Having already examined a simple regulatory network in the

previous chapter in order to offer a proof-of-concept for our validation framework, we now turn to the examination of a more complicated regulatory system in which much less is known about the structure of the corresponding regulatory network. While this offers us the possibility of unearthing new insights into the function of such a network, the results presented require independent substantiation in order to confirm their accuracy. This is the tradeoff in considering a less well-studied genetic regulatory network.

In this chapter, we consider the regulatory network responsible for controlling the expression of various genes that code for proteins involved in *Saccharomyces cerevisiae* pheromone response pathways. The protein Ste12 is the ultimate target of the pheromone response signaling pathway and binds DNA as a transcriptional activator for a number of other genes. Data from genomic location analysis indicates which intergenic regions in the yeast genome are bound by Ste12, both in the presence and in the absence of pheromone [99]. Because pheromone response and mating pathways play an essential role in the sexual reproduction of yeast and because we have access to location data regarding the binding locations of Ste12 within the yeast genome, this is a natural choice of regulatory network to examine after our previous consideration of the galactose regulatory network. We build upon the results of the previous chapter by extending our analysis to include model discovery rather than model validation. We address the danger of possible over-fitting during the model selection process by employing model averaging.

We begin this chapter by examining in Section 7.1 the collection and preparation of data for model discovery in the context of pheromone response. Section 7.2 recounts the model discovery implementations presented previously in Chapter 4. In Section 7.3, we present various results from a model selection context, including results from different search algorithms and the impact of using data from genomic location analysis to add edge constraints representing prior information. Section 7.4 extends these model selection results to the more principled context of model averaging. We conclude in Section 7.5 with a discussion of the various results presented in this chapter and how they can best be interpreted. We also offer some directions for future work.

7.1 Data preparation

A set of 320 samples of unsynchronized *Saccharomyces cerevisiae* populations of varying genotype were observed under a diversity of experimental conditions. The set of samples ranges widely but consists primarily of observations of various wild-type and mutant *S. cerevisiae* strains made under a variety of environmental conditions including exposure to different nutritive media as well as exposure to stresses like heat, oxidative species, excessive acidity, and excessive alkalinity.

Whole-genome expression data for each of these 320 observations was collected using Affymetrix Ye6100 GeneChips. These GeneChips are 50-micron Affymetrix chips and consequently, four chips are required to measure the expression of all 6135 genes in the *S. cerevisiae* genome. Thus, a total of 1280 GeneChips were used in collecting this data.

7.1.1 Expression data normalization

The reported *average difference* values from these 1280 Affymetrix GeneChips were normalized using the MAP spiked control normalization methods described in Chapter 2 of this dissertation and in Hartemink, *et al.* [55]. The output of this process was a 6135×320 matrix of normalized log expression values, one row for each gene in the yeast genome and one column for each experimental observation.

7.1.2 Expression data selection

From the 6135 genes on the yeast genome, 32 were selected either on the basis of their participation in the *S. cerevisiae* pheromone response signaling cascade or as being known to affect other aspects of the mating response in yeast. The descriptions of the roles of various genes and proteins in this section are compiled from information from a variety of sources [40, 60, 31, 126, 103, 99].

Components of the pheromone response signaling cascade include the transmembrane receptor peptides Ste2 and Ste3 (present only in MATa and MATα yeast strains respectively), the three components of the heterotrimeric G-protein Gpa1, Ste4, and Ste18 (Gα, Gβ, and Gγ, respectively), the mitogen-activated protein kinase (MAPK) Fus3, the MAPKK Ste7,

the MAPKKK Ste11, the scaffolding peptide Ste5 which holds together Fus3, Ste7, and Ste11 in a large complex, and the transcriptional activator Ste12 that is the primary target of the MAPK signaling cascade. Additional components include an alternative MAPK Kss1, the p21-activated protein kinase (PAK) Ste20, and the peptide Ste50 which has unknown function but is necessary for proper function of Ste11. A schematic depiction of the relationships between the proteins in this cascade is shown in Figure 7-1.

MFA1 and MFA2 code for the mating pheromone peptide in MATa cells called a-factor and MFALPHA1 and MFALPHA2 similarly code for the mating pheromone peptide in MATα cells called α-factor. The peptide Ste6 is responsible for the export of a-factor from MATa cells. Far1 is a substrate of Fus3 that leads to G_1 arrest and also is known to bind to Ste4 as part of a complex of proteins necessary for establishing the cell polarity required for shmoo formation after a mating signal has been received.

Fus1 is a protein required for cell fusion during mating. Aga1 is an anchor subunit of the a-agglutinin complex and mediates the attachment of Aga2, the a-agglutinin binding subunit involved in cell-cell adhesion during mating, to the cell surface. Sag1 is the counterpart of Aga2 in MATα cells and is, correspondingly, the α-agglutinin binding subunit involved in cell-cell adhesion (it is also known as Agα1). Aga2 is found only in MATa cells while Sag1 is found only in MATα cells; the two proteins bind one-to-one during mating.

The protease Bar1 degrades α-factor and is only produced in MATa cells. Sst2 is involved in desensitization to mating pheromone exposure while Kar3 is essential for the nuclear migration step of karyogamy.

A number of factors that are believed to regulate the expression of these various genes were also included among the 32 selected genes. Tec1 and Mcm1 are believed to bind cooperatively with the transcriptional activator Ste12 at various gene promoters, with Mcm1 thought to be more active during pheromone response and Tec1 thought to be more active during the induction of the filamentous (or invasive) growth response.[1] Tup1, Sin3, Snf2, and Swi1 have all been implicated in the induction or repression of numerous genes in the pheromone response pathway. In particular, Snf2 and Swi1 are both associated with the

[1]Filamentation occurs under conditions of nutrient deprivation and shares a large part of the signaling cascade responsible for transduction of a pheromone signal, including Ste11, Ste7, Ste50, Ste20, and Ste12, but with Kss1 substituting for Fus3 as the predominant MAPK protein.

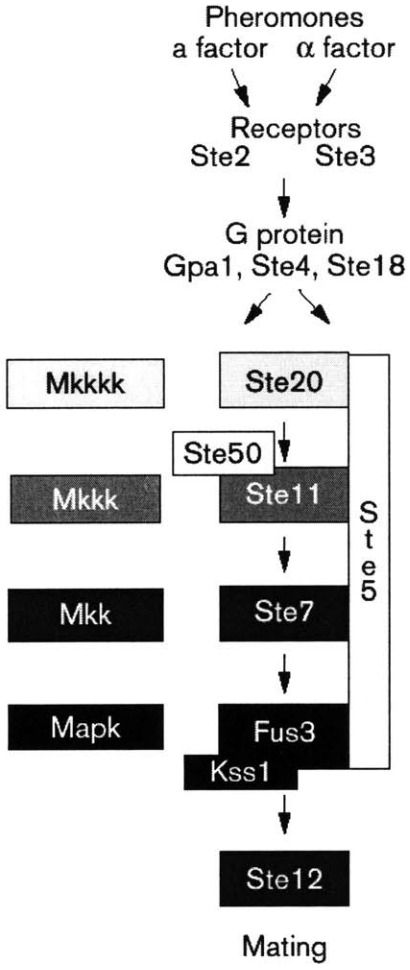

Figure 7-1: *Schematic representation of the various molecular factors believed to play a role in the pheromone response signaling cascade in* S. cerevisiae *and supposed physical relationships. This figure was taken directly from a recent review on the subject of pheromone response by Elion [40].*

SWI-SNF global transcription activator complex.

As a mnemonic device, the variables in the Bayesian network figures presented throughout this chapter are colored according to the relationships just outlined. Genes expressed only in MATa cells are colored dark blue (MFA1, MFA2, STE2, STE6, AGA2, BAR1), genes expressed only in MATα cells are colored red (MFALPHA1, MFALPHA2, STE3, SAG1), genes whose promoters are bound by Ste12 are colored cyan (STE12, FAR1, AGA1, FUS1, FUS3), genes coding for components of the heterotrimeric G-protein complex are colored bright green (GPA1, STE4, STE18), genes coding for core components of the primary signaling cascade complex are colored yellow[2] (STE5, STE7, STE11), genes coding for auxiliary or alternate components of the signaling cascade are colored magenta (STE50, STE20, KSS1), and genes whose protein products form part of the SWI-SNF complex are colored orange (SWI1, SNF2).

7.1.3 Expression data discretization

The normalized levels of expression for these 32 genes were extracted from the 6135×320 normalization output matrix to yield a matrix of data with 32 rows and 320 columns, one row for each gene and one column for each observation. This data was then log-transformed and discretized using the methods presented in Chapter 3. The information-preserving discretization level coalescence algorithm was initialized with stochastic quantile discretization of degree 160 (initial discretization intervals for each gene included two observations apiece). The total mutual information retained at each stage in the discretization level coalescence process is shown in Figure 7-2. As is evident from the figure, almost no mutual information is lost in the coalescence process until only a small number of discretization levels remain. Based on the results shown in this figure, the data were finally discretized to have four discretization levels for each gene.

In addition to these 32 variables representing discretized levels of log gene expression, an additional variable named `mating_type` was considered. The variable `mating_type` represents the mating type of the various haploid strains of yeast used in the 320 observations and can take one of two values, corresponding to the MATa and MATα mating types of

[2]FUS3 should also be in this group but is already colored cyan.

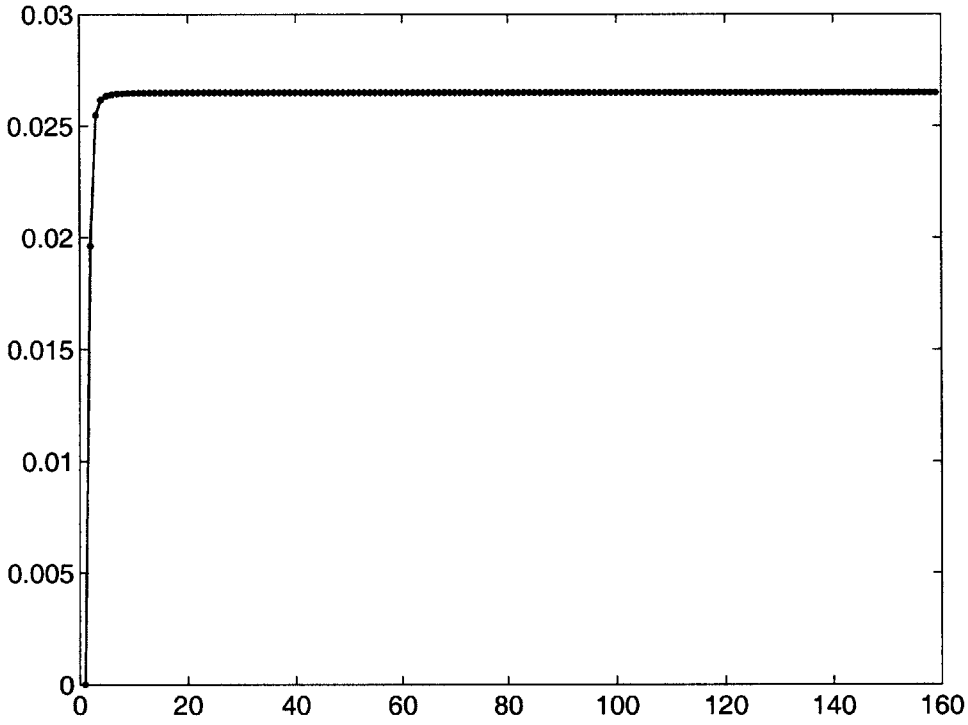

Figure 7-2: *Graph showing the total mutual information preserved during the discretization level coalescence process as a function of the number of discretization levels remaining. The long plateau indicates that even a small number of discrete levels of gene expression is likely sufficient to capture statistically predictive relationships between these 32 variables.*

yeast. The inclusion of this variable is necessary because, *e.g.*, the MFA1 and MFA2 genes responsible for producing the mating pheromone a-factor are expressed only in MATa strains of yeast.

The data used as input for the various model discovery algorithms was thus a matrix of 33 rows and 320 columns, 32 rows representing the discretized levels of log expression for 32 genes involved in pheromone response and one row representing the mating type of the strain in each experiment, either MATa or MATα.

7.1.4 Location data

As alluded to above, data from genomic location analysis of Ste12 binding was also available. This data, gathered using a chromatin immunoprecipitation assay, revealed the genes in the yeast genome whose upstream regions were bound by Ste12 under both presence and absence of pheromone. Of the 32 pheromone response genes examined in this research, STE12, FUS1, FUS3, AGA1, and FAR1 promoters are all bound by Ste12, the first three being bound significantly both before and after the addition of pheromone, and the latter two being bound significantly only after the addition of pheromone. A description of the assay and a more detailed presentation of the results can be found in the paper by Ren, *et al.* [99].

7.2 Model discovery implementations

For this research, we utilized the two model discovery implementations presented in Section 4.4.2 of Chapter 4. In particular, one implementation is a greedy random search algorithm with random restarts and the other is a simulated annealing search algorithm with a temperature schedule that allows for "reannealing" after the temperature becomes sufficiently low. In each case, we use as our prior distribution over parameters the uninformative Dirichlet prior of Buntine [16] (which Heckerman, *et al.* [59] call BDeu).

Both of these implementations were further extended to be able to search for models with constraints specifying which edges are required to be present and which are required to be absent. This allows for the incorporation of prior information about edges in the graph

since the constrained search algorithms are equivalent to the corresponding unconstrained search algorithms with a nonuniform prior over structures that gives zero weight to models that either include edges required to be absent or do not include edges required to be present. In this way, data from other sources such as location analysis can be easily incorporated.

7.3 Model selection results

7.3.1 Models selected using greedy random search algorithm

First, we employed the greedy random search algorithm, both with and without constraints on the edges present, in the context of model selection. Figure 7-3 shows the highest scoring Bayesian network model learned without any constraints on the edges. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature, as mentioned above. Edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission, defined as the factor difference between the score of the model with the edge present and a corresponding model that has the edge removed. Edges are colored black if their removal lowers the posterior probability of the model by a factor of more than a billion, colored purple if their removal lowers the posterior probability by a factor between a billion and a million, colored dark blue if their removal lowers the posterior probability by a factor between a million and a thousand, and colored light blue if their removal lowers the posterior probability by a factor between a thousand and one. Precise values for the edge strengths can be found in Table 7.1.

Clearly this metric is not the most accurate method for assessing the posterior probability that the edge should be present in the graph. The proper Bayesian way to compute the probability that an edge is necessary to explain the observed data is to integrate an index function for the presence of the edge over the entire posterior probability distribution over graph structures, a method we discuss in Section 7.4. Since such a posterior is difficult to compute with only a single model, we rely on this simple probability change metric here to provide a rough idea of how significantly an edge explains the observed data.

Figure 7-4 shows the highest scoring Bayesian network model learned with the greedy

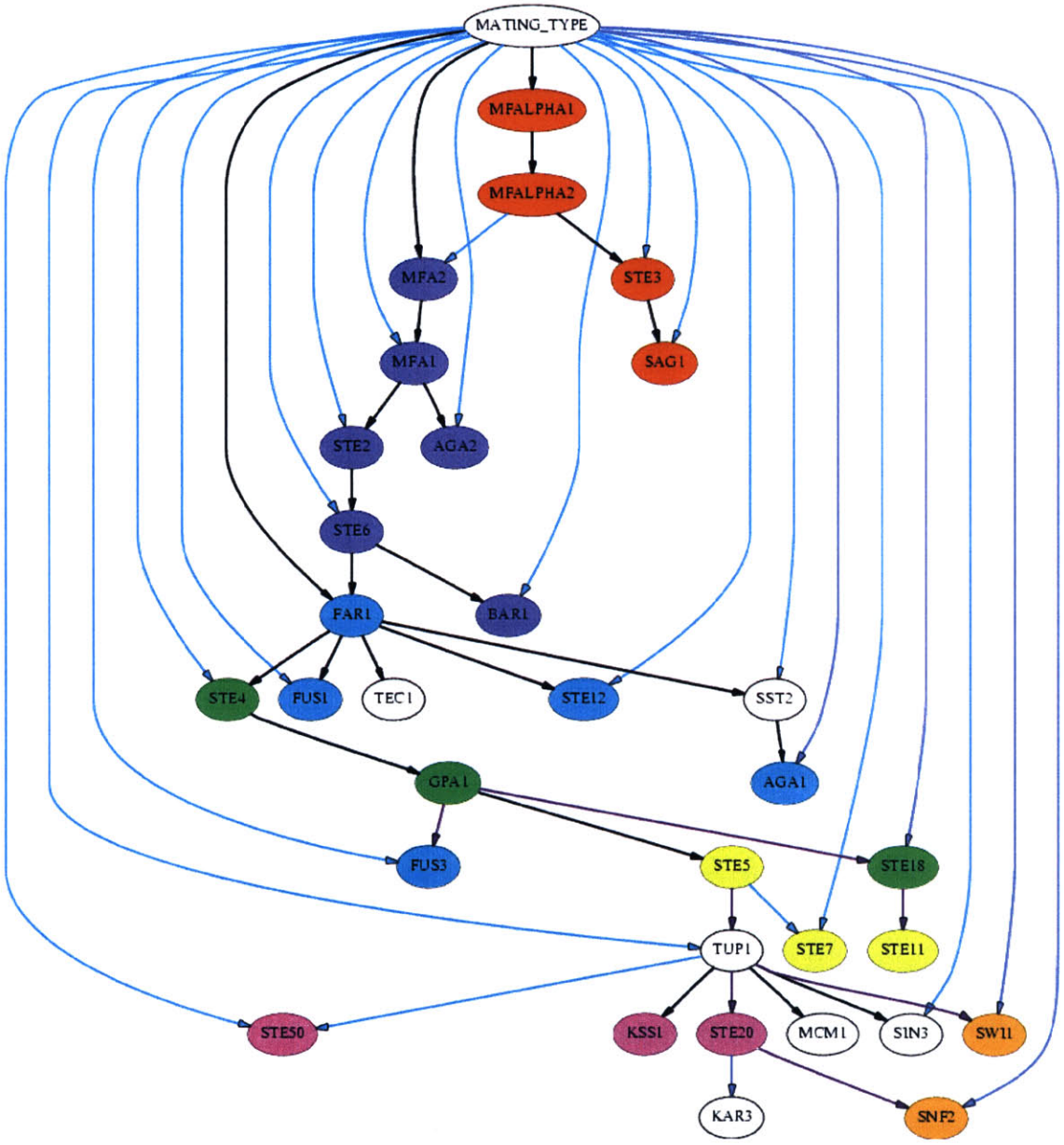

Figure 7-3: *Bayesian network model for representing the probabilistic dependencies between the 33 variables related to pheromone response in yeast as learned by the greedy random search algorithm with random restarts. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Node and edge color descriptions are included in the text. Precise values for the edge strengths can be found in Table 7.1.*

Table 7.1: *Strengths of the edges in the Bayesian network model shown in Figure 7-3 in terms of the probability change associated with their omission.*

From	To	Prob. Change	From	To	Prob. Change
STE4	GPA1	1.54073e+42	GPA1	FUS3	5.96105e+06
MATING_TYPE	MFALPHA1	2.38558e+39	STE18	STE11	3.28898e+06
FAR1	TEC1	7.98454e+38	STE5	TUP1	1.05465e+06
MFALPHA1	MFALPHA2	4.37215e+37	STE20	KAR3	9.19625e+05
MFA1	AGA2	1.37967e+37	MATING_TYPE	SWI1	15442.5
MFA1	STE2	1.14447e+35	MATING_TYPE	AGA1	8963.4
STE2	STE6	9.95713e+34	MATING_TYPE	STE18	5448.2
FAR1	STE4	1.98907e+33	MATING_TYPE	SNF2	1100.6
STE6	BAR1	1.17614e+29	MATING_TYPE	BAR1	869.2
MFA2	MFA1	1.64737e+26	TUP1	STE50	813.8
SST2	AGA1	4.31432e+25	MATING_TYPE	SIN3	380.0
STE6	FAR1	2.46397e+25	MATING_TYPE	TUP1	339.1
MATING_TYPE	MFA2	2.62561e+24	MATING_TYPE	STE3	189.8
FAR1	SST2	2.22539e+23	MATING_TYPE	MFA1	88.9
FAR1	FUS1	1.52819e+21	MATING_TYPE	STE4	71.4
FAR1	STE12	2.00733e+20	MFALPHA2	MFA2	56.4
GPA1	STE5	2.21753e+16	MATING_TYPE	STE2	54.8
TUP1	MCM1	3.83958e+15	MATING_TYPE	FUS3	43.7
TUP1	SIN3	2.34004e+12	MATING_TYPE	STE6	39.3
MFALPHA2	STE3	2.00055e+11	MATING_TYPE	FUS1	22.7
TUP1	KSS1	1.49781e+10	MATING_TYPE	AGA2	15.8
STE3	SAG1	1.36904e+10	STE5	STE7	9.5
MATING_TYPE	FAR1	1.19567e+09	MATING_TYPE	SST2	9.1
TUP1	STE20	8.84203e+08	MATING_TYPE	SAG1	8.5
GPA1	STE18	4.78989e+08	MATING_TYPE	STE7	2.5
STE20	SNF2	2.83481e+07	MATING_TYPE	STE50	2.3
TUP1	SWI1	1.98094e+07	MATING_TYPE	STE12	1.1

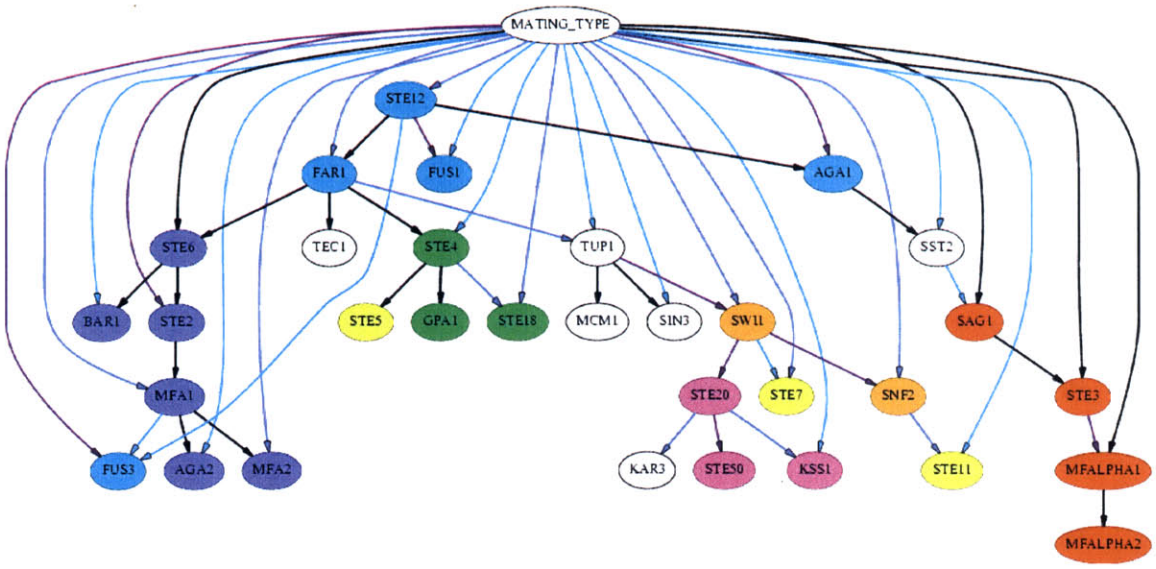

Figure 7-4: *Bayesian network model for representing the probabilistic dependencies between the 33 variables related to pheromone response in yeast as learned by the constrained greedy random search algorithm with random restarts. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Node and edge color descriptions are included in the text. Precise values for the edge strengths can be found in Table 7.2.*

random algorithm when constraints on the edges have been added based on genomic analysis of Ste12 DNA binding location. Constraints governing the inclusion and exclusion of edges in this case were derived from location analysis data that indicated the presence of Ste12 upstream of the promoters for FUS1, FUS3, AGA1, and FAR1. These edges were required to be present in the graph by modifying the prior to have no support for graphs that were missing these edges. As before, nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature and edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Precise values for the edge strengths can be found in Table 7.2.

Table 7.2: *Strengths of the edges in the Bayesian network model shown in Figure 7-4 in terms of the probability change associated with their omission.*

From	To	Prob. Change	From	To	Prob. Change
STE4	GPA1	1.54073e+42	STE3	MFALPHA1	5.16312e+06
FAR1	TEC1	7.98454e+38	STE20	STE50	2.45621e+06
MATING_TYPE	STE6	2.12412e+38	STE20	KAR3	9.19625e+05
MFALPHA1	MFALPHA2	4.37215e+37	MATING_TYPE	STE7	1.85836e+05
MFA1	AGA2	1.37967e+37	MATING_TYPE	STE12	1.32493e+05
STE2	MFA1	1.14447e+35	MATING_TYPE	STE18	42421.3
STE6	STE2	9.95713e+34	STE4	STE18	33273.2
FAR1	STE4	1.98907e+33	STE20	KSS1	19361.0
STE6	BAR1	1.17614e+29	MATING_TYPE	SWI1	15442.5
MFA1	MFA2	1.64737e+26	MATING_TYPE	SNF2	13892.7
AGA1	SST2	4.31432e+25	FAR1	TUP1	4060.9
FAR1	STE6	2.46397e+25	MATING_TYPE	MFA2	3669.4
STE12	FAR1	2.00733e+20	MATING_TYPE	FAR1	2001.8
STE4	STE5	9.47051e+15	SNF2	STE11	1134.0
TUP1	MCM1	3.83958e+15	MATING_TYPE	MFA1	1029.2
MATING_TYPE	MFALPHA1	1.43435e+15	MATING_TYPE	BAR1	869.2
MATING_TYPE	STE3	6.21574e+14	MATING_TYPE	SIN3	380.0
STE12	AGA1	9.81284e+12	MATING_TYPE	KSS1	310.0
TUP1	SIN3	2.34004e+12	MATING_TYPE	TUP1	266.8
SAG1	STE3	1.36904e+10	MATING_TYPE	SST2	151.5
MATING_TYPE	SAG1	9.91076e+09	MATING_TYPE	STE4	71.4
SWI1	STE20	6.10164e+08	MATING_TYPE	FUS1	52.6
STE12	FUS1	7.92043e+07	SWI1	STE7	18.8
TUP1	SWI1	1.98094e+07	MATING_TYPE	AGA2	15.8
SWI1	SNF2	1.37560e+07	SST2	SAG1	4.8
MATING_TYPE	FUS3	7.81302e+06	MFA1	FUS3	3.8
MATING_TYPE	STE2	7.45139e+06	MATING_TYPE	STE11	2.7
MATING_TYPE	AGA1	5.35876e+06	STE12	FUS3	0.1

7.3.2 Models selected using simulated annealing search algorithm

We next employed the simulated annealing search algorithm, both with and without constraints on the edges present, in the context of model selection. Figure 7-5 shows the highest scoring Bayesian network model learned without any constraints on the edges. Once again, nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature and edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Precise values for the edge strengths can be found in Table 7.3.

Figure 7-6 shows the highest scoring Bayesian network model learned with the simulated annealing algorithm when constraints on the edges have been added based on genomic analysis of Ste12 DNA binding location. Once again, constraints governing the inclusion and exclusion of edges in this case were derived from location analysis data that indicated the presence of Ste12 upstream of the promoters for FUS1, FUS3, AGA1, and FAR1. These edges were required to be present in the graph by modifying the prior to have no support for graphs that were missing these edges. As before, nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature and edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Precise values for the edge strengths can be found in Table 7.4.

7.3.3 Model properties and comparison

When we use the Bayesian scoring metric to compute the scores of the four networks shown in Figures 7-3 through 7-6, we get scores of -8181.93, -8281.7, -8161.18, and -8204.23, respectively. We observe that the simulated annealing algorithm finds better scoring models than the greedy random algorithm under the same set of constraints (or lack thereof). We also observe that the constraints imposed on the networks reduce the scores of the resultant models quite significantly. However, it must be remembered that the unconstrained scores are computed on the basis of the model's ability to explain only the expression data and not also the binding location data. If binding location data is incorporated in the score by giving

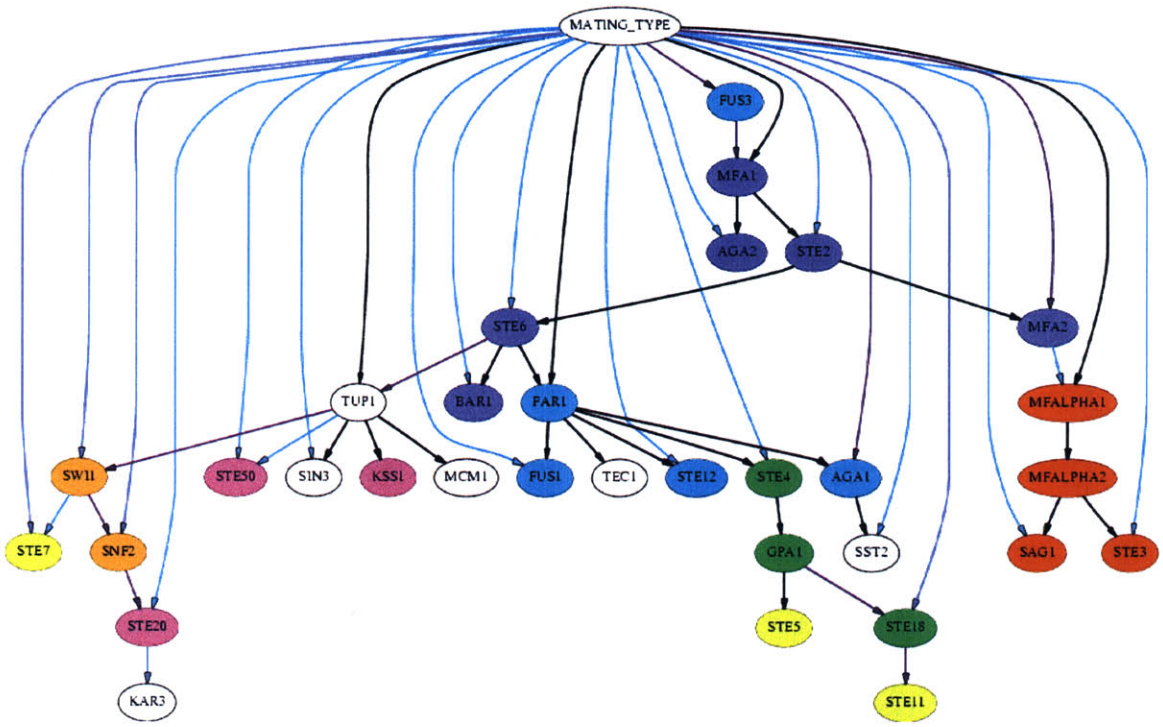

Figure 7-5: *Bayesian network model for representing the probabilistic dependencies between the 33 variables related to pheromone response in yeast as learned by the simulated annealing search algorithm with reannealing. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Node and edge color descriptions are included in the text. Precise values for the edge strengths can be found in Table 7.3.*

Table 7.3: *Strengths of the edges in the Bayesian network model shown in Figure 7-5 in terms of the probability change associated with their omission.*

From	To	Prob. Change	From	To	Prob. Change
MATING_TYPE	MFA1	3.67281e+45	SNF2	STE20	2.83481e+07
STE4	GPA1	1.54073e+42	TUP1	SWI1	1.98094e+07
FAR1	TEC1	7.98454e+38	SWI1	SNF2	1.37560e+07
MFALPHA1	MFALPHA2	4.37215e+37	FUS3	MFA1	1.06188e+07
MFA1	AGA2	1.37967e+37	MATING_TYPE	AGA1	4.32373e+06
MFA1	STE2	1.14447e+35	STE18	STE11	3.28898e+06
STE2	STE6	9.95713e+34	STE20	KAR3	9.19625e+05
FAR1	STE4	1.98907e+33	MATING_TYPE	STE7	1.85836e+05
STE6	BAR1	1.17614e+29	MATING_TYPE	SWI1	15442.5
STE2	MFA2	9.75103e+27	MATING_TYPE	SNF2	13892.7
AGA1	SST2	4.31432e+25	MATING_TYPE	STE18	5448.2
STE6	FAR1	2.46397e+25	MFA2	MFALPHA1	985.1
FAR1	AGA1	1.12857e+25	MATING_TYPE	BAR1	869.2
FAR1	FUS1	1.52819e+21	TUP1	STE50	813.8
FAR1	STE12	2.00733e+20	MATING_TYPE	SIN3	380.0
GPA1	STE5	2.21753e+16	MATING_TYPE	SAG1	217.3
TUP1	MCM1	3.83958e+15	MATING_TYPE	STE3	189.8
TUP1	SIN3	2.34004e+12	MATING_TYPE	SST2	151.5
MFALPHA2	STE3	2.00055e+11	MATING_TYPE	STE4	71.4
MFALPHA2	SAG1	2.66946e+10	MATING_TYPE	STE2	54.8
MATING_TYPE	TUP1	1.61478e+10	MATING_TYPE	STE6	39.3
TUP1	KSS1	1.49781e+10	MATING_TYPE	STE20	27.3
MATING_TYPE	MFALPHA1	1.52661e+09	MATING_TYPE	FUS1	22.7
MATING_TYPE	FAR1	1.19567e+09	SWI1	STE7	18.8
GPA1	STE18	4.78989e+08	MATING_TYPE	AGA2	15.8
MATING_TYPE	MFA2	3.05016e+08	MATING_TYPE	STE50	2.3
MATING_TYPE	FUS3	2.56035e+08	MATING_TYPE	STE12	1.1
STE6	TUP1	4.83781e+07			

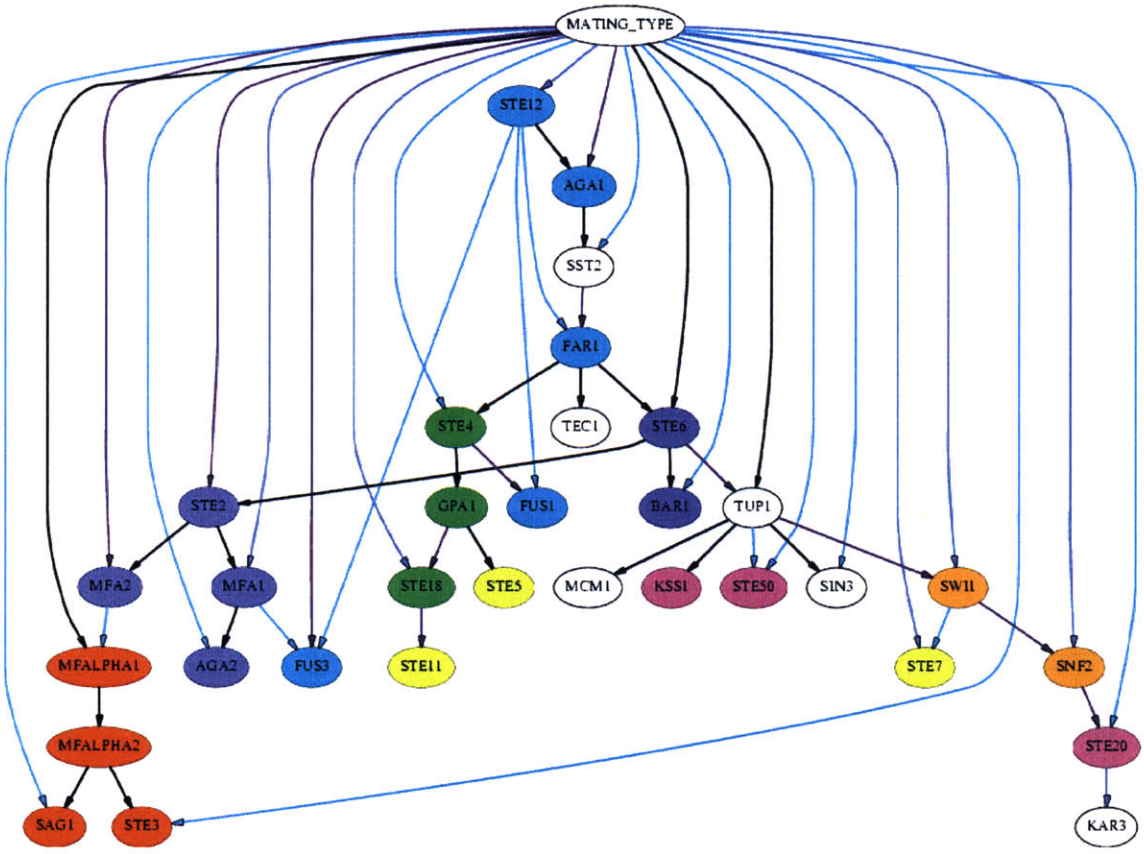

Figure 7-6: *Bayesian network model for representing the probabilistic dependencies between the 33 variables related to pheromone response in yeast as learned by the simulated annealing search algorithm with reannealing. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges have been augmented with color information to indicate the relative strengths of the edges in terms of the probability change associated with their omission. Node and edge color descriptions are included in the text. Precise values for the edge strengths can be found in Table 7.4.*

Table 7.4: *Strengths of the edges in the Bayesian network model shown in Figure 7-6 in terms of the probability change associated with their omission.*

From	To	Prob. Change	From	To	Prob. Change
STE4	GPA1	1.54073e+42	MATING_TYPE	STE2	7.45139e+06
FAR1	TEC1	7.98454e+38	MATING_TYPE	AGA1	5.35876e+06
MATING_TYPE	STE6	2.12412e+38	STE18	STE11	3.28898e+06
MFALPHA1	MFALPHA2	4.37215e+37	STE4	FUS1	2.76045e+06
MFA1	AGA2	1.37967e+37	SST2	FAR1	1.48645e+06
STE2	MFA1	1.14447e+35	STE20	KAR3	9.19625e+05
STE6	STE2	9.95713e+34	MATING_TYPE	STE7	1.85836e+05
FAR1	STE4	1.98907e+33	MATING_TYPE	STE12	1.32493e+05
STE6	BAR1	1.17614e+29	MATING_TYPE	SWI1	15442.5
STE2	MFA2	9.75103e+27	MATING_TYPE	SNF2	13892.7
AGA1	SST2	4.31432e+25	MATING_TYPE	STE18	5448.2
FAR1	STE6	2.46397e+25	MATING_TYPE	MFA1	1029.2
GPA1	STE5	2.21753e+16	MFA2	MFALPHA1	985.1
TUP1	MCM1	3.83958e+15	MATING_TYPE	BAR1	869.2
STE12	AGA1	9.81284e+12	TUP1	STE50	813.8
TUP1	SIN3	2.34004e+12	MATING_TYPE	SIN3	380.0
MFALPHA2	STE3	2.00055e+11	MATING_TYPE	SAG1	217.3
MFALPHA2	SAG1	2.66946e+10	MATING_TYPE	STE3	189.8
MATING_TYPE	TUP1	1.61478e+10	MATING_TYPE	SST2	151.5
TUP1	KSS1	1.49781e+10	MATING_TYPE	STE4	71.4
MATING_TYPE	MFALPHA1	1.52661e+09	MATING_TYPE	STE20	27.3
GPA1	STE18	4.78989e+08	SWI1	STE7	18.8
MATING_TYPE	MFA2	3.05016e+08	MATING_TYPE	AGA2	15.8
STE6	TUP1	4.83781e+07	MFA1	FUS3	3.8
SNF2	STE20	2.83481e+07	MATING_TYPE	STE50	2.3
TUP1	SWI1	1.98094e+07	STE12	FUS3	0.1
SWI1	SNF2	1.37560e+07	STE12	FAR1	0.1
MATING_TYPE	FUS3	7.81302e+06	STE12	FUS1	7.8115e-07

the corresponding informative prior sufficiently little support for models that are missing edges required to be present, then scores will accurately reflect the ability to explain both sets of data. For example, in our case where the prior probability of a structure without these required edges is zero (an extreme assumption to be sure), the Bayesian score of such a model is actually negative infinity.

Although Tables 7.1 and 7.3 contain only edges with strength greater than one (their removal decreases the overall score of the model, as expected), we notice that in Tables 7.2 and 7.4 there appear edges whose strength is less than one (their removal increases the overall score of the model). This may seem a counter-intuitive result but we note that each of these edges has Ste12 as a parent — these edges were required to be included in the graph on the basis of the Ste12 binding location analysis information collected from Ren, *et al.* [99]. Although the scores of these constrained models are lower than those that remove some of the edges from Ste12, the same reasoning as above applies: if the prior probability of a structure missing these required edges is zero, the probability change associated with their omission would be positive infinity.

In each of the networks presented in Figures 7-3 through 7-6, we observe a number of interesting properties. In all four cases, the `mating_type` variable is at the root of the tree, and contributes to the ability to predict the state of a large number of variables, which is to be expected. However, while many links exist between `mating_type` and other variables, they are for the most part among the weakest links, with the possible exception of the ones with genes known to be expressed only in MATa or MATα strains. In particular, in all four networks, highly significant (black) edges exist between at least one MATa-only gene and one MATα-only gene (always MFALPHA1; always at least one of MFA1, MFA2, or STE6; sometimes STE3 or SAG1). We also note that in all four cases there exists a directly connected subgraph consisting of genes expressed only in MATa cells (dark blue) and a directly connected subgraph consisting of genes expressed only in MATα cells (red). In each case the subgraph has the `mating_type` variable as a direct ancestor with strong predictive power, as expected.

Furthermore, in all four networks, the heterotrimeric G-protein complex components GPA1, STE4, and STE18 (bright green) form a directly connected component in the graph.

In three of four cases, SWI1 and SNF2 (orange) are adjacent, but in the fourth case, they are instead close descendants of TUP1 (as they are in the other three cases as well). The core elements of the primary signaling cascade complex (yellow) are frequently seen as descendants of G-protein complex genes, indicating statistical dependence that may be the result of common or serial regulatory control. Auxiliary signaling cascade genes (magenta) are always descendants of TUP1, sometimes directly and sometimes more indirectly, but STE50 and KSS1 are siblings in all four cases. In general, the auxiliary cascade elements STE20, STE50, and KSS1 do not tend to cluster with the core elements, indicating that the regulation of their transcript levels may occur by a different mechanism than those of the genes in the core signal transduction complex.

TUP1 is the gene with the most children across all four networks, consistent with its role as a general repressor of RNA polymerase II transcription. All four networks have MCM1 and SIN3 as children of TUP1; Tup1 and Mcm1 are known to interact in the cell [46] and this result that the level of Tup1 is helpful in predicting the level of Mcm1 suggests a possible regulatory relationship between the two. FAR1 is the gene with the second highest number of children across all four networks, consistent with its role in cell cycle arrest. FAR1 is a parent of TEC1 and STE4 in all four networks. Far1 and Tec1 are both known to be cell-cycle regulated and both are classified as being transcribed during early G_1 phase [26]; the same paper indicates that GPA1, a child of STE4 in all four networks, is also transcribed during early G_1 phase. Far1 is known to lead to G_1 arrest and also is known to bind to Ste4 as part of a complex of proteins necessary for establishing the cell polarity required for shmoo formation after a mating signal has been received.

Though it is produced at higher levels in MATa cells, it is known that Aga1 is produced in both MATa and MATα cells: "the AGA1 transcript was expressed and induced by pheromone in both a and alpha cells, suggesting that the a-specific expression of active a-agglutinin results only from a-specific regulation of [AGA2]" [105]. The graphs are all consistent with this knowledge, including a reasonably strong predictive edge from `mating_type` to AGA1, but not clustering AGA1 with other mating type specific genes (dark blue and red) as it is likely regulated differently. In all cases, AGA1 and SST2 are adjacent, consistent with the fact that the two are expressed very similarly, both peaking at the M/G_1

phase of the cell-cycle [109].

It should be remembered that the edges in these networks indicate a statistical dependence between the transcript levels of the genes, but do not necessarily specify the form or presence of a biochemical dependence. That is, there may be a statistical interaction that is non-physical. For example, in three of the four networks, a weak link appears between MFA2 and either MFALPHA1 or MFALPHA2. Although these mating factors are never both expressed in these haploid *S. cerevisiae* strains, cells expressing a lot of one are less likely, statistically, to be expressing a lot of the other; hence the link. The fact that the link is weak indicates that other variables such as `mating_type` successfully explain away most of this statistical dependence and the weak edge is therefore likely the result of uncaptured residual dependence, but the interpretation *caveat* remains.

To address the concern of basing our assessments of edge presence on a single selected model, we now consider the context of model averaging.

7.4 Model averaging results

One concern we have with the model selection process is that it is likely over-fitting the data by selecting the single maximum a posteriori model and ignoring completely other models that score nearly as well. Recall from the discussion in Chapter 4 that a more principled Bayesian approach is to compute probabilities of features of interest by averaging over the posterior model distribution rather than relying on a single model in isolation. For example, if we are interested in determining whether the data supports the inclusion of an edge representing statistical dependence between two variables X and Y, the Bayesian approach is to compute:

$$\mathrm{p}(E_{XY} \mid D) = \sum_S \mathrm{p}(E_{XY}, S \mid D) \tag{7.1}$$

$$= \sum_S \mathrm{p}(E_{XY} \mid D, S) \cdot \mathrm{p}(S \mid D) \tag{7.2}$$

$$= \sum_S 1_{XY}(S) \cdot \mathrm{e}^{\mathsf{BayesianScore}(S)} \tag{7.3}$$

where E_{XY} represents an edge from variable X to variable Y, $1_{XY}(S)$ is an indicator function that is one if and only if graph S includes E_{XY} as an edge, and $\mathsf{BayesianScore}(S)$ is the Bayesian scoring metric for graph S. However this sum is difficult to compute because the space of graphs S is enormous. Fortunately, it is possible to approximate this sum since the vast bulk of its mass lies among the highest scoring models; the exponential factor in the sum has the effect of drowning out all but the highest scoring models, even though they are relatively infrequent. For example, if we restrict our attention to the N highest scoring models, and index these by the variable i, then we have:

$$\mathrm{p}(E_{XY} \mid D) \approx \frac{\sum_{i=1}^{N} 1_{XY}(S_i) \cdot \mathrm{e}^{\mathsf{BayesianScore}(S_i)}}{\sum_{i=1}^{N} \mathrm{e}^{\mathsf{BayesianScore}(S_i)}} \tag{7.4}$$

We implemented such a model averaging strategy based on our simulated annealing search algorithm. We used the simulated annealing search implementation to visit high-scoring regions of the model posterior and present the results of two of those runs here. In the first run, we traversed the model space without constraints on the graph edges. In the second run, we incorporated the available location data by requiring edges from STE12 to FUS1, FUS3, AGA1, and FAR1. The top and center histograms in Figure 7-7 show the distributions of scores for all models visited during the unconstrained and constrained simulated annealing runs, respectively. For comparison, the bottom histogram in Figure 7-7 shows the distribution of scores for all models visited when we perform a lengthy random walk through the space of models, accepting every proposed local change (equivalent to infinite-temperature Metropolis). From this figure, we see that the simulated annealing algorithm is quite effective in gradually concentrating its efforts on extremely high scoring models.

We also observe as before that the constraints imposed on the networks reduce the scores of the resultant models. However, it must again be remembered that the unconstrained score is computed on the basis of the model's ability to explain only the expression data and not also the binding location data. If binding location data is incorporated in the score by giving the corresponding informative prior sufficiently little support for models that are missing

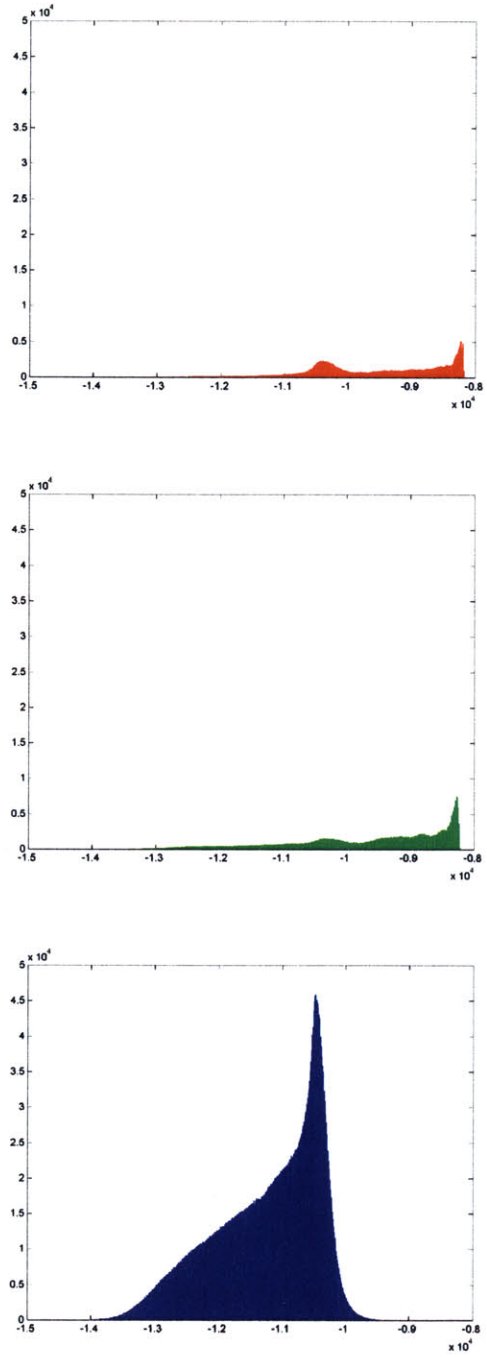

Figure 7-7: *Histograms of scores for all models visited during simulated annealing runs. The top and center histograms are for the unconstrained and constrained simulated annealing runs, respectively. For comparison, the bottom histogram was generated by a random walk through the space of models, accepting every proposed local change.*

edges required to be present, then scores will accurately reflect the ability to explain both sets of data.

After gathering the five hundred highest scoring models that were visited during each run of the search algorithm, we computed the probability of edges being present by using the weighted average approximation shown in Equation 7.4 (with $N = 500$). Results of this computation for the unconstrained and constrained searches are presented in Tables 7.5 and 7.6, respectively. We then compiled a composite network for each of these that consists of all edges with estimated posterior probability over 0.5. These networks are shown in Figures 7-8 and 7-9. Edges colored black have posterior probability of 1, edges colored purple have posterior probability between 1 and 0.99, edges colored dark blue have posterior probability between 0.99 and 0.75, and edges colored light blue have posterior probability between 0.75 and 0.5. In particular, the strengths of edges have a different interpretation than they did before. Whereas previously the strength of an edge indicated a rough measure of how *significantly* a parent node contributes to the ability to explain the child node, now the strength of an edge indicates an approximate measure of how *likely* a parent node is to contribute to the ability to explain the child node.

In both of the composite networks presented in Figures 7-8 and 7-9, we observe a number of interesting properties, many of which are common with those identified previously in the context of model selection. In both composite networks, the `mating_type` variable is at the root of the tree, and contributes to the ability to predict the state of a large number of variables, which is to be expected. The links are generally quite strong indicating that their presence was fairly uniform among the five hundred highest scoring models. Almost all the links with genes known to be expressed only in MATa or MATα strains occur with posterior probability above 0.99. We also note that in both networks there exists a directly connected subgraph consisting of genes expressed only in MATa cells (dark blue) and a directly connected subgraph consisting of genes expressed only in MATα cells (red). In each case the subgraph has the `mating_type` variable as a direct ancestor with strong predictive power, as expected.

The heterotrimeric G-protein complex components GPA1, STE4, and STE18 (bright green) form a directly connected component in the constrained graph but only GPA1 and

Table 7.5: *Posterior probabilities of edges being present in the unconstrained simulated annealing search as estimated by a weighted average over the five hundred highest scoring models.*

From	To	Posterior Prob.	From	To	Posterior Prob.
MATING_TYPE	MFA1	1.0000000	MATING_TYPE	STE4	0.9310940
STE6	TUP1	1.0000000	MATING_TYPE	SAG1	0.9191640
MATING_TYPE	TUP1	1.0000000	MATING_TYPE	STE50	0.8662340
FAR1	FUS1	1.0000000	MFA1	MFA2	0.6304400
TUP1	SWI1	1.0000000	SWI1	STE7	0.6292930
MATING_TYPE	SWI1	1.0000000	STE18	STE11	0.6159960
MATING_TYPE	MFALPHA1	1.0000000	SWI1	SNF2	0.5659030
MATING_TYPE	SNF2	1.0000000	MATING_TYPE	STE20	0.5551610
TUP1	SIN3	1.0000000	SNF2	STE20	0.5533950
MATING_TYPE	SIN3	1.0000000	STE20	SNF2	0.4340970
FAR1	AGA1	1.0000000	TUP1	STE20	0.4154630
MATING_TYPE	AGA1	1.0000000	STE50	STE11	0.3824810
MATING_TYPE	MFA2	1.0000000	STE2	MFA2	0.3695600
TUP1	MCM1	1.0000000	MATING_TYPE	STE12	0.2741810
MATING_TYPE	SST2	1.0000000	MATING_TYPE	KSS1	0.1964160
FAR1	TEC1	1.0000000	STE5	STE7	0.1710230
GPA1	STE18	1.0000000	STE50	STE7	0.1708470
MATING_TYPE	STE18	1.0000000	MATING_TYPE	MFALPHA2	0.0708669
STE6	FAR1	1.0000000	MFALPHA2	MFALPHA1	0.0596000
MATING_TYPE	FAR1	1.0000000	STE3	MFALPHA2	0.0596000
STE6	BAR1	1.0000000	GPA1	STE4	0.0594723
MATING_TYPE	BAR1	1.0000000	MFALPHA1	MFA1	0.0569844
FAR1	STE12	1.0000000	MFA1	FUS3	0.0339687
TUP1	KSS1	1.0000000	SWI1	STE20	0.0311423
MFA1	STE2	0.9998720	FUS3	MFA1	0.0255037
STE4	STE5	0.9998720	STE11	STE7	0.0090948
AGA1	SST2	0.9998720	MATING_TYPE	STE11	0.0017818
STE2	STE6	0.9998720	STE2	MFALPHA1	0.0016013
STE3	SAG1	0.9998720	SNF2	STE11	0.0015236
FAR1	GPA1	0.9998720	MFA1	MFALPHA1	0.0005666
MFA1	AGA2	0.9998720	AGA2	MFALPHA1	0.0003554
TUP1	STE50	0.9997690	STE11	KAR3	0.0002888
STE20	KAR3	0.9997110	STE6	STE2	0.0001277
MATING_TYPE	STE3	0.9995500	AGA2	MFA1	0.0001277
MATING_TYPE	GPA1	0.9994340	GPA1	STE5	0.0001277
MATING_TYPE	STE2	0.9979080	FAR1	SST2	0.0001277
MATING_TYPE	FUS3	0.9965830	SAG1	STE3	0.0001277
MATING_TYPE	STE6	0.9955280	SST2	SAG1	0.0001277
MATING_TYPE	FUS1	0.9937100	STE4	GPA1	0.0001277
MATING_TYPE	AGA2	0.9850980	STE2	AGA2	0.0001277
MATING_TYPE	STE7	0.9753010	AGA2	FUS3	0.0001277
FAR1	STE4	0.9405280	STE20	STE50	0.0001235
MFALPHA2	STE3	0.9404000	STE12	FUS3	0.0000620
MFALPHA1	MFALPHA2	0.9404000	STE18	STE50	0.0000619
GPA1	FUS3	0.9404000	MATING_TYPE	TEC1	0.0000519
MFA2	MFALPHA1	0.9349580			

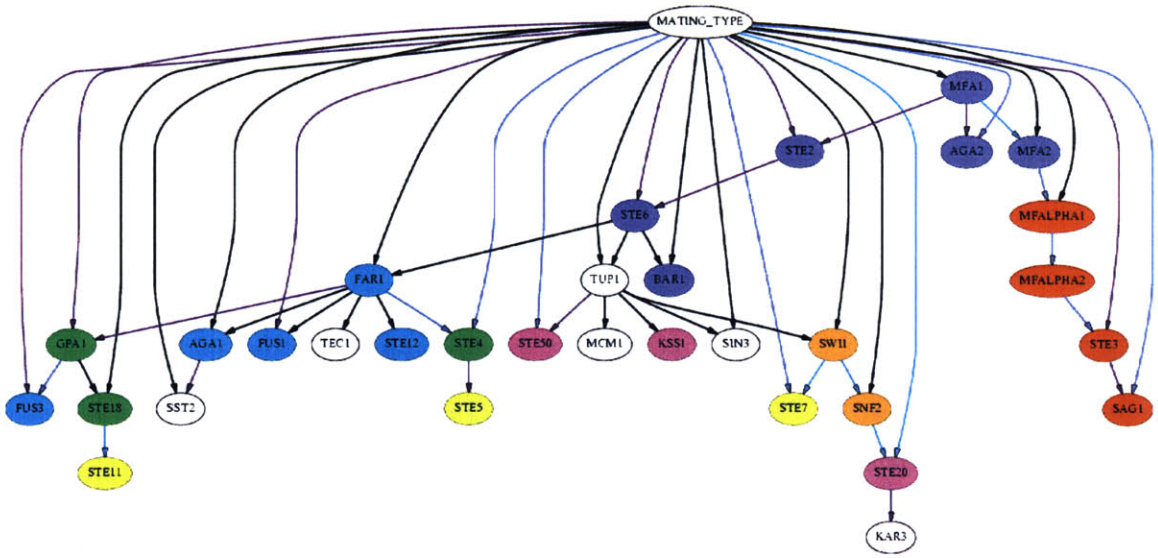

Figure 7-8: *Bayesian network model learned by model averaging over the five hundred highest scoring models visited during the unconstrained simulated annealing search run. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges are colored according to the posterior probability of their inclusion, as estimated by a weighted average over the five hundred highest scoring models. Edges are included in the figure if and only if their posterior probability exceeds 0.5. Node and edge color descriptions are included in the text.*

Table 7.6: *Posterior probabilities of edges being present in the constrained simulated annealing search as estimated by a weighted average over the five hundred highest scoring models. Since the four edges required by location analysis appear in all visited graphs, their posterior probability is 1 by definition.*

From	To	Posterior Prob.	From	To	Posterior Prob.
STE6	STE2	1.0000000	MATING_TYPE	GPA1	0.9979470
MATING_TYPE	STE2	1.0000000	MATING_TYPE	STE7	0.9959610
STE2	MFA1	1.0000000	SST2	FAR1	0.9956230
MATING_TYPE	MFA1	1.0000000	MATING_TYPE	AGA2	0.9915840
GPA1	STE5	1.0000000	MFA2	MFALPHA1	0.9911710
STE6	TUP1	1.0000000	MATING_TYPE	STE50	0.8627090
MATING_TYPE	TUP1	1.0000000	GPA1	STE4	0.7190930
STE12	FUS1	1.0000000	SWI1	STE7	0.6980530
TUP1	SWI1	1.0000000	MFA1	FUS3	0.3288450
MATING_TYPE	SWI1	1.0000000	FAR1	STE4	0.2809070
MATING_TYPE	MFALPHA1	1.0000000	MATING_TYPE	STE4	0.2809060
TUP1	SIN3	1.0000000	MATING_TYPE	KSS1	0.1904580
STE12	AGA1	1.0000000	STE50	STE7	0.1808790
MATING_TYPE	AGA1	1.0000000	AGA2	FUS3	0.1517050
MATING_TYPE	MFA2	1.0000000	STE5	STE7	0.0452417
TUP1	MCM1	1.0000000	STE11	STE7	0.0114721
AGA1	SST2	1.0000000	MATING_TYPE	MFALPHA2	0.0081962
MFALPHA2	STE3	1.0000000	MFA1	MFALPHA1	0.0044375
MATING_TYPE	STE6	1.0000000	MATING_TYPE	FAR1	0.0043766
FAR1	TEC1	1.0000000	STE2	MFALPHA1	0.0024661
GPA1	STE18	1.0000000	MATING_TYPE	STE5	0.0008114
MATING_TYPE	STE18	1.0000000	STE18	STE50	0.0004177
MFALPHA2	SAG1	1.0000000	AGA2	MFALPHA1	0.0003914
STE12	FAR1	1.0000000	MATING_TYPE	KAR3	0.0003614
STE6	BAR1	1.0000000	STE20	STE50	0.0003008
MATING_TYPE	STE12	1.0000000	SWI1	SNF2	0.0002817
TUP1	KSS1	1.0000000	SNF2	STE20	0.0002817
MFA1	AGA2	1.0000000	MATING_TYPE	STE20	0.0001019
STE12	FUS3	1.0000000	STE11	KAR3	0.0000791
MATING_TYPE	FUS3	1.0000000	STE6	MFALPHA1	0.0000523
MATING_TYPE	SNF2	0.9999970	STE4	GPA1	0.0000420
FAR1	STE6	0.9999950	MFA1	MFA2	0.0000371
MATING_TYPE	BAR1	0.9999950	STE7	STE50	0.0000239
MFALPHA1	MFALPHA2	0.9999950	MATING_TYPE	STE11	0.0000221
STE4	FUS1	0.9999860	MATING_TYPE	TEC1	0.0000164
MATING_TYPE	SIN3	0.9999840	STE50	STE11	0.0000148
STE18	STE11	0.9999790	FAR1	FUS1	0.0000138
STE2	MFA2	0.9999630	SNF2	STE7	0.0000085
FAR1	GPA1	0.9999580	MFALPHA2	MFALPHA1	0.0000050
MATING_TYPE	STE3	0.9998550	GPA1	STE6	0.0000050
STE20	SNF2	0.9997180	MFA2	MFALPHA2	0.0000050
TUP1	STE20	0.9997180	STE18	KAR3	0.0000038
STE20	KAR3	0.9995980	STE11	STE50	0.0000024
MATING_TYPE	SAG1	0.9994460	SNF2	STE11	0.0000020
TUP1	STE50	0.9983870	MFALPHA2	STE50	0.0000018
MATING_TYPE	SST2	0.9979600			

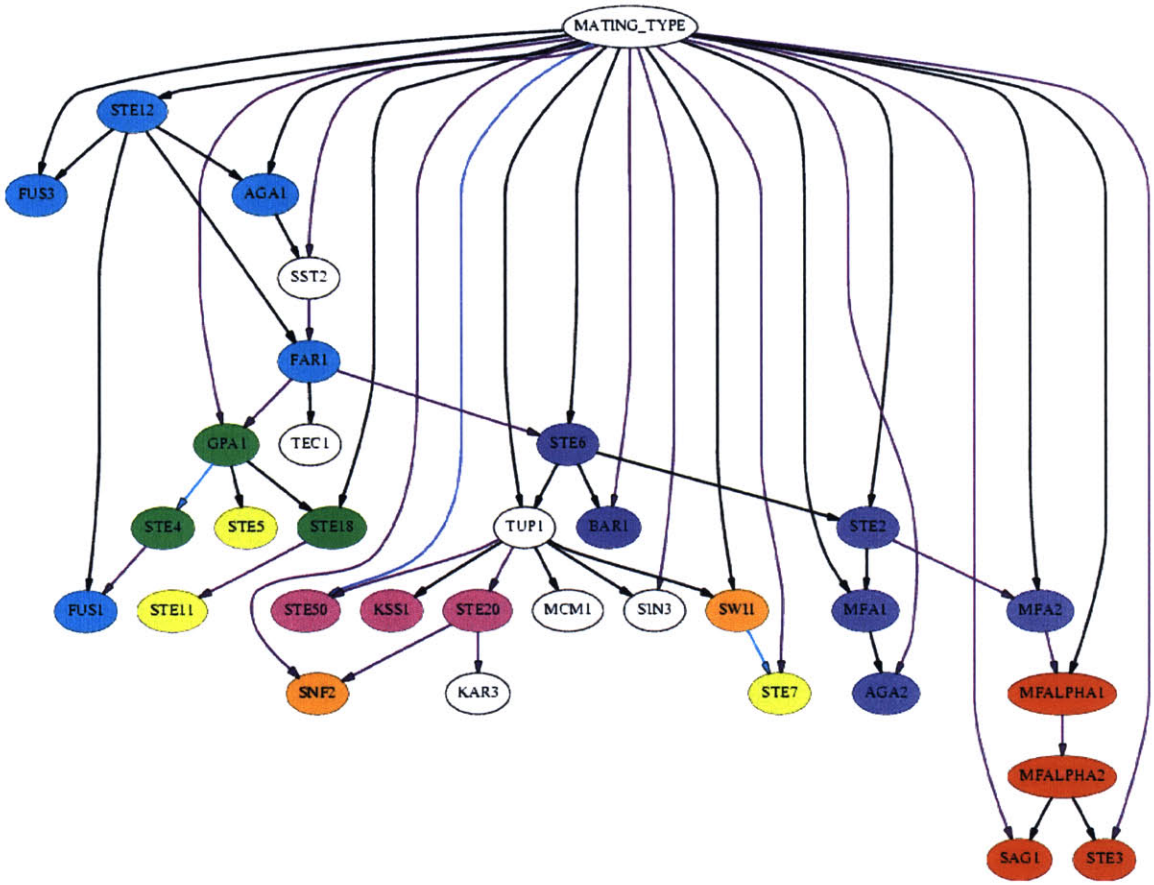

Figure 7-9: *Bayesian network model learned by model averaging over the five hundred highest scoring models visited during the constrained simulated annealing search run. Nodes have been augmented with color information to indicate the different groups of variables with known relationships in the literature. Edges are colored according to the posterior probability of their inclusion, as estimated by a weighted average over the five hundred highest scoring models. Edges are included in the figure if and only if their posterior probability exceeds 0.5. The posterior probability of the four edges required by location analysis is 1 by definition. Node and edge color descriptions are included in the text.*

STE18 are connected in the unconstrained graph. Indeed, even the link between GPA1 and STE4 in the constrained graph is fairly weak. Also, SWI1 and SNF2 (orange) are weakly adjacent in the unconstrained graph, but not adjacent in the constrained graph, though in both cases they are close descendants of TUP1. STE11 and STE5, two of the core elements of the primary signaling cascade complex (yellow), are seen as descendants of G-protein complex genes, indicating statistical dependence that may be the result of common or serial regulatory control. STE7 occurs elsewhere, however. Auxiliary signaling cascade genes (magenta) are always descendants of TUP1, sometimes directly and sometimes more indirectly, but STE50 and KSS1 are siblings in both cases, as before. In general, the auxiliary cascade elements STE20, STE50, and KSS1 do not tend to cluster with the core elements, indicating that the regulation of their transcript levels may occur by a different mechanism than those of the genes in the core signal transduction complex.

As before, TUP1 is the gene with the most children across both networks, consistent with its role as a general repressor of RNA polymerase II transcription. Both networks have MCM1 and SIN3 as children of TUP1; Tup1 and Mcm1 are known to interact in the cell [46] and this result that the level of Tup1 is helpful in predicting the level of Mcm1 suggests a possible regulatory relationship between the two. FAR1 is the gene with the second highest number of children across both networks, consistent with its role in cell cycle arrest. FAR1 is a parent of TEC1 and GPA1 in both networks. Far1, Tec1, and Gpa1 are all known to be cell-cycle regulated and all three are classified as being transcribed during early G_1 phase [26]. This result suggests that Far1 may play a role in regulating the expression of Tec1 and Gpa1, providing a possible mechanism for their previously observed G_1 phase co-expression.

Though it is produced at higher levels in MATa cells, it is known that Aga1 is produced in both MATa and MATα cells: "the AGA1 transcript was expressed and induced by pheromone in both a and alpha cells, suggesting that the a-specific expression of active a-agglutinin results only from a-specific regulation of [AGA2]" [105]. The graphs are each consistent with this knowledge, including a reasonably strong predictive edge from `mating_type` to AGA1, but not clustering AGA1 with other mating type specific genes (dark blue and red) as it is likely regulated differently. In all cases, AGA1 and SST2 are

adjacent, consistent with the fact that the two are expressed very similarly, both peaking at the M/G_1 phase of the cell-cycle [109].

7.5 Discussion

First, we compare the results of model discovery presented in this chapter with those that result from a data-driven methodology commonly employed in the analysis of genomic expression data: hierarchical clustering. Figure 7-10 shows two dendograms that are output by a popular hierarchical clustering program [39]. The dendogram on the left was generated using normalized but non-discretized log reported expression values while the tree on the right was generated using normalized and discretized log reported expression values. Although certain variables cluster similarly to how they cluster in the Bayesian network representation, the output is limited in the sense that it is a linear ordering and is computed on the basis of only pair-wise correlations between variables.

Similarly, we note that while many of the high-level features of the networks learned in the model selection context are consistent with those learned in the model averaging context, the latter approach is more robust to noise and more theoretically sound. It may seem reassuring that the differences between the models discovered in the two contexts are not extreme, but that can be deceptive: most reasonable model induction methods are probably competent at picking out the major features in the data (as even hierarchical clustering does to some extent).

When we compare the results of the discovery process both before and after the inclusion of genomic binding location data for Ste12, the composite network resulting from model averaging based only on genomic expression data has a few apparent limitations. Most strikingly, the search method is unable from expression data alone to learn the correct regulatory relationships between Ste12 and its targets. By fusing expression data with location data, the constrained search is able to consider statistical dependencies in the expression data that are consistent with the relationships already identified in the location data. In this way, location data proves to be quite complementary to expression data: since it can help identify network edges directly, location data dramatically decreases the amount

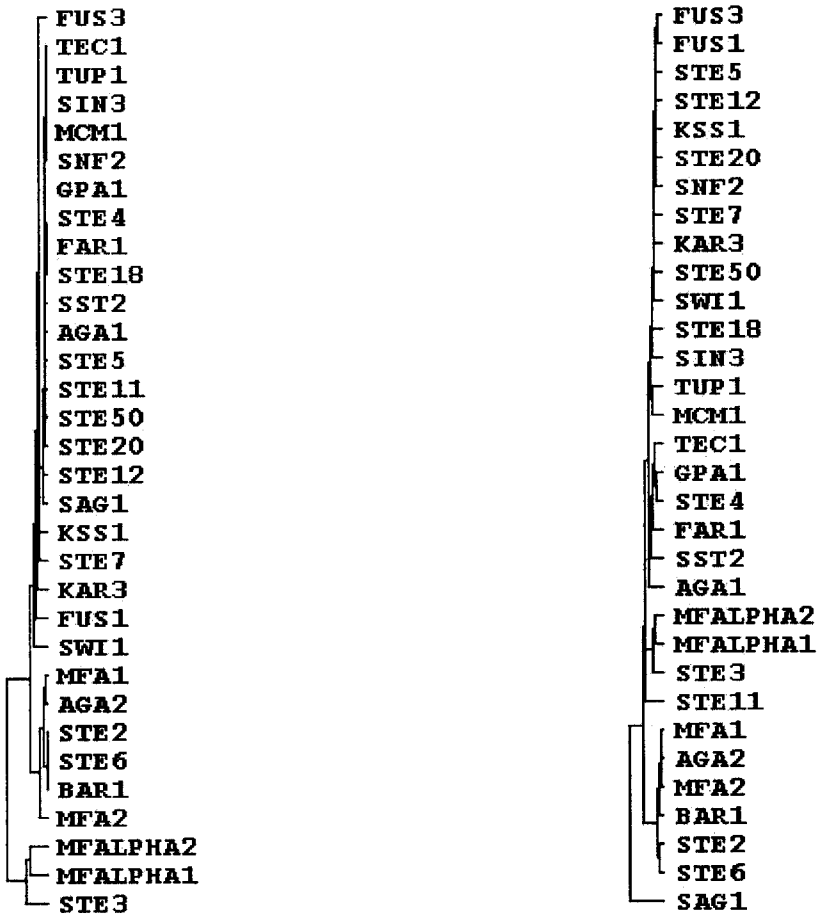

Figure 7-10: *Dendograms showing the output of the traditional hierarchical clustering algorithm commonly used to analyze gene expression. The dendogram on the left was generated using normalized but non-discretized log reported expression values while the dendogram on the right was generated using normalized and discretized log reported expression values. Although certain variables cluster similarly to how they cluster in the Bayesian network representation, the output is limited in the sense that it is a linear ordering and is computed on the basis of only pair-wise correlations between variables.*

of expression data needed to discover regulatory networks.

When we interpret automatically generated Bayesian networks, however, it should be remembered that edges indicate a statistical dependence between the transcript levels of the genes, but do not necessarily specify the form or presence of a physical dependence. For example, in both networks in Figures 7-8 and 7-9, a link appears between MFA2 and MFALPHA1. Although these mating factors are never both expressed in haploid *S. cerevisiae* strains, cells expressing a lot of one are less likely, statistically, to be expressing a lot of the other; hence the link. The fact that the link is weak indicates that other variables such as `mating_type` are frequently successful in explaining away this statistical dependence.

In general, multiple biological mechanisms may map to the same set of statistical dependencies and thus be hard to distinguish on the basis of statistical tests alone. Moreover, if sufficient data does not exist to observe a system in a number of different configurations, we may not be able to uncover certain dependencies. These two limitations are not specific to this methodology, however, but rather are true of statistical methods in general.

There remain a number of ways to extend this work in the future. Among these are the use of heuristic search algorithms that more frequently visit high scoring regions of the model search space, incorporation of data from other sources besides expression and location data, and adding the ability to discover annotated network edges refining the type of relationship learned between model variables as mentioned in [56] and in Chapter 5 of this dissertation. Also, although the modeling framework permits it in theory, software for dealing with time series and interventional knockout data explicitly still needs to be implemented.

Chapter 8

Conclusion

As this dissertation comes to a close, it is important to address a number of residual issues. Section 8.1 offers a discussion of a number of the subtle issues associated with computational modeling of biological systems in general, and with using Bayesian networks to model genetic regulatory networks in particular. Building on this discussion, Section 8.2 suggests some concrete directions for future research to extend the work presented here. We finally conclude the dissertation in Section 8.3 by examining the contributions of this work in light of the current research in this field.

8.1 Discussion of subtleties

8.1.1 Proper interpretation of Bayesian network structure

Certain limitations exist when using Bayesian networks for modeling genetic regulatory networks. The most important of these is the caution with which models must be interpreted. While graphs are highly interpretable structures for representing statistical dependencies, they have the potential to be misleading if interpreted incorrectly. It is important to distinguish between statistical interaction and (physical) biological interaction.

For example, if the data strongly support the inclusion of an edge between two variables X and Y, that may indicate a physical interaction between these two factors in the cell. Alternatively, it is possible that an unmodeled variable Z actually intermediates between X and Y, such that X and Y exhibit statistical dependence but no physical interaction.

Caution must be used when interpreting models that may be missing critical explanatory variables. In general, multiple biological mechanisms may map to the same set of statistical dependencies and thus be hard to distinguish on the basis of statistical tests alone. We note that although multiple biological mechanisms may map to the same set of dependence statements, that frequently the opposite is true so the method is typically likely to be useful. Moreover, interventional data can help resolve certain ambiguities that can arise in the context of purely observational data.

In contrast, if the data strongly supports the exclusion of an edge between two variables X and Y, that may indicate the absence of physical interaction between these two factors in the cell. Alternatively, we may not have observed the cell under an appropriate set of conditions where this interaction could have been observed. If sufficient data does not exist to observe a system in a number of different configurations, we may not be able to uncover certain dependencies. As an example, consider an analogy from logical relationships between three variables. Imagine that in actuality, X and Y are independent variables but Z is the `AND` of X and Y. However, assume that because of the particular set of experimental conditions under which these variables are observed, that X is always observed to be 1 while Y fluctuates between 0 and 1. Under these circumstances, Z fluctuates between 0 and 1 correspondingly. In looking for statistical patterns among the variables in the data set, it will likely be observed that Y is a necessary predictor of Z but that X is irrelevant, which is incorrect. As this example shows, models based on data sets that do not exercise the entire set of parent configurations can satisfactorily explain statistical dependencies and at the same time fail to identify edges in the graph that are necessary in explaining actual biological mechanisms.

These two limitations are not specific to this methodology, however, but rather are true for statistical methods in general. Statistical dependence, statistical causality, and mechanistic causality are distinct notions. For example, the observation that a cell expresses a-factor and the observation that a cell expresses α-factor are statistically related though not mechanistically. Of course, a mechanism exists to explain the statistical dependence but it operates at a level much lower than the level of gene expression since it is an observation that wild-type haploid *S. cerevisiae* cells are either of one mating type or the other and thus

have either genes for the production of a-factor or genes for the production of α-factor but not both. Readers are recommended to read the work of Pearl (for example [93, 94]) for more detailed discussion of the philosophical and mathematical implications of this distinction.

In our case, this distinction is important since it demands we remember that biological mechanism and computational mechanism are not the same thing. The two dwell in different domains and although each informs the other, they are not identical. Explanation of a biological system at a computational level is distinct from explanation of the system at a biological level. As a consequence, the two explanations of a given system may differ even if both are perfectly well understood. For example, we may employ computational metaphors like bandpass filter, amplifier, or switch to explain a biological system but these high-level descriptions of the computational behavior of the system may not map directly to well-defined biological mechanisms. At the most direct level of interpretation, the methods we develop are useful for describing the computational and statistical mechanisms rather than the biological mechanisms giving rise to the observed data. It is our hope that these computational and statistical mechanisms help to reveal the biological mechanisms ultimately responsible for orchestrating living cells, but this is merely our hope. We have no reason to doubt that these methods are useful for this task, but it is important to state clearly the appropriate understanding of their application; the models act as a guide to our understanding rather than as absolute truth in and of themselves.

8.1.2 Incorporating additional data sources

A number of other sources of data beyond genomic expression and location data can be exploited in the model validation and discovery processes. We have already seen in Chapter 7 how genomic location data can be used to complement genomic expression data in directly identifying regulatory targets of DNA-binding factors like Ste12. It complements expression data in the sense that location data offers very direct facts describing biological mechanism while expression data operates at the computational or statistical level to explain dependencies that are the sum of direct and indirect factors. Location data greatly amplifies the amount of available data because of its ability to identify necessary graph edges directly. The promise of principled information fusion is that various different data sources can each

contribute to the process of elucidating genetic regulatory networks, each source filling in where others leave gaps. What additional data sources are available?

Textual data

A tremendous amount of information regarding biological systems is represented in textual form. Principally, the results of biological experiments over the course of the past fifty to one hundred years are contained in a body of literature comprised of millions of texts and published papers. Even if we only consider the abstracts of such papers, a tremendous wealth of information can be exploited. The primary problem, of course, is that this information needs to be transduced into representations other than natural language text. While the text often exists in electronic form (or could be scanned into electronic form automatically), the information still needs to be transduced from natural language text to some representation with a readily interpretable semantics in the context of automatic learning of biological dependencies from uncertain data. Many people are beginning to work on this problem, employing a variety of different methods for extracting information from textual sources like paper abstracts [63, 92]. One of the especially interesting questions is not how to transduce the information but rather to what computationally interpretable representation it should be transduced in order to best use the information downstream.

Untapped genomic expression data

Currently, genomic expression data exists in fragmented collections in isolated laboratories around the globe. However, it is clear that this data will be of even greater value if it can be collected into large publicly available databases much as PDB or GenBank have done for protein structures and gene sequences, respectively. As mentioned in Chapter 2, efforts are currently under way to establish data interchange formats and public submission databases to collect and curate the vast quantity of gene expression data being gathered today. With pressure from publishing outlets (*e.g.*, requiring data to be submitted to public databases before publication is permitted), it is eminently foreseeable that most genomic expression data being gathered will eventually make its way into public expression databases, becoming highly available for others to access and leverage. However, the reason for the

vast success of databases like PDB and GenBank in the biological community is that they contained data that was inherently comparable. As discussed in the introductory chapter, a number of different technologies for measuring gene expression have been developed, including lithographically deposited oligonucleotide arrays such as those of Affymetrix, printed cDNA arrays such as those of the MicroArrayer, Incyte, and Corning, and other methods like SAGE (serial analysis of gene expression) and RT-PCR. Storing the raw data from these technologies in a single database is of little use as the raw data are hardly comparable. Before such a venture can meet with success, a result format that is common to all platforms needs to be standardized and deployed, along with methods for converting from the raw results of given expression technology platforms to this common result format. We discuss in Section 8.2.1 some of our initial efforts at developing a cross-platform result format for Affymetrix and printed cDNA array data. Once data become interchangeable and comparable, public databases of gene expression will become a tremendous resource to the community and will be able to be exploited for the automatic elucidation of biological knowledge from data.

Proteomic expression data

While gene expression can now be measured reliably in high throughput ways (whole genomes at a time), protein expression measurement technologies are still in their infancy. Nevertheless, a number of methods for measuring protein expression (*e.g.*, 2-D gel assays) are being developed and this continues to be an area where large amounts of research funds are being expended because the expected payoff is so significant. Consequently, it seems likely that protein expression data will be gathered in increasingly large quantities to accompany the genomic expression data already being gathered. Putting these two sets of data together should not prove to be very difficult as both are measurements of the abundance of particular variables in the cell at an instant of time. Other sources of data will prove to be more of a challenge to integrate.

Other sources of data

In addition to genomic and proteomic expression data, a wide range of other experimental data sources could be leveraged for elucidating genetic regulatory networks if such data can be suitably represented and fused with existing data. We have already discussed *in vivo* DNA-protein interaction data as determined by binding location analysis. Other data sources to possibly exploit include protein localization, protein modification, protein structure, *in vitro* and *in vivo* protein-protein interaction, small metabolite concentrations, cell morphology, systematic gene knockout phenotypes, sequence analysis, and cross-organismal evolutionary sequence comparison. We present later in Section 8.2.2 some thoughts on how all these data sources can be fused together to assist in the network elucidation process.

8.1.3 Incremental prior specification

The Bayesian approach to learning has the nice property that there are principled ways of combining prior knowledge about a domain with the information about the domain that is captured in the form of observed data. In a Bayesian framework, the prior is meant to capture all the information that is known about the domain before observations of the domain have been made. In the limit of no prior information about a domain, it is possible to use an uninformative prior, as we sometimes have in this dissertation. However, in general, much is known about a domain, either from intuitions about the domain that arise from previous experience with the domain (apart from the observations that led to the available data regarding the domain) or from various constraints that must hold in the domain (gravity not being violated, entropy not decreasing, *etc.*).

The typical formulation of this problem in a Bayesian setting is for the user to first represent in a prior the totality of his information about the domain, and then for this prior to be modified, upon observation of the data, into a posterior that combines in a principled way the prior beliefs of the user with the information gathered from the data. The challenge in such a setting, however, is in requiring the user to somehow express all his prior information about the domain in an appropriate way so that it can be combined with the observed data. A great deal of effort has gone into characterizing straightforward ways of capturing prior information effectively and accurately, but the problem is still a difficult

one.

Because the prior by its very nomenclature implies that it represents information that is known to the user before observation of the data, it is typically imagined that the prior must be specified before the Bayesian framework can proceed with learning from the observed data. This, however, is not required. Since it is possible to represent the user's prior information in the form of equivalent simulated data [59, 72], from a mathematical perspective the learning process can consider the observed data and the prior-equivalent data interchangeably. In particular, we can imagine learning processes that alternate between consideration of the observational data and consideration of the prior information of the user.

Of course, a significant *caveat* must be mentioned. The biggest advantage of gathering all prior information before learning from data is ensuring that the observed data do not influence the assessment of the prior. That is, one might imagine in a framework in which the user's prior information is considered alternately with the observed data that the prior information of the user might actually become tainted by his observation of the data. In such a context, this "prior" is not at all prior but some combination of prior and data, effectively overestimating the relative impact of the data on the posterior. But if the user is careful to eliminate, or at least minimize, this effect, then the framework of alternating consideration offers a number of possible advantages in the learning process.

First, the user need not specify everything that is already known before the learning process, a task which is recognized to be notoriously difficult [59, 57]. Instead, the user can specify prior information that is especially relevant to the learning process as the need arises. For example, imagine that the learning process reports on the basis of an uninformative prior that two models are equally effective at explaining the observed data in that both are given equally high scores after the data have been considered. However, one such model may be in total opposition to what is already known about the domain while the other is more consistent. In such a setting, the user can specify the relative prior strength he would ascribe to one model with respect to the other. Thus, he provides prior information helping to distinguish between models where it is likely to be relevant but does not need to distinguish between other models which so poorly explain observed data that his prior comparison of

their relative strengths is unlikely to change the outcome. Again, it is important that this "selective prior information" not become too biased. In particular, it is possible that if the user is only required to provide priors with respect to models that score well after learning from observed data, that he or she may bias consideration in favor of those models. It is important to employ methods that do not ascribe inappropriate prior information to models about which the user is not queried for prior information.

Second, we can now begin to imagine a learning framework in which the user and the computational tools cooperate in the learning process. Rather than develop a framework in which computers learn from large piles of data in isolation, either without any prior information or after having extracted this information from the user and never again consulting him, we can now imagine scenarios in which the user and the computer together engage in the learning process. In the computational community, a natural temptation (or challenge) is to develop computational frameworks that are as good or better than humans in accomplishing certain tasks, including learning. This is the entire basis of the Turing test, the measure by which we are able to determine if a computer is truly intelligent. We attempt to develop robots that think as well as people, algorithms that process data and learn as well as people, voice recognition and speech synthesis systems that converse as well as people, even computers like Deep Blue that play chess as well as people. But what about systems that, rather than match or exceed human abilities, instead attempt to augment or complement human abilities? What if Deep Blue was not designed to play against Gary Kasparov, but rather to help Gary Kasparov play? Imagine combining the deep, intuitive, structural knowledge of chess positions and strategies that Kasparov has with the exhaustive computational move-processing power of Deep Blue — that would be a formidable chess opponent.

Rather than developing a single monolithic algorithm that unearths major biological insights automatically from large mounds of data, we can consider algorithms that work to develop these insights by combining the user's deep intuition about the operation of biological systems with the computational learning that is possible with vast quantities of data. Thus, an important goal is to develop algorithms and tools that are capable of augmenting the user's intuition as well as extracting prior information incrementally and

without bias. Bayesian frameworks are ideally suited to combining the prior information of users with the information embedded in repeated observation of the system in question, but we will likely need to consider tools that do not gather all the prior information in advance and then discard the user, but rather tools that extract helpful information from the user as it is needed in the learning process. This process will likely be interactive and online, not limited simply to batch learning from data. Ideally, these tools will be able to suggest new experiments to be conducted and the interactive process will proceed as new data continue to be generated.

8.1.4 Incremental data collection

Not only can model discovery tools evolve to incrementally query the user for prior information but also to incrementally collect data. The necessity of observing a system in a number of configurations in order to best elucidate its structure, as emphasized earlier in this chapter and elsewhere in this dissertation, suggests the possibility of performing *experimental suggestion* in the future. In such a context, existing models and data could be used to generate suggestions for new experiments, yielding data that would optimally elucidate a given regulatory network.

As discussed above, elucidation of genetic regulatory networks must not simply be a batch learning process. The space of possible models to consider is so large that we cannot even begin to imagine gathering sufficient data to allow an algorithm to simply churn away and produce a correct model without any intervention. Rather, we need to consider learning that is incremental and learning algorithms that are online.

In particular, rather than gathering data sampled from the joint probability space over all relevant variables in cellular genetic regulatory networks, it is important to carefully design experiments to learn information about the specific portions of these networks that remain ambiguous. Being able to suggest the next series of experiments to conduct is especially valuable in this context of learning from genomic expression data because the data is costly to gather, in terms of both laboratory time and money. It is quite useful to know in advance which are likely to be the most informative experiments to conduct for elucidating biological mechanisms of interest.

8.2 Extensions for future work

8.2.1 Cross-platform comparability of genomic expression data

Although Chapter 2 presented a methodology for making data from different Affymetrix arrays comparable, it is limited to Affymetrix arrays. However, as discussed above, a large amount of genomic expression data has been gathered using printed cDNA arrays. Because of inherent spot-size variability, data from these arrays is typically not even comparable with itself unless a common control sample has been applied to multiple arrays.[1] In such a scenario, the expression ratios across arrays become comparable, within the limits of the noise associated with the reproducibility of samples across different arrays. However, not only is it the case that these data are not *per se* comparable with data gathered from Affymetrix arrays, but also these data are not comparable with data gathered from another collection of printed cDNA arrays if different control samples have been applied to the two collections of printed arrays.

One possible approach for making printed cDNA array data comparable with itself and with data gathered on Affymetrix arrays is to characterize on Affymetrix arrays the common samples used as controls on printed arrays. For instance, if fifty observations of a variety of cancer tissues are made on printed arrays, but each is measured as a series of expression ratios with respect to a single common control, then if we knew the absolute level of expression for the genes in this single common control, we could use that information to estimate absolute expression levels for all fifty samples.

We have done some preliminary work along these lines but do not present results here for lack of space and time. We mention that this approach may in general prove to be more difficult to implement than it is worth. As printed cDNA arrays continue to be manufactured under higher standards of spot-size variability, the problem may soon resolve itself, previously collected data notwithstanding.

[1] This is possible in the microarray context because of the presence of two different (competitively hybridized) samples on each array.

8.2.2 Increased information fusion

In this dissertation, we have demonstrated how genomic expression and location data can be combined to elucidate genetic regulatory networks, but as we have seen above, many other sources of data can be exploited for this task as well. In terms of information fusion, it is important to distinguish between two classes of additional data. The first consists of data that can be measured simultaneously with gene expression. Examples include protein expression, *in vivo* protein modification, levels of metabolites or other small molecules, or even cell morphology. As long as these are observed in tandem with the levels of gene expression, they can be modeled simply by adding additional variables to the graph.

The second class consists of data that cannot be gathered at the same time as the levels of gene expression are measured. Examples include learning from a two-hybrid screen that two proteins interact, learning from location analysis that a transcription factor binds to the upstream sequences of certain genes, or learning from sequence analysis that two genes share a common promoter motif. These assays produce results that are not measured in terms of factor abundance but rather in terms of factor interaction, factor presence, and factor spatial localization. Some of these results are qualitative rather than quantitative (factors do or do not interact, *e.g.*). In theory, the Bayesian methodology provides principled ways for incorporating this additional information as it has a natural provision for incorporating prior information into its scoring metric; in practice, giving appropriate weight to each of these sources of information poses a significant challenge.

8.2.3 Experimental suggestion

Experimental suggestion is a natural area to consider as a possible extension of this work. As mentioned in the introductory chapter, the model-driven analysis paradigm allows for experimental suggestion quite naturally. This field is generally known as active learning and a sizable literature exists that can be applied and extended in this domain (see, *e.g.*, Tong and Koller [116, 117]). Of special interest is the ability to suggest experiments for collecting not only observational data but also interventional data. In the context of genetic regulatory networks, this can be implemented by deleting a gene so that it cannot be expressed or by constitutively over-expressing a gene. Interventional data needs to be treated differently

from observational data in the context of learning as mentioned in Chapter 4, but the framework easily extends to handle interventional data [29, 94, 96].

8.2.4 Other extensions and improvements

Other directions for future work that were mentioned earlier in this dissertation include using graphical models such as dynamic Bayesian networks to model the simple dynamics of genetic regulatory networks, using variational methods to produce efficient tools for scoring regulatory networks with latent variables, increasing the variety of edge annotations permitted in models, and fast algorithms to search for appropriate annotations during the model induction process. Additional software advances include extension of the model validation and discovery tools to handle stochastically discretized data as discussed in Chapter 3 and interventional data as discussed in Chapter 4. Improved normalization and discretization methods should also be examined, and in this context, the propagation of uncertainties throughout the analysis pipeline would be quite useful.

8.3 Contributions

In this dissertation, we have made a number of contributions to the existing literature on the normalization, discretization, and analysis of genomic expression data, on the application and annotation of graph models and Bayesian networks, and hopefully eventually on the understanding of the pheromone response regulatory network in *Saccharomyces cerevisiae*.

We propose and advocate the use of a model-driven paradigm for the analysis of genomic expression data, in contrast to the data-driven analysis paradigm which predominates today. Concurrently with but independent of similar work by Murphy and Mian [91] and Friedman, *et al.* [45], we suggest the use of graphical models, and Bayesian networks in particular, for modeling biological systems in the presence of large amounts of data. Moreover, we extend standard Bayesian network semantics to include annotated edges for modeling biologically meaningful network interactions and we develop and implement a method for scoring these annotated network models.

We apply this modeling framework in both model validation and model discovery contexts, using posterior model averaging in the latter context to guard against over-fitting of model structure. We also apply the framework to fuse both genomic expression and genomic location data, enabling the induction of models not readily discovered if the data sources are considered in isolation. The regulatory network models of the pheromone response system we discover in Chapter 7 are generally consistent with existing understandings of the regulatory network but also offer a number of novel hypotheses for explaining observed data that can be investigated experimentally in the future.

In order to demonstrate the end-to-end application of this modeling framework, we develop suitable normalization and discretization methods for this particular domain. In the case of normalization of genomic expression data, we derive a novel method for combining multiple sources of exogenous normalization information into a single estimate for the optimal scaling factor associated with each Affymetrix array. We demonstrate experimentally that this method is better able to normalize arrays across a wide variety of experimental conditions than other methods proposed to date.

Finally, we hope that this dissertation offers a contribution in its careful treatment of the various issues and subtleties associated with the computational elucidation of genetic regulatory networks and the directions for future work presented in this chapter.

Bibliography

[1] Affymetrix GeneChip expression analysis manual. Appendix 5: GeneChip 3.1 Expression Analysis Algorithm Tutorial.

[2] Affymetrix GeneChip spotted array technology. http://www.affymetrix.com/technology/tech_spotted.html.

[3] Tatsuya Akutsu, Satoru Miyano, and Satoru Kuhara. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In *Pacific Symposium on Biocomputing*, volume 4, pages 17–28, 1999.

[4] Ash A. Alizadeh, Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore Lossos, Andreas Rosenwald, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[5] Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences, USA*, 97(18):10101–10106, August 2000.

[6] David Applegate and Ravi Kannan. Sampling and integration of log-concave functions. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 156–163, 1991.

[7] Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics*, 149:1633–1648, 1998.

[8] ArrayExpress. http://www.ebi.ac.uk/arrayexpress/.

[9] Hagai Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 21–30. Morgan Kaufmann Publishers, 1999.

[10] David Auerbach. Supercooling and the Mpemba effect. *American Journal of Physics*, 63(10):882–885, October 1995.

[11] Ziv Bar-Joseph, David K. Gifford, and Tommi S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. In *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology (ISMB 2001)*. ISCB, July 2001.

[12] Tracy Bergemann, Filemon Quiaoit, Jeffrey J. Delrow, and Lue Ping Zhao. Statistical issues in signal extraction from microarrays. In *International Symposium on Biomedical Optics (BiOS 2001)*, pages 24–34. SPIE, January 2001.

[13] Xavier Boyen, Nir Friedman, and Daphne Koller. Discovering the hidden structure of complex dynamic systems. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 91–100. Morgan Kaufmann Publishers, 1999.

[14] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 33–42. Morgan Kaufmann Publishers, 1998.

[15] Norbert Braendle, Horst Bischof, and Hilmar Lapp. Generic and robust approach for the analysis of spot array images. In *International Symposium on Biomedical Optics (BiOS 2001)*, pages 1–12. SPIE, January 2001.

[16] Wray Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers, 1991.

[17] Christopher B. Burge and Samuel Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.

[18] Christopher B. Burge and Samuel Karlin. Finding the genes in genomic DNA. *Current Opinion in Structural Biology*, 8:346–354, 1998.

[19] Atul J. Butte and Isaac S. Kohane. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. In *Pacific Symposium on Biocomputing*, volume 5, pages 415–426, 2000.

[20] Atul J. Butte, Pablo Tamayo, Donna Slonim, Todd R. Golub, and Isaac S. Kohane. Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences, USA*, 97(22):12182–12186, October 2000.

[21] T. Chen, V. Filkov, and S. S. Skiena. Identifying gene regulatory networks from experimental data. In *3rd Annual International Conference on Computational Molecular Biology (RECOMB'99)*. ACM-SIGACT, April 1999.

[22] David Maxwell Chickering. Learning equivalence classes of Bayesian network structures. In P. Besnard and S. Hanks, editors, *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 150–157. Morgan Kaufmann Publishers, 1995.

[23] David Maxwell Chickering. A transformational characterization of equivalent Bayesian network structures. In P. Besnard and S. Hanks, editors, *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 87–98. Morgan Kaufmann Publishers, 1995.

[24] David Maxwell Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: AI and Statistics V*, chapter 12, pages 121–130. Springer-Verlag, 1996.

[25] David Maxwell Chickering, Dan Geiger, and David Heckerman. Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Research, November 1994.

[26] Raymond J. Cho, Michael J. Campbell, Elizabeth A. Winzeler, Lars Steinmetz, Andrew Conway, Lisa Wodicka, Tyra G. Wolfsberg, Andrei E. Gabrielian, David Landsman, David J. Lockhart, and Ronald W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, July 1998.

[27] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

[28] Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[29] Gregory F. Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 116–125. Morgan Kaufmann Publishers, 1999.

[30] Corning to enter fast-growing market for DNA microarrays used in genomic research. http://www.corning.com/CMT/WhatsNew/BusinessRelease.asp, September 2000.

[31] M.C. Costanzo, M.E. Crawford, J.E. Hirschman, J.E. Kranz, P. Olsen, L.S. Robertson, M.S. Skrzypek, B.R. Braun, K.L. Hopkins, P. Kondu, C. Lengieza, J.E. Lew-Smith, M. Tillberg, and J.I. Garrels. YPD, PombePD, and WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information. *Nucleic Acids Research*, 29(1):75–79, 2001.

[32] T. Dean and K. Kanasawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[33] Joseph L. DeRisi, Vishwanath R. Iyer, and Patrick O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 728:680–686, 1997.

[34] Joseph L. DeRisi, Vishwanath R. Iyer, and Patrick O. Brown. *The MGuide, a complete guide to building your own microarrayer, Version 1.1*, 1998.

[35] Patrick D'haeseleer, Xiling Wen, Stefanie Fuhrman, and Roland Somogyi. Mining the gene expression matrix: Inferring gene relationships from large scale gene expression

data. In R. C. Paton and M. Holcombe, editors, *Information Processing in Cells and Tissues*, pages 203–212. Plenum Publishing, 1998.

[36] Patrick D'haeseleer, Xiling Wen, Stefanie Fuhrman, and Roland Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. In *Pacific Symposium on Biocomputing*, volume 4, pages 41–52, 1999.

[37] Ron O. Dror, Jonathan G. Murnick, Nicola J. Rinaldi, Voichita D. Marinescu, Ryan M. Rifkin, and Richard A. Young. A Bayesian approach to transcript estimation from gene array data: The BEAM technique. *submitted*, 2001.

[38] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed. Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. Technical Report 578, UC Berkeley Department of Statistics, August 2000.

[39] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences, USA*, 95:14863–14868, December 1998.

[40] Elaine A. Elion. Pheromone response, mating and cell biology. *Current Opinion in Microbiology*, 3(6):573–581, December 2000.

[41] Stephen P. A. Fodor, J. Read, M. Pirrung, L. Stryer, A. Lu, and D. Solas. Light-directed spatially addressable parallel chemical synthesis. *Science*, 251:767–773, 1991.

[42] J. E. Forman, I. D. Walton, D. Stern, R. P. Rava, and M. O Trulson. Thermodynamics of duplex formation and mismatch discrimination on photolithographically synthesized oligonucleotide arrays. *ACS Symposium Series*, 682:206–228, 1998.

[43] Nir Friedman and Moises Goldszmidt. Discretization of continuous attributes while learning Bayesian networks. In *Proceedings of 13th International Conference on Machine Learning*, pages 157–165, 1996.

[44] Nir Friedman and Daphne Koller. Being Bayesian about network structure. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of the Sixteenth Annual Con-*

ference on Uncertainty in Artificial Intelligence, pages 201–210. Morgan Kaufmann Publishers, 2000.

[45] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. In *4th Annual International Conference on Computational Molecular Biology (RECOMB 2000)*. ACM-SIGACT, April 2000.

[46] I. M. Gavin, M. P. Kladde, and R. T. Simpson. Tup1p represses Mcm1p transcriptional activation and chromatin remodeling of an a-cell-specific gene. *Embo Journal*, 19:5875–5883, 2000.

[47] Gene expression markup language, GEML. http://www.geml.org/.

[48] Gene expression omnibus, GEO. http://www.ncbi.nlm.nih.gov/geo/.

[49] Genetic analysis technology consortium, GATC. http://www.gatconsortium.org/.

[50] GeneX. http://www.ncgr.org/research/genex/.

[51] GeneXML. http://www.ncgr.org/research/genex/genexml.html.

[52] Leon Glass and Stuart A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39:103–129, 1973.

[53] Todd R. Golub, Donna K. Slonim, Pablo Tamayo, C. Huard, M. Gaasenbeek, Jill P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and Eric S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

[54] P. J. E. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proceedings of the National Academy of Sciences, USA*, 95(12):6750–6755, June 1998.

[55] Alexander J. Hartemink, David K. Gifford, Tommi S. Jaakkola, and Richard A. Young. Maximum likelihood estimation of optimal scaling factors for expression array normalization. In *International Symposium on Biomedical Optics (BiOS 2001)*, pages 132–140. SPIE, January 2001.

[56] Alexander J. Hartemink, David K. Gifford, Tommi S. Jaakkola, and Richard A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pacific Symposium on Biocomputing*, volume 6, pages 422–433, 2001.

[57] David Heckerman. A tutorial on learning with Bayesian networks. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. Kluwer Academic Publishers, 1998.

[58] David Heckerman and David Maxwell Chickering. A comparison of scientific and engineering criteria for Bayesian model selection. Technical Report MSR-TR-96-12, Microsoft Research, 1996.

[59] David Heckerman, Dan Geiger, and David Maxwell Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[60] P. E. Hodges, A. H. X. McKee, B. P. Davis, W. E. Payne, and J. I. Garrels. The yeast proteome database (ypd): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Research*, 27(1):69–73, 1999.

[61] Frank C. P. Holstege, Ezra G. Jennings, John J. Wyrick, Tong Ihn Lee, Cristoph J. Hengartner, Michael R. Green, Todd R. Golub, Eric S. Lander, and Richard A. Young. Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, 95:717–728, November 1998.

[62] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[63] Ioannis Iliopoulos, Anton J. Enright, and Christos A. Ouzounis. TEXTQUEST: Document clustering of MEDLINE abstracts for concept discovery in molecular biology. In *Pacific Symposium on Biocomputing*, volume 6, pages 384–395, 2001.

[64] IncyteGenomics. http://www.incyte.com/.

[65] Vishwanath R. Iyer, Michael B. Eisen, Douglas T. Ross, Greg Schuler, Troy Moore, Jeffrey C. F. Lee, Jeffrey M. Trent, Louis M. Staudt, James Hudson Jr., Mark S. Boguski, Deval Lashkari, Dari Shalon, David Botstein, and Patrick O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, January 1999.

[66] Tommi Jaakkola and Michael Jordan. Variational probabilistic inference and the QMR-DT database. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[67] F. V. Jensen. *Introduction to Bayesian Networks.* Springer-Verlag, 1996.

[68] M. Johnston. A model fungal gene regulatory mechanism: the GAL genes of Saccharomyces cerevisiae. *Microbiological Reviews*, 51(4):458–476, 1987.

[69] Laura M. Kegelmeyer, Lisa Tomascik-Cheeseman, Melinda S. Burnett, Paul van Hummelen, and Andrew J. Wyrobek. Groundtruth approach to accurate quantitation of fluorescence microarrays. In *International Symposium on Biomedical Optics (BiOS 2001)*, pages 35–45. SPIE, January 2001.

[70] M. Kathleen Kerr, Mitchell Martin, and Gary A. Churchill. Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, 7, 2000.

[71] Uffe Kjærulff. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Annual Conference on Uncertainty in Artificial Intelligence*, pages 121–129. Morgan Kaufmann Publishers, 1992.

[72] Petri Kontkanen, Petri Myllymäki, Tomi Silander, Henry Tirri, and Peter Grünwald. A comparison of non-informative priors for Bayesian networks. Technical Report NC-TR-98-002, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1998.

[73] Thomas S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, Chicago, IL, 1970.

[74] Eric S. Lander et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, February 2001.

[75] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:154–227, 1988.

[76] J. Leclercq and J. E. Dumont. Boolean analysis of cell regulation networks. *Journal of Theoretical Biology*, 104:507–534, 1983.

[77] Cheng Li and Wing Hung Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences, USA*, 98(1):31–36, January 2001.

[78] Shoudan Liang, Stefanie Fuhrman, and Roland Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[79] Chao-Lin Liu and Michael P. Wellman. Incremental tradeoff resolution in qualitative probabilistic networks. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 338–345, July 1998.

[80] Chao-Lin Liu and Michael P. Wellman. Using qualitative relationships for bounding probability distributions. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 346–353, July 1998.

[81] David J. Lockhart, Helin Dong, Michael C. Byrne, Maximillian T. Follettie, Michael V. Gallo, Mark S. Chee, Michael Mittmann, Chunwei Wang, Michiko Kobayashi, Heidi Horton, and Eugene L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, December 1996.

[82] 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, 1995.

[83] Hiten D. Madhani and Gerald R. Fink. Combinatorial control required for the specificity of yeast MAPK signaling. *Science*, 275:1314–1317, February 1997.

[84] David Malakoff. Bayes offers a 'new' way to make sense of numbers. *Science*, 286(5444):1460–1464, November 1999.

[85] Harley H. McAdams and Adam Arkin. Simulation of prokaryotic genetic circuits. *Annu. Rev. Biophys. Biomol. Struct.*, 27:199–224, 1998.

[86] Glenn McGall. Personal communication, 1999.

[87] Glenn McGall, Jeff Labadie, Phil Brock, Greg Wallraff, Tiffany Nguyen, and William Hinsberg. Light-directed synthesis of high-density oligonucleotide arrays using semiconductor photoresists. *Proceedings of the National Academy of Sciences, USA*, 93:13555–13560, 1996.

[88] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[89] Stefano Monti and Gregory F. Cooper. A multivariate discretization method for learning Bayesian networks from mixed data. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 404–413. Morgan Kaufmann Publishers, 1998.

[90] T. Murata. Petri nets: properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[91] Kevin Murphy and Saira Mian. Modelling gene expression data using dynamic Bayesian networks. University of California at Berkeley, 1999.

[92] Jong C. Park, Hyun Sook Kim, and Jung Jae Kim. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. In *Pacific Symposium on Biocomputing*, volume 6, pages 396–407, 2001.

[93] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, Los Altos, California, 1988.

[94] Judea Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, Cambridge, England, 2001.

[95] A. C. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. A. Fodor. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Sciences, USA*, 91:5022–5026, 1994.

[96] Dana Pe'er, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. In *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology (ISMB 2001).* ISCB, July 2001.

[97] Gang Peng and James E. Hopper. Evidence for Gal3p's cytoplasmic location and Gal80p's dual cytoplasmic-nuclear location implicates new mechanisms for controlling Gal4p activity in Saccharomyces cerevisiae. *Molecular and Cellular Biology*, 20(14):5140–5148, July 2000.

[98] Adam Platt and Richard J. Reece. The yeast galactose genetic switch is mediated by the formation of a Gal4p–Gal80p–Gal3p complex. *Embo Journal*, 17(14):4086–4091, 1998.

[99] Bing Ren, Francois Robert, John J. Wyrick, Oscar Aparicio, Ezra G. Jennings, Itamar Simon, Julia Zeitlinger, Jörg Schreiber, Nancy Hannett, Elenita Kanin, Thomas L. Volkert, Chris Wilson, Stephen P. Bell, and Richard A. Young. Genome-wide location and function of DNA-binding proteins. *Science*, 290(5500):2306–2309, December 2000.

[100] John R. Rhode, Jennifer Trinh, and Ivan Sadowski. Multiple signals regulate GAL transcription in yeast. *Molecular and Cellular Biology*, 20(11):3880–3886, June 2000.

[101] J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

[102] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society B*, 49:223–239, 1987.

[103] Christopher J. Roberts, Bryce Nelson, Matthew J. Marton, Roland Stoughton, Michael R. Meyer, Holly A. Bennett, Yudong D. He, Hongyue Dai, Wynn L. Walker,

Timothy R. Hughes, Mike Tyers, Charles Boone, and Stephen H. Friend. Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science*, 287:873–880, February 2000.

[104] Frederick P. Roth, Jason D. Hughes, Preston W. Estep, and George M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnology*, 16:939–945, October 1998.

[105] A. Roy, C.F. Lu, D.L. Marykwas, P.N. Lipke, and J. Kurjan. The AGA1 product is involved in cell surface attachment of the Saccharomyces cerevisiae cell adhesion glycoprotein a-agglutinin. *Molecular and Cellular Biology*, 11(8):4196–4206, August 1991.

[106] Eric E. Schadt, Cheng Li, Byron Ellis, and Wing H. Wong. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. UCLA Statistics Series, 2000.

[107] Eric E. Schadt, Cheng Li, Cheng Su, and Wing H. Wong. Analyzing high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry*, 80:192–202, 2000.

[108] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Neural Information Processing Systems 1999*, 1999.

[109] Paul T. Spellman, Gavin Sherlock, M. Q. Zhang, Vishwanath R. Iyer, K. Anders, Michael B. Eisen, Patrick O. Brown, David 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.

[110] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

[111] Peter Spirtes, Clark Glymour, and R. Scheines. *Causation, Prediction, and Search.* Springer-Verlag, New York, 1993.

[112] Harald Steck. On the use of skeletons when learning in Bayesian networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 558–565. Morgan Kaufmann Publishers, 2000.

[113] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S. Lander, and Todd R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences, USA*, 96:2907–2912, March 1999.

[114] Saeed Tavazoie, Jason D. Hughes, Michael J. Campbell, Raymond J. Cho, and George M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, July 1999.

[115] Rene Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585, 1973.

[116] Simon Tong and Daphne Koller. Active learning for parameter estimation in Bayesian networks. In *Neural Information Processing Systems 2001*, 2001.

[117] Simon Tong and Daphne Koller. Active learning for structure in Bayesian networks. In *International Joint Conference on Artificial Intelligence 2001*, 2001.

[118] Eugene P. van Someren, L. F. A. Wessels, and M. J. T. Reinders. Linear modeling of genetic networks from experimental data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*. ISCB, July 2000.

[119] J. Craig Venter et al. The sequence of the human genome. *Science*, 291:1304–1351, February 2001.

[120] Thomas S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 220–227. Morgan Kaufmann Publishers, July 1990.

[121] Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

[122] Xiling Wen, Stefanie Fuhrman, George S. Michaels, Daniel B. Carr, Susan Smith, Jeffery L. Barker, and Roland Somogyi. Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Sciences, USA*, 95:334–339, January 1998.

[123] Frank Wittig and Anthony Jameson. Exploiting qualitative knowledge in the learning of conditional probabilities of Bayesian networks. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, 2000.

[124] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, and Terence P. Speed. Normalization for cDNA microarray data. In *International Symposium on Biomedical Optics (BiOS 2001)*, pages 141–152. SPIE, January 2001.

[125] Ken-ichi Yano and Toshio Fukasawa. Galactose-dependent reversible interaction of Gal3p with Gal80p in the induction pathway of Gal4p-activated genes of Saccharomyces cerevisiae. *Proceedings of the National Academy of Sciences, USA*, 94:1721–1726, March 1997.

[126] Julia Zeitlinger. Personal communication, 2001.