Prediction, expansion, and visualization of biological pathways and networks using perturbation data and cyclical graphical models

Charles Vaske

August 30, 2007

Contents

List of Figures

Abstract

Cellular processes are the interaction of multiple proteins, genomic sites, RNAs, small molecules, and their complexes. The set of these interactions and their contexts provide biological understanding of functionality beyond single-gene annotation. Biological networks have emerged as the dominant method of communicating, modeling, and understanding cellular processes and pathways.

Computational prediction of interactions and networks is an open problem under active research. New high-throughput experimental techniques for measuring and perturbing gene expression, detecting protein-protein interactions, and detecting protein-DNA interactions are providing rich datasets for predictions. Current methods are able to accurately predict some known pathways using of these data types.

However, current methods have some limitations and are not able to fully analyze current datasets. For example, current models cannot learn negative feedback, a common network motif. Limited models of gene regulation also fail to find additive interactions. Also, current learning methods do not explicitly provide biologists guidance on follow-up experiments.

I propose a new computational framework using factor graphs to address these limitations. The framework can model signed cycles of regulation, can incorporate multiple-gene knockdowns to infer additive interactions, and suggests follow-up experiments based on an information theoretic active learning approach. Preliminary results show the ability to recover signaling networks using gene-expression effects from gene knockdowns.

Finally, I propose a new web tool for visualizing networks. This tool will be able to confirm network predictions through comparison with independent high-throughput datasets. The web tool will also allow easy collaboration and sharing of network predictions and network layouts.

Part I

Background

Chapter 1

Biological Networks and Processes

Biological networks aim to describe the large scale activity of cells in terms of the interactions of a cell's components. Discoveries in molecular biology have told us that these activities are performed by various classes of biological entities: *e.g.* DNA, RNA, proteins, membranes, and small molecules. Molecular biology has also informed us of many common types interactions that happen between biological entities. Thus, biological network is a rather broad term, that can refer to a description of how a complex mulitigenic phenotype arises from genotype, to a simple biochemical reaction involving only an enzyme, substrate molecule, and a product molecule.

This network representation of biological processes, specifically a graph representation, pervades system biology [77]. The dominant machine readable formats, SBML [38, 18], BioPAX [59, 82], PSI MI [34] and CellML [11] all describe biological processes as graphs, but each format places different emphases on structure, dynamics, and supporting evidence.

In Parts II and III, I will focus on three primary interaction classes: complex formation, gene regulation, and signal transduction. A fourth class of interaction, biochemical reaction, is important and represented in nearly all systems biology ontologies, but there is currently no high-throughput assay that allows for computational predictions of such reactions, so I will not treat them explicitly. §1.4 in this chapter shows that omitting this type of interaction will still result in coherent networks.

1.1 Multi-component complexes and protein-protein interaction

Often, multiple molecules of RNA, DNA, or protein act together in a process. When these molecules stably bond, it is referred to as a complex. Networks model both the individual components and the complex.

As an example, the ribosome is a complex with a stable core set of RNA and proteins. The ribosome core consists of two major RNA subunits and many proteins. Humans have approximately 80 ribosomal proteins [81] and *E. coli* have more than 50 [3]. This core complex is stable enough to exist while the ribosome is not performing translation, and to be isolated from cell extracts. During translation, additional proteins and RNAs associate with the ribosome in a more transient manner. Biological networks attempt to capture and describe not only the stable parts of the complex, but also the transient associations.

The spliceosome, like the ribosome, consists of both RNA and protein, but instead of being a stable complex, assembles as needed. This complex removes introns from precursor mRNAs as part of mRNA processing. The spliceosome is itself modeled as a complex of complexes, each referred to in general as small nuclear ribonucleoproteins (snRNPs). Full complex assembly is performed in sequential steps, delimited by ATP-dependent energy wells [69].

1.2 Gene regulation

As the first step in the link between genotype and phenotype, the determining factor between morphologically distinct cells/tissues in the same organism, and perhaps the major evolutionary difference between humans and their closest neighbors [52], gene regulation features prominently in the structure of biological networks. Gene regulation refers to the production of an active product from the genome, where the product is a protein or RNA. Regulation can occur either by controlling the transcription of the gene or by modulating the steps in between transcription and transformation into the active product. Biological networks can model the state of the regulatory elements, the presence or abundance of active gene product, and the state of the gene product intermediates, though the simplest networks model only the presence of the final gene product. Often, models of gene regulation will not state the exact mechanism of regulation. A gene regulation network may elide some elements in a chain of multiple steps of regulation. For example, if a protein A activates chromatin machinery to silence a gene b, preventing the protein B from activating transcription of gene c, then it may be said that a regulates c.

Both activation and inhibition are important aspects of regulation, and both are found extensively in gene regulation networks. As an example of gene regulation, I will describe the regulation of the *E. coli lac*

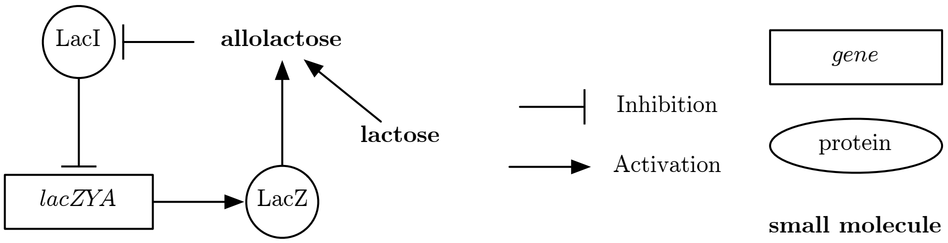


Figure 1.1: Gene regulation in the *E. coli* lactose metabolism pathway.

operon. Prior to its discovery, it had been known that some enzymes in bacteria would only be produced when needed. First reported by Jacob and Monod in 1961 [45], the *lac* operon was the first example of gene regulation to be characterized. This network is only activated in *E. coli* when its preferred sugar, glucose, is not present. In the presence of lactose and absence of glucose, *E. coli* uses the *lac* operon to convert lactose to glucose. The network that includes the *lac* operon exhibits activation, inhibition, multiple types of components, extra-cellular signaling, and cycles. This network has all the characteristics that I aim to predict.

Figure 1.1 shows the core of the *lac* operon network. The two proteins involved in this core, LacI and LacZ, are an inhibitory regulator and an enzyme respectively. The nodes **allolactose** and **lactose** are both small molecules. The node *lacZYA* is an operon, a gene with multiple products. The tee-arrow (⊣) indicates that the presence of LacI inhibits *lacZYA*. Similarly, the tee-arrow from node **allolactose** to node LacI indicates that the biochemical observation that allolactose inhibits LacI, preventing it from acting. The arrow (→) from node *lacZYA* to node LacZ indicates "activation," which mean that the target of the link, LacZ, is also activated, here by transcription and translation. There are two arrows into the **allolactose** node, one from LacZ and one from **lactose**, both indicating activation. In my representation of networks, the presence of more than one link into a node indicates that their effects combine "multiplicatively." This means that when the incoming links are activation, all of the regulators must be present in order for the regulatee to be activated. In this case, it has been experimentally observed that LacZ converts lactose into allolactose, and that the presence of both enzyme and substrate are necessary for the product.

Recall that the enzyme for conversion of lactose, LacZ, is only active when the substrate is present, meaning the LacZ is regulated by lactose. The cycle in the network explains how LacZ is regulated and produced in only the proper conditions. The cycle in the network consisting of nodes LacI, *lacZYA*, LacZ, and **allolactose**, is regulated by the upstream node **lactose**. Ignoring this upstream node, the sequence of links in the cycle dictates two consistent solutions for the presence/activation and absence/inactivation of the entities: { LacI = *inactive*, *lacZYA* = *active*, LacZ = *present*, **allolactose** = *present*} and { LacI = *active*, *lacZYA* = *inactive*, LacZ = *absent*, **allolactose** = *absent*}. When we consider the link from the node **lactose** to node **allolactose** we see that in the absence of **lactose** then **allolactose** must also be absent, and therefore node LacI must be active and node *lacZYA* inactive. When the node **lactose** is present, the other solution is consistent, as long as LacZ is also present. If there is absolutely no LacZ, then the addition of lactose will not change the system.

The usage of all these terms is very loose, and such looseness is necessary in order to have a generalized way of talking about these systems. The concept of "presence" and "absence" is different for different entities in the system. When referring to node LacI, the active and inactive refer to its ability to inhibit node *lacZYA*. Biologically, LacI binds the promoter of the *lacZ* gene, preventing transcription. Biochemically it was observed that allolactose is an allosteric effector of LacI, and when bound LacI can no longer bind the promoter, making LacI "inactive." However, when referring to LacZ, **allolactose**, and **lactose**, the terms "presence" and "absence" mean that the cell has a larger or smaller quantity of the entity. And though I refer to node LacZ as "absent," it is never entirely absent biologically, even in the absence of lactose. This is because *lacZYA* is expressed whenever not bound by LacI, and LacI binds loosely enough for a few errant transcripts to occur, resulting in a small amount of endogenous LacZ. This small amount allows the cycle to switch to the solution with **allolactose** present when lactose is introduced to the network.

Transcriptional repressors such as LacI are just one type of a larger class called transcription factors. There are also transcriptional activators, that increase the rate of transcription. Transcription factors act on *cis*-regulatory elements, DNA sequences that are on the same DNA molecule that contains the gene.

Eukaryotes have a very rich toolbox for regulating gene expression beyond *cis*-regulatory elements. The methods of regulation in eukaryotes include chromatin structure/domains [48], DNA methylation and imprinting [71], microRNAs [33] and RNA mediated interference [19], nonsense-mediated decay [57], and even the three-dimensional organization of chromosomes in the nucleus [20]. Many of these regulatory methods have only been recently discovered, and advances in molecular biology may discover yet more.

1.3 Signal transduction

Signal transduction is the process of a cell turning the perception of something external, usually a molecule, into a response inside the cell. This response will almost always be energy dependent, and often involve protein phosphorylation. I will first describe the *E. coli* how the *lac* network performs a function like signal

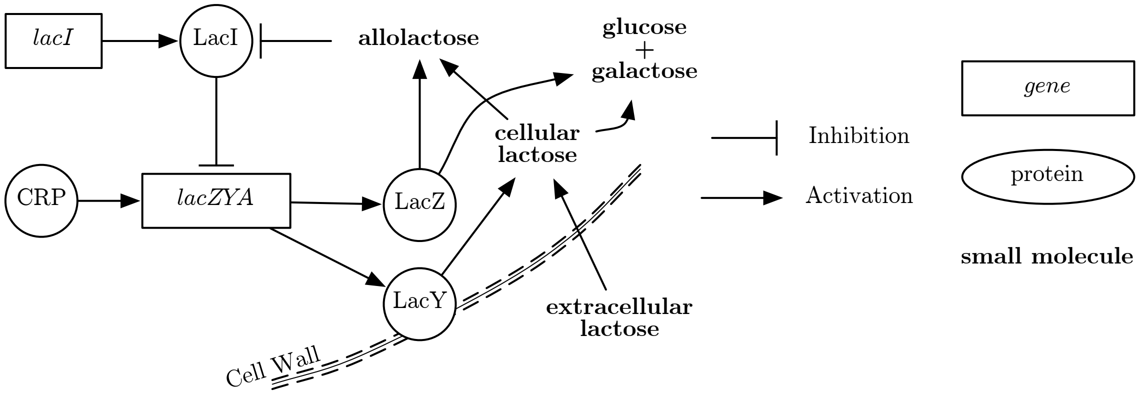


Figure 1.2: Extended *lac* operon network.

transduction. Then, I will explain some of what is known about signal transduction in the datasets that I use later in this proposal.

Figure 1.2 shows an elaborated network for the *lac* operon. The core components are LacI, *lacZYA*, LacZ, **cellular lactose**. What was labeled **lactose** in Figure 1.1 has been more accurately relabeled as **cellular lactose** here. The new network includes LacY, another product of the *lacZYA* operon, which is a transmembrane protein that uses ion flux to move lactose into the cell. In the absence of extracellular lactose, there are very low levels of LacY and LacZ, since LacI almost fully represses *lacZYA*. When extracellular lactose is present, it will interact with the small amount of endogenous LacY, resulting in a small amount of cellular lactose, which results in a small amount of allolactose, releasing LacI and permitting the *lac* operon to be transcribed.

The eukaryotic signal transduction repertoire includes ion-flux transporters as in *E. coli* and many kinase based systems. These are referred to as mitogen-activated protein (MAP) kinases, since they are activated as the result of external small molecules. In the later sections I will predict networks on two pathways induced by signal transduction through kinase: one path is a member of the G-protein couple receptor (GPCR) family, and the second pathway is a member of the receptor tyrosine kinase (RTKs).

GPCR based signal transduction is responsible for a large class of the studied cellular responses in humans, including processes as broad as vision, neurotransmission, and histidine response. GPCRs are integrated into membranes with seven transmembrane alpha helices. The portion of the GPCR on the exterior of the membrane binds to specific ligand or a specific class of ligands. When the ligand is bound, the conformation of the GPCR on the inner side of the membrane changes, through mechanisms that are not entirely understood. On the inner side of the membrane, GPCRs are coupled to G proteins, which are named for their guanosine binding properties. G proteins are localized on the cellular membrane, next to a GPCR. Upon conformational change of the GPCR, the G protein exchanges bound guanosine diphosphate for guanosine triphosphate, and is no longer localized to the cell membrane. This G protein is now activated to continue a MAP kinase response elsewhere in the cell.

Similar to GPCRs, RTKs are located in the cellular membrane and contain a ligand-binding receptor domain on the exterior of the membrane. Upon binding a ligand in the receptor domain, the RTK is activated to phosphorylate a tyrosine target on a phosphotyrosine binding (PTB) domain on another protein.

1.4 Scope in biological networks

An important aspect of biological networks for my proposal is that they can consistently and accurately be viewed at multiple scope and scales. By scope, I mean the portion of the network that is under examination or investigation. In Figure 1.2, the network shows some indication of how the *lac* network connects to other parts of the entire cellular network. The protein CRP is another transcriptional regulator that effects not just the *lac* operon, but almost 200 other transcriptional units in *E. coli* according to the database EcoCyc [51]. Also shown are the primary products of LacZ, glucose and galactose. Using data from EcoCyc, we could connect **glucose** to four other biochemical pathways in *E. coli* as a substrate.

In principle, the entire cell and all its functions could be modeled with such a network of a very large size. Despite the size and interconnectedness of the cellular network, we are able to narrow the scope of investigation to a small number of genes.

In addition to narrowing our investigation to a small number of entities, we can also narrow scope to just a few types of entities and retain a consistent and informative view of the biological process. Figure 1.3 shows a network of just two proteins and the external lactose input. If we were only able to measure these two proteins, and control the external availability of lactose, this would still be an accurate description of the network.

This consistency under reduction of scope is essential to our ability to investigate and discover biological networks. We currently have no physical techniques for the simultaneous measurement of all entities in a cellular network, and the size of such measurements would outstrip our mathematical and computational tools for inference. However, we are able to explore small pieces of the network, and even without knowing the full context of the scope, we are able to accurately infer interactions.

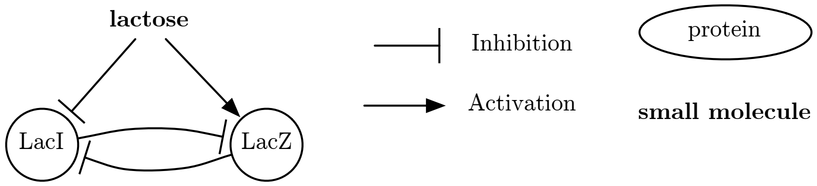


Figure 1.3: A reduced scope *lac* network.

When reducing scope in this way, our activation and inhibition links may no longer correspond to direct physical interaction or immediate causes. The links from **lactose** node to the protein nodes are direct in Figure 1.3. However, there is a distinction between the model and the known biology: we know that the physical interactions are mediated by allolactose in the case of the link from the **lactose** node to LacI. The chain of physical interactions is longer for the link from node **lactose** to LacZ. The entities **allolactose**, LacI, and *lacZYA* are all involved in describing the activation link from node **lactose** to node LacZ.

Even the extended network in Figure 1.2 omits much of our knowledge of the network. For example, we know more of the structure of the promoter of *lacZYA*, and we ignore such essential components such as the transcription and translational machinery.

This consistency under change of scope is particularly essential to my aims. My proposal focuses on determining these networks not by measuring any of the entities directly, but only by looking for downstream changes in gene expression. In addition, I am only placing proteins in the network, ignoring genes and small molecules. Under these conditions, I expect to predict networks that are consistent with the true biological network, but amenable to further study in two ways. First, there may be additional intervening proteins on the links that I predict. Second, by adding more types of entities using other investigation techniques, my predictions could be filled with more direct physical causes.

Chapter 2

Measuring Biological Networks

The discovery of the *lac* operon network was the culmination of many studies using wide array of techniques to measure and perturb biological entities. Today, the biological techniques for network inference follow the same lines. The particular techniques for measurement and perturbation have become easier, more directed in the case of perturbation and more general in the case of measurement, but the principles are the same

This chapter discusses the biological methods that are useful for investigating networks and are relevant to my aims. I will first discuss the methods for perturbing elements in the network. I will then discuss the measurement technique that my method uses. Finally, I will discuss high-throughput methods for detecting the presence of direct links in the biological network, which I will use for verifying my predictions.

2.1 Perturbation Methods

Perturbation is an extremely powerful tool for establishing causal relationships, as discussed in §3.4. The essence of the increased causal power of perturbation is that perturbation disconnects the perturbed element from its other causes, and therefore causes a structural change in the network. There exist several techniques for directing perturbation of biological systems. These fall into two classes: genetic perturbation and impulse perturbation. With genetic perturbation, measurements can only be taken at least one cell cycle after the perturbation has occurred, allowing responses to fully propagate throughout the cellular network. Impulse perturbation allows measurements to be taken within the same cell cycle, when not all responses to perturbation have yet traveled through the cellular network. Both techniques are used in the datasets I will analyze in my aims.

2.1.1 Genetic perturbation

Genetic perturbation usually involves reducing the functionality of a gene. This can be done either by gene deletion in which case there is no gene function, or by merely reducing the activity, resulting in a hypomorph.

Gene deletions are of such utility that the Saccharomyces Genome Deletions Consortium has deleted over 90% of *S. cerevisiae*open reading frames [80, 30]. These deletions proceed in several steps [5]. First, a DNA construct is created to replace the target gene. This construct contains a selectable gene, so that cells with successful replacements can be differentiated from those that have not been successfully replaced. The other ends of the construct are homologous to the 5' and 3' end of the target gene, allowing for recombination once the construct is inserted into cells. These constructs are inserted into a yeast culture using a lithium acetate treatment [31], which is called transformation in microbes. Similar techniques have been developed for *V. cholerae* [23, 58].

Since such deletions last for the entire lineage of the cell, the effects on the rest of the cellular network are very broad. There has been some concern that using observations from broad changes will make confound network inference. However, recent experiments with synthetic data [74] give preliminary indications that long term perturbations like these in fact allow better predictions.

2.1.2 Impulse perturbation

Impulse perturbation, unlike genetic perturbation, allows for the evaluation of the change in a network over time. The cellular network will respond to such perturbations dynamically, and depending on when measurements are taken after the perturbation, different effects will be observed. Though I do not use such temporal effects in my proposed methods, it is important to note the effect as it can have consequences for the interpretation of observations of the network and is a concern when using data of this sort.

I will discuss three different types of impulse perturbations. In the *E. coli lac* network we saw an example of lactose, an external signal, providing a perturbation in the network. There are two techniques for affecting specific entities with in the cell: genetic modifications allow the creation of proteins that are sensitive to certain drug or temperature treatments, and the RNAi system in many eukaryotes.

Signal transduction, as discussed in §1.3, induces changes in the state of the network in the cell. These external stimuli activate, or deactivate, different parts of the cellular network. These perturbations are extremely useful for discovering which biological entities are related to each other, by identifying a connected component of the cellular network. However, these perturbations are not structural, in that they do not disconnect any elements from their causes, since by definition external variables do not have any causes within the cell.[1]

An example of impulse perturbation of internal elements of the network is the tetracycline-regulatable promoter in *S. cerevisiae* [25]. In this system, a gene is placed behind a special promoter such that whenever the cell culture is exposed to tetracycline that the gene is effectively transcriptionally silent. When tetracycline is not present, the promoter is promiscuous and transcription of the gene may be increased 1000-fold. Here, it is clear how perturbation removes other causes: the promoter is changed such that the normal inputs into the gene are entirely missing. The normal gene regulatory program has been replaced with one that is easy to manipulate externally.

Microbial organisms have long had systems for genetic transformation, but a quick and inexpensive system for general perturbation of metazoans has only recently been discovered. First reported in *C. elegans*, cytoplasmic double-stranded RNA induces a pathway that digests messenger RNAs complementary to the double-stranded RNA. This response is called RNAi, and it has quickly gained popularity as a method performing gene knockdowns in many eukaryotic systems. Since its initial discovery, it has been found that regulation via micro RNAs shares many of the same components as the RNAi pathways [53]. The two primary entities involved in the process are called Dicer and RISC [42]. If the double-stranded RNA is long, then the Dicer enzyme cuts the dsRNA into 21-25 nucleotide double-stranded fragments.

Any such small 21-25 nt dsRNA, referred to as small interfering RNA (siRNA), is incorporated into the RNA-induced silencing complex(RISC). The siRNA strands are unwound to single strands, and the RISC complex remodeled in an ATP-dependent process, which results in an activated RISC capable of recognizing and degrading complementary RNA. This process also works with synthetic siRNA, so by inserting the siRNA into the cell, nearly any transcript can be targeted with specificity.

2.2 High-throughput gene expression measurement

In recent years, gene expression profiling has emerged as a powerful tool for quickly and easily assaying thousands of phenotypes in a cell. These phenotypes are particularly valuable, as they correspond directly to both a gene and usually a protein, which give obvious directions for further investigation. Gene expression profiling also provides a wider scope on the cellular network than any other single technique available, in that they capture an essential stage for a very large proportion of the known cellular network. Whole-genome gene expression microarrays also offer a relatively unbiased way to search for activity, something that few

[1] Note that so far we have only discussed cellular networks, and not discussed the interaction of a cell with its environment. In some situations, such as quorum sensing signals in *V. cholerae* growth, the cellular network affects the extracellular environment and other cellular networks.

other techniques can boast.

First used to report on 45 gene expression profiles in *Arabidopsis thaliana* [72], DNA microarrays are somewhat analogous to a large-scale Southern or Northern blot. Gene expression microarrays aim to quantify the amounts of many different types of RNA species in a cell. This is done by using tendency of nucleotide sequences to base-pair, or hybridize. An individual run of a microarray is often called a hybridization.

For each RNA assayed by the microarray, there are one or more single stranded DNA probes of length 25-1000 nt which complement that RNA. Most probes are designed to match only one sequence of RNA or a single gene. When a probe complements more than one sequence (perhaps with a few mismatches), that probe will suffer cross-hybridization, which complicates the interpretation of that probe's signal. Additional issues at the probe result from variation in melting temperature, density of probe DNA, and homogeneity of the probes. Various wet-lab and mathematical techniques have been developed to normalize different probes to each other to allow comparison of responses between probes.

There are many types of microarrays, but they can be grouped into categories by a few parameters. First, there are cDNA and oligo arrays. In cDNA arrays the probes are made from reverse-transcribing full length mRNAs, and can be of varying length and quality. Oligo arrays use short, synthetic DNA probes, with uniform lengths between 25-70 nt. Though they are a consistent size, oligo probes can vary in their binding affinity due to differences in GC-content or because of secondary structure effects.

2.3 High-throughput network structural measurement

At the most detailed scope of the cellular network, links are defined by physical interactions. Ultimately we hope to find explanations of biological processes that identify all the components, and specifically notice how they interact. Lab techniques used for detecting protein-protein and protein-DNA interactions are now being scaled to the degree that thousands or tens of thousands of such assays can be performed in a single study. This permits investigation of the structure of the cellular without bias towards previously investigated genes, and provides a dataset which is useful to investigations that later use any of the assayed genes.

2.3.1 Protein-Protein Interaction

The network of protein-protein interactions determines much of the skeleton of allowed pathways. Nearly all known cellular processes, from transcription and translation to signal transduction, depend on the binding of proteins to each other in a highly specific manner. These protein-protein interactions can refer to transient binding, more lasting binding in a complex, and the "self" binding of homo-multimers.

Ascertaining protein-protein interactions is complicated by the context specificity of protein interaction. Many protein-protein interactions only make sense in a specific context. For example, many mitogen activated protein kinase signaling pathways are highly localized, and this localization is essential the signal specificity [10]. Some protein-protein interactions must be mediated by chaperons. The PDZ family of domains [41], present in both prokaryotes and eukaryotes, as well as approximately 350 human proteins [56], bind specific peptide sequences to assist in the assembly of complexes and general protein targeting. Presently both *in vitro* and *in vivo* methods suffer this specificity problem. Using an localization database in conjunction with protein-protein interaction can help resolve this context specificity.

The two-hybrid system is a technique for reporting when two proteins bind well enough to activate a transcript. It requires fusing a domain to each queried protein. The TAP-Mass spectrometry system requires fusing a sequence to each queried protein. Most high-throughput studies have been performed in yeast, which has well-established and favorable genetic systems and culturing conditions for performing large scale experiments.

Two-hybrid systems

Initially reported in 1989 by Fields and Song [16] in *S. cerevisiae*, the two-hybrid system is inspired by the activity of the GAL4 gene. GAL4 contains two domains, one which binds upstream of its genomic location, and own which activates transcription of itself. Both domains are required, if the activation domain is not localized to the promoter the activator will not work and if the binding domain is not attached to the activating domain, GAL4 is not activated. To test if two proteins X and Y interact, two hybrid proteins are created. The first hybrid protein is the fusion of the binding domain of GAL4 and X, the second hybrid protein is the fusion of Y and the GAL4 activating domain. These hybrids are introduced into strains without GAL4 and with a β-galactosidase reporter gene downstream of the sequence bound by GAL4. Significant amounts of β-galactosidase activity then indicate binding of proteins X and Y.

By 1994, the two-hybrid system was in widespread use [17], using a variety of reporter, binding, and activating constructs. The system has also been expanded to DNA-protein, RNA-protein, and protein-small molecule interaction [79]. The first genome-wide study of protein linkage was conducted in 1996 in *E. coli* bacteriophage T7 [4].

Two independent, large-scale investigations in *S. cerevisiae* used the two-hybrid system to determine interactions. Both studies created libraries of strains for both binding and activating hybrids, and then crossed all strains to create an array of double-hybrids. Uetz *et al.* [78] compared two different methods of detecting positives in a 192 by 6000 (binding and activating hybrids, respectively) screen, one with higher accuracy and one with greater throughput. Ito *et al.* [44] completed a thorough scan (3,278 proteins in interactions) subsequently, with somewhat non-overlapping results.

Due to the necessary gene fusion steps, two-hybrid systems inherently have high false negative rates [46]. The fused binding or activating domain has the potential of obstructing both normal protein folding and the interacting sites of the proteins. The Uetz *et al.* higher accuracy screen found an interaction for a binding hybrid only 45% of the time.

TAP-Mass spectrometry

Increasingly precise mass spectrometric methods now allow the identification of proteins and protein complexes from cell extracts. A single species, isolated *e.g.* by gel electrophoresis or centrifugation, may be digested by trypsin, resulting in small fragments whose amino acid constituents can be identified via mass spectroscopy. Searching against a database of potential peptide sequences can quite often identify unique proteins that match the observed weights.

In order to improve isolation of single species of compounds, studies often use a single “bait” protein, which has been modified with an easily immunoprecipitated tag [9]. The FLAG tag is particularly popular due its ability to precipitate without denaturing complexes [14]. Interactions found this way may not necessarily be direct since a whole compound may be pulled down and not all members may share full contact.

Again in *S. cerevisiae*, two genome wide screens have been performed to find binding ability [35, 29]. Rates of positive interactions were much higher than in the yeast two-hybrid experiments, with 78%–82% of the baits finding partners, compared to a best case of 45%. Positives were also much more repeatable than in the two-hybrid studies, approximately 75% vs. 20%.

Databases

There are several databases of protein-protein interaction with both manually-curated interactions taken from literature and computationally predicted studies. These databases are generally used as the gold standard in verifying high-throughput studies. Mathivanan *et al.* [62] compare the human specific parts of seven such databases (BIND, DIP, IntAct, Mint, MIPS, PDZBase, and Reactome) to their own database of protein-protein interactions [68]. Several of these databases include additional information beyond protein-protein interaction.

2.3.2 Protein-DNA Interaction

Chromatin immunoprecipitation with DNA microarray analysis (ChIP-chip [36]) or DNA sequencing (ChIP-sequencing [43]) promises to give high-throughput results of protein-DNA interaction. Previous wet-lab methods for determining protein-DNA interaction included DNase footprinting, primer extension, and gel shift assays, and were generally limited to a very small number of queries. Alternatively, a protein's binding site could characterized computationally (often by a weight matrix), and then genomic binding sites could be predicted, usually with a high false positive rate.

ChIP-chip and ChIP-sequencing experiments first cross link transcription factors to bound DNA with formaldehyde *in vivo*. DNA-protein complexes are extracted and cut randomly via sonication or other method. The protein, and any cross-linked DNA, is selected via immunoprecipitation of the target factors or epitope tags. Finally, the DNA is amplified via PCR, and the sequences that were bound are queried with either microarrays or with DNA sequencing. Both methods produce similar results among their top-ranking predictions [15], but require appropriate controls for identifying entirely novel binding motifs [47].

Large-scale ChIP-chip studies have been published in human [87] and *S. cerevisiae* [55]. In *C. elegans*, protein-DNA interactions have been investigated using a yeast one-hybrid system [12].

Chapter 3

Probabilistic and Other Graphical Models

Computational tools for dealing with networks of variables under probabilistic constraints have been developed and widely used in statistical physics, machine learning, computer vision, coding theory, and much of bioinformatics. These computational tools are now being applied in biological networks, both for modeling and discovery. In this chapter I will discuss common types of computational graphical models, algorithms for inferring values from a given model, algorithms for learning a model from data, and the implications of causality and perturbation in these models.

3.1 Graph formulations

Traditionally, probabilistic graphical models are presented in two categories: undirected and directed models. In both of these types of models, each node of the graph is a random variable. The edges in both types of models encode the probabilistic dependencies between random variables, though both models encode the dependencies in different ways. Decoding the probabilistic dependencies requires examination of the local structure around a variable.

These two categories, also known by the names Markov random fields and Bayesian networks, are capable of representing different probability spaces, but share some overlap. Sometimes, a third type of graphical modeling containing both directed and undirected edges is presented to generalize the two and unify the set of representable probability spaces. However, a different formulation of this generalization, called factor graphs, has seen growing popularity. I find the factor graph representation far preferable to the mixed directed/undirected model formulation, and in many cases preferable to both Markov random fields and Bayesian networks due to the explicit representation of the characteristic function of the network.

Before proceeding with the definitions, I will define some notation conventions. First, capital letters such as A or X refer to variables over a domain. In general, the domain of a variable can be any set, countable or uncountable, but in my proposal I use only finite sets or the real numbers, $\Re$.

A "factor" is a function whose domain is a set of variables and whose range is the real numbers. Many functions can be interpreted as factors, so their notation varies. For example, $\text{Prob}\,(A, B)$, f_{AB}, or $\text{Prob}\,(A|B)$ could all be considered factors in the rest of this proposal. Factors are sometimes referred to as potentials. Both joint and conditional probability distributions are quintessential examples of factors, though they need not be restricted in the ways that probability distributions are defined. For example, the identity $F = MA$ over three variables with a real-valued domain could be defined as a factor N with f, m, and a all real numbers:

$$N(f, m, a) = \begin{cases} 0 & \text{if } f = ma \\ -\infty & \text{otherwise} \end{cases}$$

As is often done with probability distributions, I will use abbreviated notation for factors, where using sets as argument instead of a single values represents a new factor. The domain of this factor is the cross product of all the sets that were used as arguments, and the range is the values of the original factor on each value in the domain.

There are two primary factor operations that are used in graphical models: factor product and factor marginalization. The product of two factors $f(X)$ and $g(X, Y)$, denoted $f(X)g(X, Y)$ or fg, is another factor. The resulting factor's domain is the cross product of the union of the domains of each operand, $X \times Y$ in this example. The value for each element in the result is the value of operand evaluated at that point, *i.e.* in this example $(x, y) \mapsto f(x)g(x, y)$ for all $x \in X$ and $y \in Y$. The other operation used on factors is marginalization, sometimes called summarization. Marginalization results in a factor with one less variable in the domain. There are two variants of marginalization which are commonly used; one variant uses addition and the other the max function. If X in the above examples is discrete, then marginalization of X out of g is denoted and defined as:

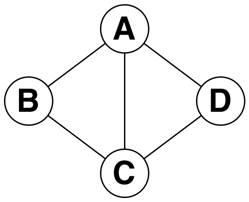


Figure 3.1: Example Markov network.

$$\sum_{X} g(X,Y) \equiv y \mapsto \sum_{x\in X} g(x,y) \text{ for all } y \in Y$$

If X is a continuous variable, then

$$\int_{X} g(X,Y) \equiv y \mapsto \int_{x\in X} g(x,y) \text{ for all } y \in Y$$

It is possible to marginalize a factor down to zero variables, in which case the result is a factor with no variables and a single real number in the range. This full marginalization has sensible interpretations in some contexts: marginalizing probabilistic factors by addition results in a probability mass, marginalization by max of probabilistic factors results in the most probable assignment, and marginalization of an arbitrary factor results in the partition function.

3.1.1 Markov random fields

Markov random fields were originally developed in statistical physics to describe systems of small particles, where the state of one particle interacts with the state of nearby particles. Such interactions are represented by lines between variables. A Markov network describes a probability distribution over its variables. For every maximal clique C in the graph, a factor ϕ_C describes the interactions over that variable. If the set of all such factors is denoted Φ, then the probability distribution over all variables $\bar{X}$ is defined to be:

$$\text{Prob}\left(\bar{X}\right) \equiv \frac{1}{Z} \prod_{\phi_c \in \Phi} \phi_c \tag{3.1}$$

For the example in Figure 3.1, $\text{Prob}\left(A,B,C,D\right) = \frac{1}{Z}\phi_{ABC}(A,B,C)\phi_{ACD}(A,C,D)$. The constant Z is known as the partition function, and is calculated by $Z = \prod_{\phi_c\in\Phi} \phi_c$. Though the expression is simple, such a calculation is not usually trivial, and is in effect similar to calculating the probability of the data in a Bayesian statistics model. There is flexibility in this parameterization, as the weights for $\mathrm{Prob}\left(A, B\right)$ can be distributed in either ϕ_{ABC} or ϕ_{ACD}.

3.1.2 Bayesian networks

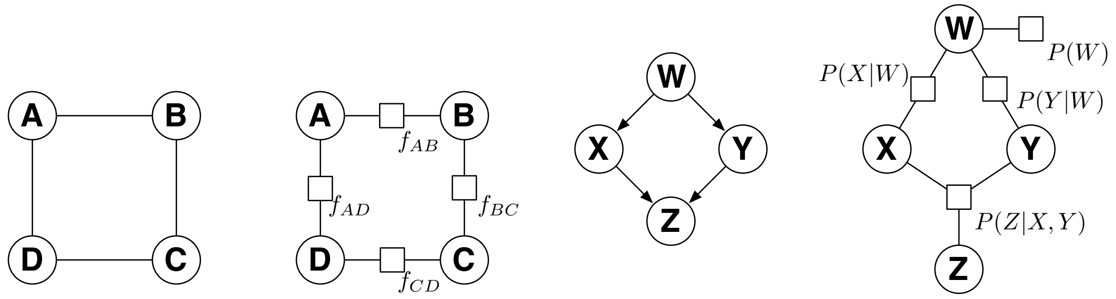
Bayesian networks are a representation of a probability distribution based on the conditional probabilities. Conditional probabilities offer the advantage of often being able to characterize given a known real-world system, and given a known Bayesian network the meaning of each conditional probability has a fairly clear interpretation. This is an advantage over a clique potentials in Markov random fields, which may have an unclear meaning. However, specifying a Bayesian network from a probability distribution is done through the conditional independences, which can be difficult to assess.

In a Bayesian network, there is a conditional probability distribution for each node. The directed graph structure is determined by these conditional probability tables, there is an arrow into each node for every variable on which it is conditioned. Additionally, this directed graph must be acyclic. Let $\bar{X}$ be the set of variables in the network, and $\mathrm{Parents}\left(X\right)$ for $X \in \bar{X}$ be the set of variables on which X is conditioned.

$$\mathrm{Prob}\left(\bar{X}\right) = \prod_{X \in \bar{X}} \mathrm{Prob}\left(X \middle| \mathrm{Parents}\left(X\right)\right) \tag{3.2}$$

There is great flexibility in the parameterization here, as in the Markov Random Field case, since any probability distribution can be expanded into conditional probabilities in any order. There are many cases where the probability distribution can be conditioned in a different order, but still encode precisely the same conditional independences, resulting in a network with flipped arrows but the same probability distribution as the original Bayesian network. For this reason Bayesian networks can be slightly deceptive, as the directionality is sometimes assumed to encode causality, but this may not be the case.

Many graph-based bioinformatics problems are easily formulated as Bayesian Networks of a very certain form. Algorithms on phylogenetic trees, for example, are special cases of the algorithms used on Bayesian networks. Hidden Markov models can be represented as a chain with one variable for each hidden state and one variable for each observed symbol.



(a) A Markov random field $\text{Prob}\,(A, B, C, D) = \frac{1}{Z} f_{AB} f_{BC} f_{CD} f_{AD}$

(b) A Bayesian network $\text{Prob}\,(W, X, Y, Z) = \text{Prob}\,(Z|X, Y)\,\text{Prob}\,(X|W)\,\text{Prob}\,(Y|W)\,\text{Prob}\,(W)$

Figure 3.2: A Markov random field and a Bayes next to their corresponding factor graphs.

3.1.3 Factor graphs

Both Markov random fields and Bayesian networks are mathematically specified by an objective function, namely their probability distribution. Factor graphs are a representation of any such objective function over a set of variables, and thus generalize both Markov random fields and Bayesian networks in that sense.

Figure 3.2 shows the factor graph representations of both a Markov random field and a Bayesian network. Factor graphs represent both the variables as nodes and the factors as nodes, with edges from each factor tho the variables in that factors domain, resulting in a bipartite graph. A factor graph is then a very general representation of constraints on variables, and has even been used to represent problems such as n-SAT [63] and fast Fourier transforms [1, 2].

3.2 Inference methods

Inference in graphical models aims to find out something about the distribution of variables. For example, common goals are to calculate the maximally likely assignment of values to each variable or the distribution of some set of variables given observations of the value of some disjoint set of variables. In a probabilistic setting, any conditional query reduces to two

$$\text{Prob}\left(\bar{X}|\bar{Y}\right) = \frac{\text{Prob}\left(\bar{X}, \bar{Y}\right)}{\text{Prob}\left(\bar{Y}\right)} \tag{3.3}$$

Such computations are in general bounded by the size of the largest domain during execution, since the size of a discrete domain factor grows exponentially with the number of variables in the domain. Therefore the main aim of inference algorithms is to minimize the largest such factor that is created during the calculation

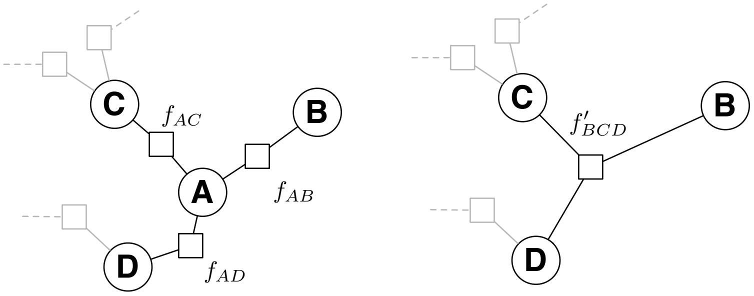


Figure 3.3: Variable elimination step for A. The new factor f'_{BCD} is equal to $\sum_A f_{AC} f_{AD} f_{AB}$.

of the final answer.

Inference algorithms for Markov random fields, Bayesian networks, and factor graphs are common to all representations [54]. I will illustrate the inference algorithms I have used in my preliminary results and that I plan to use in terms of factor graphs, as the representation of both variables and factors in the graph allows for simpler and clearer description.

3.2.1 Exact Inference

There are two basic methods for exact inference in graphical models: variable elimination and message passing on tree structures. Both of these algorithms take as input an ordering over all the variable nodes in the graph, and the largest factor used in algorithm depends on both the ordering and the structure of the graphical model. The general solution to finding the most efficient such ordering is NP-hard, but there are certain cases where an ordering is known to be optimally efficient. For example, in factor graph trees, any order that respects the connectivity of the tree is optimal.

Variable elimination transforms the factor graph, removing one variable per step, until only the variables of interest are left. The elimination of a variable node removes that variable node and all adjacent factor nodes, replacing them with a factor node. If the neighbors of variable X are the set of factors F, then the factor f' that results from eliminating X is:

$$f' = \sum_X \prod_{f \in F} f \tag{3.4}$$

Figure 3.3 shows the graph transformation under elimination of A. The memory and time costs of variable eliminations step depend on the maximum size of product in equation 3.4. Heuristic variable elimination algorithms try to choose an ordering of variable elimination that results in the smallest such product.

In the special case of a tree-shaped factor graph, it can be shown that the most efficient ordering always takes a variable node with the fewest number of neighbors. Using transformations of the graph, namely by combining two variables into a single variable with a larger domain, any cycle in the graph can be eliminated, resulting in a tree structure. Such a structure is called a junction tree.

Variable elimination on the tree collapses in a predictable manner, without creating a factor nodes that grows beyond a predictable size. This collapse of the tree suggests an algorithm where messages are passed between all nodes, each message a factor itself which represents the current local belief in the setting of a variable or a set of variable. If, during variable elimination no nodes are removed from the graph, but instead the newly created factors are stored as "messages", the results can be remembered and reused such that all variables are solved for in turn. I will discuss this algorithm in greater detail in the next section, where the message passing does not occur on a tree, where exact answers are guaranteed, but on a cyclic structure, which will approximate the appropriate answer.

3.2.2 Approximation with message passing

The message passing algorithm, also known as belief propagation or affinity propagation, has been invented many times, at least twice in the coding community with low-density parity check codes [24] and turbo codes [6], and once in Bayesian network community [67]. It has gained great popularity, as the message passing has been found to approximate exact results with very little computation in difficult problems. Approximate message passing has seen a surge of recent interest, with the creation of generalized message passing algorithms [86, 85] and surprisingly successful applications on problems such as clustering [21].

There are two types of messages passed in the algorithm. Variable nodes pass messages to factors summarizing the current belief in the value of the variable. Factor nodes pass messages to variables summarizing what the most likely value of the variable is, based on the other variables in the factor and the joint valuation encoded in the factor. Note that every message is a valuation over the settings of a single variable. Let $m_{n_1 \to n_2}$ denote the message form node n_1 to node n_2, and $\text{Neighbors}(n)$ the set of all nodes adjacent to n. Then the message sent from a variable node v to a factor node f, where $f \in \text{Neighbors}(v)$ is simply:

$$m_{v \to f} = \prod_{f' \in \text{Neighbors}(v) \setminus \{f\}} m_{f' \to v} \tag{3.5}$$

All the incoming messages are factors over just the variable v, so therefore all the outgoing messages have the same domain. The message from a factor node f to a variable node v is calculated by:

$$m_{f \to v} = \sum_{\text{Neighbors}(f) \backslash v} f \cdot \prod_{v' \in \text{Neighbors}(f) \backslash v} m_{v' \to f} \tag{3.6}$$

Note that the product of the factor and the incoming messages is marginalized by every variable except for v, ad there the message is a factor over just the domain of v. In this scheme messages are passed iteratively. At any given moment, the belief in a variable X is approximated by multiplying all the incoming messages.

Scheduling of message passing is an area of active research, with few general results. In the case of a tree factor graph, waiting until all incoming messages are ready, and passing each message at most once results in the exact result. When there are cycles in the graph, passing messages according to a schedule that causes data to be counted more than once can lead to poor approximations.

Generally, message passing is performed until the messages "converge," usually detected by measuring successive changes in the messages. Some difficulties can be encountered when there are long-range correlations, meaning when the value of one variable is highly correlated to the value of a variable a large number of nodes away. Additionally, it is known that message passing can be prevented from detecting convergence when there is a multi-modal cycle to the pattern of the messages. Message damping by averaging successive messages can help prevent such oscillations.

3.3 Structure learning

Though an important aspect of graphical models, there are few general methods for structure learning as it is a hard problem. If nothing is known *a priori* about the structure of the variable distribution, then the amount of data required to learn about conditional dependencies and independencies grows with the number of variables, making it impractical in most conditions. Therefore, most structure learning is highly dependent upon the nature of the data and problem space. In the next chapter, Chapter 4, I describe some previous methods for structure search for finding biological networks.

3.4 Causality in graphical models

The arrows in Bayesian networks are highly suggestive of causal influences, particularly when combined with the common introductory Bayesian network examples. However, a Bayesian network need not represent any causal structure at all, and for any Bayesian network which coincidentally does represent a causal structure, there are many other Bayesian networks which represent the exact same probability space but have entirely different structures. Therefore, Bayesian networks are not causal networks, and learning a Bayesian network on a dataset will not learn causality on that dataset.

The notion of causality has often been avoided by statisticians, as there are many philosophical pitfalls to avoid and a general lack of theory. However, recent years have seen some theoretical work on rigorously defining causality in probabilistic and modeling terms. In particular, a framework of causality has been developed with directed and undirected graphical models [66].

As it concerns my aims and preliminary results, causation provides a framework for predicting effects under the perturbation, or intervention, of a variable. In particular, perturbing a variable disconnects that variable from its causes, while leaving the effects of that variable intact. Thus, a full causal model predicts not only a standard distribution over the variables, but also predicts the distribution of the variables under all perturbations.

The methods in my preliminary results and research aims use causal models in this form. These methods use graphical models that specify how biological elements interact together, but also predict how they behave under perturbation. In this sense, these methods are causal, and links in the network represent cause and effect. These causes may be from direct physical interaction, or they may represent the transitive chain of several direct physical interactions. The methods that I propose in §10.3 and §11.3 aim not only to expand upon the network, but also to help find more direct causes when a predicted link is actually the result of a chain of physical interactions.

Chapter 4

Network Prediction Methods

Computational construction of biological networks from high-throughput data is an active area of research. Gene expression data, protein-protein interaction data, and protein-DNA interaction data have all been used in various combinations to predict networks. In recent years, availability of gene-expression under knockdown has increased the number of methods dealing with such data. In this chapter, I will first review the methods used to infer the connectivity of the *lac* operon in *E. coli.* I will then describe the methods used for modeling biological networks, followed by some methods used to predict pathways from data *de novo*.

4.1 Inferring the *lac* operon network

Recall the *lac* operon network from Figure 1.2. This network describes the activation of the *E. coli* lactase enzyme, LacZ, in the presence of lactose. The protein LacI serves as a general inhibitor of LacZ and the operon *lacZYA*, preventing the expression of both the lactase and the transporter which moves lactose across the cell wall. However, if there is a small amount of LacZ and a small amount of cellular lactose, then LacZ will produce some amount of allolactose, which inhibits LacI, freeing up the promoter of *lacZYA* and activating the metabolic pathway in response to an external signal.

All of this was discovered in piecemeal fashion through the use of structural perturbations to the cellular network. This network was the culmination of over a decade's worth of work, and resulted in the discovery of the nature of *cis*- and *trans*- gene regulatory elements. Inferences in all steps of the network were aided by perturbations: analogs of the small molecules allolactose and lactose were used to perturb the corresponding nodes in the network and mutants were discovered that perturbed the function of the proteins LacI, LacZ, and LacY. Critically, an *E. coli* strain was used which had a functioning LacI protein, but a mutation in the binding site of LacI near the promoter of *lacZYA*, establishing the existence of *cis*-regulatory elements.

The framework I propose for biological network discovery follows much the same path. To begin network inference, we propose a set of models based on a limited amount of perturbation data. From this set of models, we determine which further effects are most likely to disambiguate our existing models and expand upon the search. The search procedure can then iterate until the network is completely elaborated or the limits of gene-expression in the network prevent further research.

4.2 Network modeling and refinement

Being able to accurately model a biological network is a necessary step towards being able to learn them *de novo*. Gat-Viks *et al.* [28, 26, 27] sought to use interaction and regulatory links found in literature to create a network. This network modeled protein interactions, gene regulation, mRNA quantities, and protein quantities. Accurately modeling a network also requires being able to model any cycles that may appear, so they used a factor graph formulation of the regulatory network. Combined with expression data, they were able to identify regions where the regulatory network was poorly characterized, and then refine it. Gat-Viks *et al.* used several *S. cerevisiae* systems for biological verification, including the osmotic stress response network.

The growing abundance of protein-protein interaction and protein-DNA interaction data, described in §2.3.1 and §2.3.2, is particularly amenable to proper modeling, due to uncertainty of the accuracy of these new methods. In addition, working on the scaffolding of a protein-protein and protein-DNA net permit the exploration of direct causal links in a network. Yeang *et al.* [83] sought to explain knockout gene expression data in conjunction with protein-protein and protein-DNA interactions. Their inference algorithm assigned direction to protein-protein edges, and a sign (activation or inhibition) to protein-protein and protein-DNA edges. Yeang *et al.*, explained the *S. cerevisiae* pheromone response network, among others.

In addition, Yeang *et al.*, proposed a framework for choosing knockdowns to maximally distinguish between competing hypothesis networks [84]. In my aims I propose extensions to this method to not only distinguish between competing explanations of the data, but also to expand upon the existing network.

4.3 Computational predictions

Building upon their experience with Bayesian networks, Friedman *et al.*were the first to build causal networks from gene expression data to explain a pathway [22]. Their methods are somewhat different from the efforts of their successors in that they built their networks entirely from observational gene-expression microarrays. Additionally, they were able to learn causal Bayesian networks by searching over equivalency classes of partially directed acyclic graphs. Friedman *et al.*, were able to predict the *S. cerevisiae* cell cycle network over the Spellman [75] data set, which consists of an unusually large number of microarrays over a time-course on a synchronized culture. Observational studies with such large numbers of microarrays are very rare, limiting the application of this technique.

4.3.1 Protein-protein based predictions

Interpretation of high-throughput protein-protein interaction data is often confounded by two factors: the inaccuracy of the assay, and false positives from proteins that bind but are never expressed such that they can co-localize. Several separate groups have been able to learn accurate networks from noisy protein-protein interaction data by combining it with co-expression data, with increasingly computationally-efficient methods [76, 73, 39]. This methods relies on searching the protein-protein interaction network for regions with high co-expression. It has recently been extended to include not only protein-protein interaction data, but also protein-DNA interaction data and the results from knockdown experiments, allowing the inference of some causal relationships [64].

4.3.2 Perturbed gene-expression predictions

Markowetz *et al.* [60, 61] developed methods for predicting networks in a common and practical experimental setup: gene expression profiling under perturbation of genes known to be involved in the phenotype of interest. Their method starts with genes known to be related due to a common loss-of-function phenotype when each gene is knocked down. Next, microarrays are used to profile expression of the genome under each perturbation. Finally, they build a model based on the nesting of effects under each knockdown.

Since their predictions are based on secondary effects, they can predict networks over biological entities which are difficult to directly assay. For example, Markowetz *et al.* predict a signaling network in *D. melanogaster* which involves no change in gene expression among the signaling genes and is therefore invisible to current high-throughput techniques.

Since I believe their data setup to be the most appropriate, I base my early prediction methods on their model setup. I have since identified extensions to their approach, outlined in Chapters 7 and 10, which I will use to predict networks.

Chapter 5

New web technologies for collaboration and data visualization

Commonly, piecing together all the components of a biological network has been the work of many different biology labs, each finding some of the interactions that take part in the network. In this way, collaboration has allowed for synthesis of these results into a single, well-supported biological network. New high-throughput studies and new network prediction techniques can speed the pace of discovery, but communicating results, particularly of large datasets such as protein-protein interaction, requires a central database.

Biological networks and supporting datasets such as protein-protein and protein-DNA interaction datasets, usually have high connectivity, and many nodes. This often renders textual descriptions of the links via tables and lists useless, and graphical visualizations of the network can easily become cluttered. I believe that the best way to deal with such complexity is to let people interact with networks, moving nodes and watching edges follow, so that the powerful visual cues of motion can assist in determining connectivity. There are a growing number of such network visualization programs, which are beginning to allow new, hierarchical views of the networks [37]. The Internet has traditionally enabled many types of collaboration between investigators, and I believe that developing web technologies will allow particularly effective collaboration for discovery of biological networks.

5.1 Vector graphics on the web

A requirement for being able to interactively visualize networks in a web browser is a vector graphics language. Vector graphics allow the drawing of arbitrary lines and shapes. Image, *e.g.* PNG or JPEG, based systems are unable to redraw lines quickly enough to update the display with edges as nodes are moved around on the display. Fortunately, all major browsers now include support for vector graphics by default. The standard vector graphics language for the web is Scalable Vector Graphics, SVG, and is supported by all major browsers except for Internet Explorer. Internet Explorer supports VML, a simpler vector language that is capable of displaying networks.

In addition to supporting investigation, network visualization programs are also useful for creating figures. There are free editing programs for SVG which also allow export to PostScript or PDF for inclusion as figures in manuscripts.

5.2 Web application paradigms

Recent developments in the web programming community have enabled a new type of web site that is much more interactive and responsive than most web pages. Common support for Dynamic HTML (DHTML) allow web developers to make changes to the current web page without reloading, parsing, and painting the current web page. Also, new web programming patterns allow web pages to use an XMLHttpRequest (XHR) object to transfer small amounts of data to and from a server, without the overhead of a whole page reload. Both of these techniques are boons for a web-based network visualization tool, enabling interactivity that can't be achieved from static web pages.

Part II

Preliminary Results

Chapter 6

Model averaging finds *D. melanogaster* signaling pathway

The availability of RNAi for knockdown and microarrays for profiling gene expression make for a natural combination of measuring many phenotypes under perturbation. Markowetz *et al.* [60] describe a method for inferring a signaling network from such data. They defined a signaling network as a graph with two parts. The genes subject to RNAi knockdowns, referred to as S-genes, are arranged in a directed graph describing the transference of signal from the upstream source to downstream effects. Some genes from the microarrays, referred to as E-genes, were each attached to a single signaling gene. The signaling network describes the response of the E-genes under perturbation.

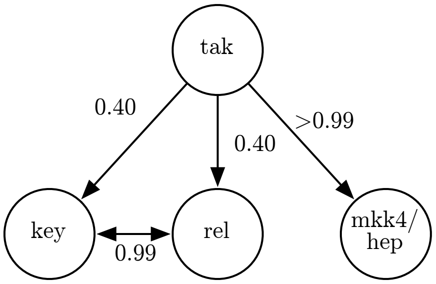
Given such a network, Markowetz's method is able to score the likelihood of the observed microarray data. When the network space is tractably enumerable, then the best network can be found simply by scoring all networks. However, if the number of signaling genes is n then number of possible networks scales as $\mathrm{O}\left(2^{(n^2)}\right)$. Thus, finding the optimal network using exhaustive search is tractable only for a small handful of genes. I wished to use the method to place eight human genes in a pathway, is beyond the current computing power available to me. To approximate the correct answer, I used model averaging.

6.1 Model setup

I evaluated the likelihood of a subset of models and then assigned a posterior probability to each potential link. By "link," I mean either the "forwards" and the "backwards" directions between a pair of genes. For

	key	rel	tak	mkk4hep
key		$-2.5 \cdot 10^{-7}$	-130	-94
rel	$-5.7 \cdot 10^{-7}$		-130	-94
tak	$-9.2 \cdot 10^{-1}$	$-9.2 \cdot 10^{-1}$		$> -1 \cdot 10^{-99}$
mkk4hep	$-7.8 \cdot 10^{1}$	$-7.0 \cdot 10^{1}$	-69	

(a) Log posterior probabilities for all links



(b) Correct links and their estimated posterior probabilities

Figure 6.1: Estimated posterior probabilities of each link. The likelihood of the data for every linear network was calculated, and the posterior probabiliy was calculated with equation 6.1 and a prior link probability of 0.25. In table 6.1(a) the source gene of the link is on the left and the target is along the top. Highlighted links were chosen to be in the network.

example, $A \rightarrow B$ and $B \rightarrow A$ are two separate links, present or absent in the network independently. Denote the likelihood of the data as D, and the subset of possible networks as $\mathcal{M}$. Then I assign the probability of any edge between A and B as approximately:

$$\text{Prob}\left(A \rightarrow B|D\right) \approx \frac{\sum_{M \in \mathcal{M}} \text{Prob}\left(D|M\right) \text{Prob}\left(A \rightarrow B|M\right) \text{Prob}\left(M\right)}{\sum_{M \in \mathcal{M}} \text{Prob}\left(D|M\right) \text{Prob}\left(M\right)} \text{Prob}\left(A \rightarrow B\right) \tag{6.1}$$

I assemble a network from posterior probabilities by choosing a threshold probability, and then taking only those links that pass that threshold. Since there are $\mathrm{O}\left(n^2\right)$ links, each with its own posterior probability, there are at most $\mathrm{O}\left(n^2\right)$ thresholds Each threshold results in a unique network, and I find the most likely network by evaluating the likelihood of each.

6.2 Linear models recover *D. melanogaster* immune response

In order to test model averaging, I evaluated a *D. melanogaster* LPS signaling dataset [7] that had previously been analyzed via exhaustive search [60] successfully. In order to ensure a sample that considers all possible orientations of edges, I used the subset $\mathcal{M}$ of all linear permutations of the signaling genes. This subset, though still super-exponential, *i.e.* $\mathrm{O}\left(2^{n \ln n}\right)$, can still be evaluated up to approximately 10 genes on a desktop computer in less than a day using my current implementation. In order for the computation to be numerically stable, I performed all calculations in log space using the usual identity for addition of log space numbers [13].

Linear model averaging recovers the signaling network found by both the original biological investigation [7] and the subsequent computational investigation [60]. I chose a prior on network features of 0.25, on the assumption that genes are at least minimally connected, *i.e.* that there are at least three of the possible twelve directed edges. The gene *tak* is at the top of the signaling hierarchy, signaling to all other genes. The genes *rel* and *key* are linked equivalently, indicating that their phenotypes under knockdown are indistinguishable. Linear networks without the link *tak* → *mkk4/hep* have data likelihoods so much worse than those with the link that the posterior probability of the link is greater than $1 - 2^{-2048}$

The links from *tak* to *key* and *rel* have the lowest posterior probability of the true links, but are still orders of magnitude stronger than the posterior probabilities of any of the false links. Further, with a prior probability on the links that is based on the true model, 5/12, these links would have a posterior probability of 2/3. One possible reason for the lower probability of these links is due to the equivalence of *key* and *rel* in that they have links in both directions between them. Therefore removing just one of the links from *tak* will still result in an identical signaling network, since the signal is transitive in this model

An advantage of the model averaging approach over exhaustive search is that it provides posteriors on individual network links. Using Markowetz's method assigns a single likelihood to the entire network. Ranking individual features can guide further investigations, and help assess where to trust the most likely network.

6.3 Predicted cancer networks

Attempting to predict cancer networks failed to produce any confident networks. Discretization of the data proved quite challenging, as there are few replicates from which to estimate variance. Without a reliable estimate of variance, placing a discretization boundary at an arbitrary level only emphasized platform effects, and a proper normalization was difficult.

Chapter 7

Reconstructing signed gene networks using bottom-up approximation

Model averaging successfully found the posterior probability of individual network links in the network. This suggests that another method might also work: scoring each pair of genes independently. This would eliminate the need to find a representative sample of networks in the super exponential space, and allow scaling to larger networks than is possible via linear model averaging.

I will continue to use a similar network structure in that each gene subject to RNAi knockdown is a network gene, and the genes which are profiled on microarrays are response genes. However, I will inhibition to the model, in addition to activation. Not only is inhibition an essential part of biological regulation, but I will show that it is more sensitive for learning.

7.1 Method and computational costs

I scores a pairwise link by approximating the score of the best full model that contains that link. For example, with the link $A \to B$ we would hypothetically consider the subset of the model space containing that link, denoted by $M(A \to B)$. We will continue to assume that each response gene, indexed by j, has a single attachment point, θ_j. When considering a single pair this means that the attachment point θ_j, I consider attachment to A, attachment to B, or attachment to neither. For example, the calculation for the score for a link between A and B is roughly:

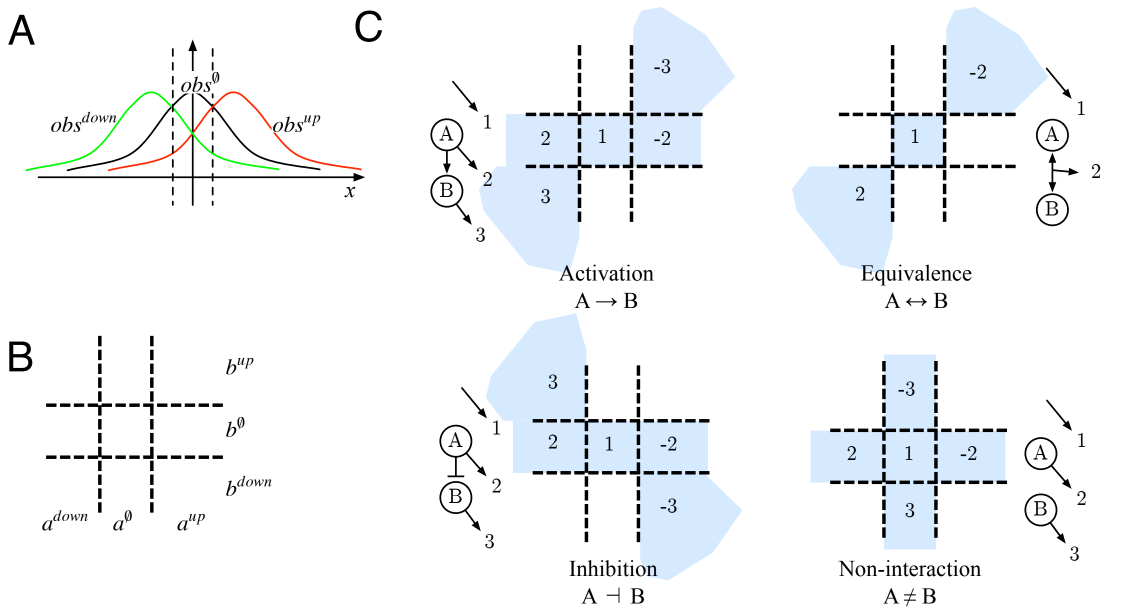


Figure 7.1: Pairwise scoring model. (A) Each expression measurement is modeled to come from one of three overlapping Gaussian distributions, up, down, or no change. (B) When considering the contribution of a single response gene to a pairwise score, the observations under both perturbations a and b are used. Every coordinate has a maximally likely distribution from a and b, the dotted lines indicate these quantization boundaries. (C) Every model of pairwise interaction has different allowed regions depending on where the response gene is attached. Allowed regions are indicated by shading.

$$\begin{aligned} S(A \rightarrow B) &\approx \max_{M \in M(A \rightarrow B)} \text{Prob}\left(D|M\right) \\ &\approx \max_{M \in M(A \rightarrow B)} \prod_{j \in \text{response}} \text{Prob}\left(D_{Aj}, D_{Bj}|M\right) \\ &\approx \prod_{j \in response} \max_{\theta_j} \text{Prob}\left(D_{Aj}, D_{Bj}|\theta_j, A \rightarrow B\right) \text{Prob}\left(\theta_j\right) \end{aligned} \tag{7.1}$$

Each θ_j is optimized individually from the others. In addition to inhibition among the network genes, the attachment from a network gene to a response gene may also be either inhibition or activation.

Each type of interaction will have a different scoring model, based on the Figure 7.1 summarizes the scoring model for interaction types between two networks genes A and B. There are four possible relations activation, inhibition, equivalence, or non-interaction. Figure 7.1 also shows the possible values of θ_j for each interaction model, each shaded blue region corresponds to a different value of θ_j.

I model observations as continuous measurements from a mixture of three Gaussian distributions. Gene expression, as would be observed from the normal state of an activated network, is modeled by the $obs^{\emptyset}$ distribution. Expression that is below or above normal when perturbed is modeled by the obs^{up} and obs^{down} distributions.

The Markowetz [60] likelihood has a direct adaptation under the new model space. For simplicity I assume a single observation of a response gene under each perturbation; the extension to replicates is simple. Let D_{ij} refer to the observation of response gene j, in the range of $1, \ldots, m$, under perturbation of network gene i, in the range of $1, \ldots, n$. Let θ_j refer to the attachment point of response gene j in the network, with a range of $-n, -(n-1), \ldots, -1, 0, 1, \ldots, (n-1), n$, where negative values refer to inhibitory attachment and 0 refers to no attachment in the network. Then the data likelihood given a model is:

$$\mathrm{Prob}\left(D|M\right) \propto \prod_{j=1}^{m} \sum_{i'=-n}^{n} \prod_{i=1}^{n} \mathrm{Prob}\left(D_{ij}|M, \theta_j = i'\right) \tag{7.2}$$

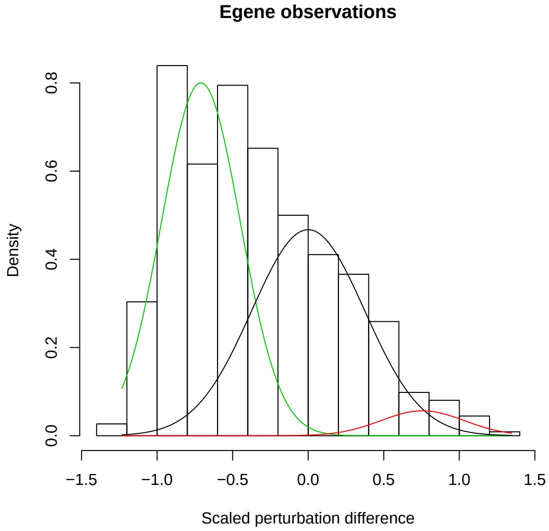
The model M specifies the response of network gene $|i'|$, which when multiplied by the sign of i', selects the appropriate observation Gaussian with -1 mapping to obs^{down}, 0 to $obs^{\emptyset}$, and 1 to obs^{up}. The posterior distribution on attachment points is also easily adapted to the new model space:

$$\mathrm{Prob}\left(\theta_j = i'|D, M\right) \propto \prod_{i=1}^{n} \mathrm{Prob}\left(D_{ik}|M\theta_j = i'\right) \tag{7.3}$$

7.2 *D. melanogaster* pathway recovered with continuous observations

As a positive control for the method I predicted networks on the *D. melanogaster* immune response data set. This data set contains single-channel arrays for unperturbed cells with and without the immune response activated as well as microarrays with the immune response activated, but perturbed. In order to use these single channel perturbation arrays with the new observation model, I converted observations into ratios.

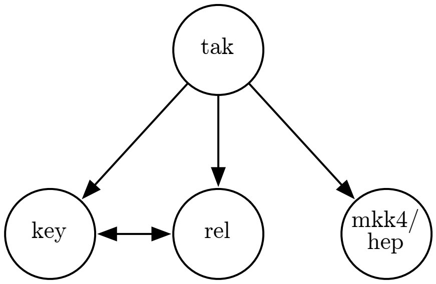
I treated this data as similarly as possible to the discrete case. The data was normalized by first subtracting out each response gene's unperturbed expression value from the expression value under knockdown. Next, this value was scaled by dividing the perturbation difference by the difference between the negative and positive control arrays. This scaling removes allows different genes to use the same observation distributions, even though they may have different response ranges in the cells or microarray probes of different sensitivities.



(a) Observation destributions

A	B	A → B	B → A	A ⊣ B	B⊣A	A ≠ B	A↔B
rel	key	-387.1	-393.8	-495.0	-554.1	-485.4	-375.5
rel	tak	-546.0	-395.7	-592.7	-537.1	-444.0	-541.5
key	tak	-530.5	-364.8	-575.6	-455.7	-399.3	-520.9
rel	mkk4hep	-476.1	-480.8	-471.9	-532.2	-399.0	-555.9
key	mkk4hep	-451.6	-430.6	-455.9	-451.1	-369.4	-510.6
tak	mkk4hep	-331.3	-379.2	-455.8	-600.1	-448.2	-350.0

(b) Log likelihood of link models



(c) Predicted network

Figure 7.2: Pairwise scoring of *D. melanogaster* network.

I used expectation-maximization to estimate the observation distribution as a mixture of Gaussian distributions. These distributions, shown in Figure 7.2(a) along with a histogram of the observation data, have mixture weights roughly similar to the counts of the discretized data calls. I calculated the likelihood of all link models for all link pairs, which are summarized in Table 7.2(b). Finally, I constructed a network by taking the most likely model for each pair of genes. The predicted network, shown in Figure 7.2(c) is an exact match to the true network. The pairwise models of activation, equivalence, and non-interaction are all successfully predicted. This shows that local, pairwise prediction recovers the same links as methods that score the full network.

7.3 Modeling inhibition improves network reconstruction on synthetic data

In order to assess the performance of pairwise scoring, I would like to test the approach on many more datasets in which genes have been knocked down for known signaling networks. Unfortunately, there are very few such networks with appropriate data. To approximate this assessment, I generated random networks and data according to the likelihood model, and then evaluated the predictions from pairwise scoring. Though less ideal than biological data, such investigations show the theoretical limits of the scoring scheme and can be informative to the type of biological data that needs to be collected.

I generated random networks by first building a random tree containing between 5 and 15 genes. Next, I added cycles by creating a random number of links from descendants to ancestors in the tree. The full network was completed by creating a pool of response genes that is twenty times the size of the signaling network, and then randomly attaching each response gene to one signaling gene. I randomly assigned inhibition in the network according to a tunable parameter, which was set to 0.25 of the network for the following results. Random data was generated by simulating a perturbation of each signaling gene and then sampling observations from the appropriate observation distributions. In order to compare unsigned to signed models, I reran the learning procedures after taking the absolute value of all generated samples.

To ascertain appropriate parameters for the observation distribution in the synthetic data, I evaluated expression data from approximately 5000 microarrays in three species. The ribosome and the proteasome perform roughly opposite tasks and when one is up-regulated the other is generally not up-regulated. The essential parameter of the observation distributions is the signal-to-noise-ratio, the mean divided by the standard deviation. For each array, I found the set of genes annotated as ribosome and proteasome by Gene

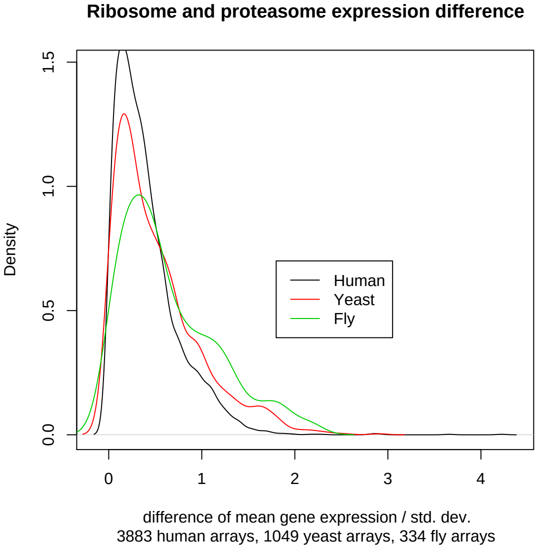
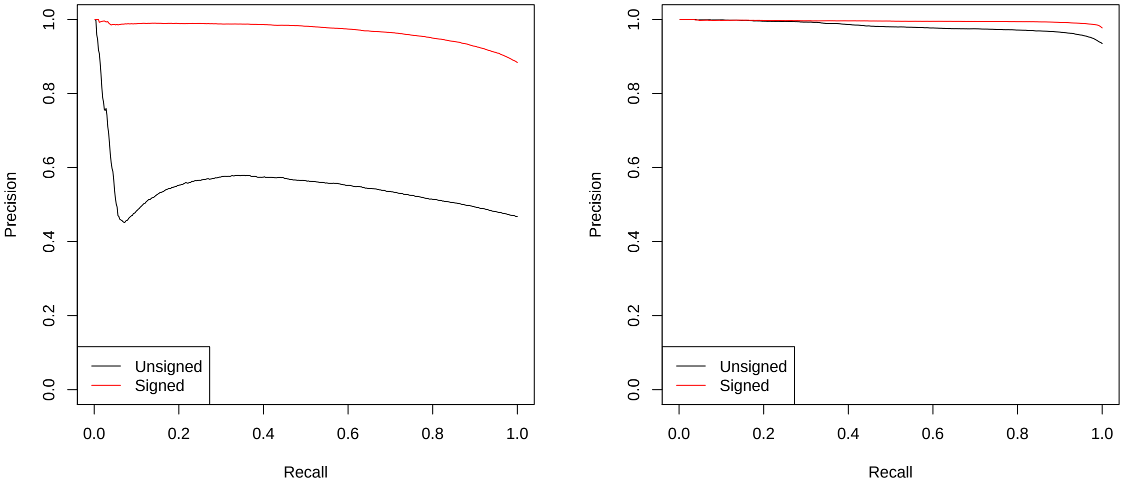


Figure 7.3: Compendium estimate of activated vs. deactivated network expression. The mean difference in expression between the proteasome and ribosome genes across three species of microarray expression data.

Ontology, and then found the mean and standard deviation of each. I then took the difference of the means of the two sets, and divided the difference by the square root of the standard deviation of each set. I plotted the estimated densities of the separation of the two pathways in Figure 7.3. The right tail of Figure 7.3 is the most relevant, as these are the arrays where the proteasome and ribosome are differentially regulated. Based on these plots, I used observation distributions which are separated from 1.5 to 3.0 standard deviations.

I predicted networks on both signed and unsigned data with observation separations from 1.5 to 3.25 in increments of 0.25. For each observation separation I generated 500 networks and predicted networks for both the signed and the unsigned data. I assigned each pairwise prediction a score: the likelihood win over the next most likely pairwise interaction model. Then, within each observation separation I pooled all of predicted pairwise interactions and compared them to the true pairwise interaction

I plotted precision and recall using the likelihood win score in Figure 7.4. At the lower signal-to-noise ratios, the unsigned model has great difficulty making any correct predictions, and the signed model outperforms the unsigned model. As the separation increases, the difference between signed and unsigned predictions lessens. Synthetic data shows that incorporating signed edges and data greatly increases accuracy.



(a) Separation 1.75

(b) Separation 3.0

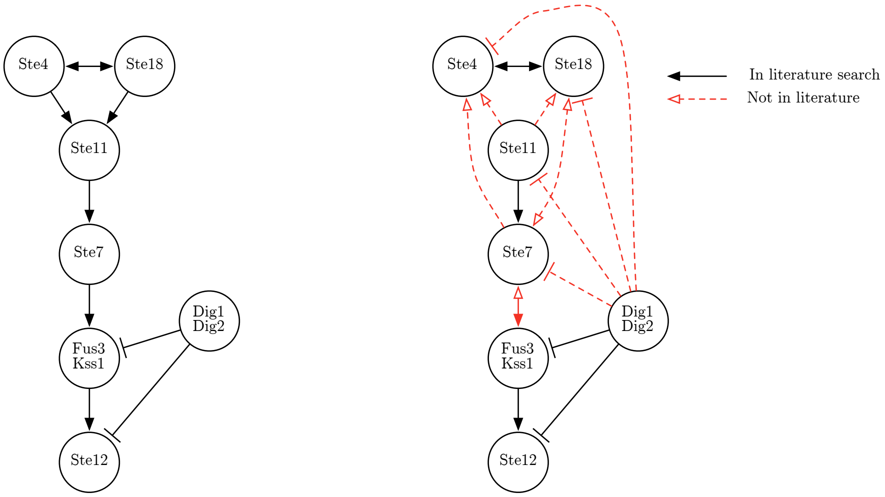
Figure 7.4: Performance on synthetic data, varying the signal-to-noise ratio. Note the different scales on the y-axis between 7.4(a) and 7.4(b).

7.4 Pairwise scoring recovers *S. cerevisiae* inhibitory network

Inhibitory regulation is an essential aspect of biological networks, since regulation in both directions is necessary for control of a process. Here I show that I am able to predict inhibition from biological data. One of the most well known datasets of gene expression under perturbation was published by Hughes, *et al.* [40]. This study performed gene expression profiling of over 260 deletion strains of *S. cerevisiae*, covering many known pathways. Previous studies, using both the Hughes dataset and protein-protein interaction datasets, have been able to recover some of the *S. cerevisiae* pheromone signaling network. I used pairwise scoring to predict the pheromone network using just the expression profiling under perturbation, without protein-protein datasets.

The *S. cerevisiae* pheromone signaling network is a MAP kinase system. Figure 7.5(a) shows the known components of the signaling pathway which are in the Hughes perturbation compendium. Ste4 and Ste18 are components of a G-protein. The proteins Ste11, Ste7, and Fus3/Kss1 are attached to the same scaffold, and are part of the signaling cascade. Ste12 is a transcription factor for mating-specific genes, and Dig1/2 is an inhibitor both of Ste12 and of the Fus3/Kss1 MAP kinase.

Pairwise scoring recovers many aspects of the signaling network without protein-protein interaction data, as shown in Figure 7.5(b). Ste12 is correctly identified as the furthest downstream component. Additionally,



(a) *S. cerevisiae* pheromone network from literature (b) Predictions on *S. cerevisiae* pheromone network

Figure 7.5: *S. cerevisiae* pheromone network

Dig1/2 is found to inhibit both the Ste12 transcription factor and the Fus3/Kss1 kinase.

However, some of the cascade is weakly predicted and incorrect. The G-protein is predicted to be downstream of the kinases, which is incorrect. Additionally, Ste7 is predicted to be in complex with Fus3/Kss1, and though this is not entirely incorrect, it obscures the cascade. Finally, Dig1/2 is found to inhibit all elements of the network, for which there is no support in the literature, but also no contradicting evidence. Though it may be that Dig1/2 inhibits the MAPK kinase and the MAPKK kinase, it seems biologically unlikely that Dig1/2 also inhibits the G protein.

Some of the predictions in this network result in intransitive triples of genes, which is disallowed in the signaling context where this model originates. This intransitivity, among Ste4, Ste7, and Ste18, includes incorrect predictions. These links are also weakly predicted, as other links were nearly as likely. While this is a failing of the pairwise scoring for this model space, I formulate two possible resolutions, explored in Chapter 10 and Chapter11. First, it may by that by taking a suboptimal pairwise model for one pair, the overall network likelihood could be increased. In Chapter 10 I explore a method to resolve the pairwise scoring. Second, it may be that the model space is wrong and that multiplicative regulation is insufficient to explain the data. In Chapter 11 I explore how to unite a different regulation model with pairwise scoring. In Chapter 8 I present preliminary results using this different regulation model.

Chapter 8

Restricted acyclic networks recovers *V. cholerae* biofilm formation

In the previous chapters I presented network predictions from single-gene knockdowns datasets. These networks were all multiplicative, meaning that if a gene has more than one regulator, than knocking down any one of these regulators will perturb the regulatee. However, we know that there exist more complex regulatory motifs in biological networks. If a gene is subject to additive regulation by two regulators, than the presence of either regulator is sufficient to activate the gene. Multiple-gene knockdowns are required to discover additive regulation, and in this chapter I will present models and data capable of detecting additive regulation.

Perturbation has direct consequences in causal probabilistic systems, resulting in causal conclusions that are stronger than those from purely observational experiments. The theoretical framework for causality in acyclic directed graphs has been fairly well established. In this chapter, I discuss the implementation of directed acyclic probabilistic network library, a restricted set of such acyclic networks suitable for biological networks, and learning a known network from perturbation data.

8.1 Java Bayesian Network Library

Though there are many libraries for Bayesian network algorithms and data structures, most have been abandoned, and of those that are still being maintained none have implement factor graphs and variational learning, both of which I anticipate using. I therefore implemented Bayesian Network algorithms and data structures that can be extended to factor graphs and variational algorithms. Currently, the library supports multiplication and summarization of arbitrary factors/potentials, variable elimination, and junction trees. With Pinal Kannabar, I implemented structure learning by hill climbing.

8.2 Learning network models from joint interventions

In collaboration with the Yildiz lab, we have the data from gene expression profiling from strains of *V. cholerae* with deletions of known biofilm regulating genes. They have provided us data for all possible single-gene, double-gene, and triple-gene deletions of *vpsT*, *vpsR*, and *hapR* [8, 32].

We have modeled *V. cholerae* biofilm formation with binary Bayesian networks, with three extensions from the networks as in Pe'er. First, we restricted the conditional probability tables (CPTs) for additive and multiplacitve regulatory logic. Second, we construct a separate network for each perturbation to model the change in network structure that accompanies interventions. Finally, our observations are of comparisons of perturbations, so we introduce variables for observations that span perturbed networks. We call the resulting data structure a joint intervention network.

Regulation is modeled in the CPTs of nodes. Each regulation link is in one of three categories: an additive activator, a multiplicative activator, or a multiplicative inhibitor. If the regulators of gene X in each of these categories are denoted $A^+_\bullet$, $A^\times_\bullet$, and $R_\bullet$ respectively, then the CPT for node X is:

$$\text{Prob}\left(X|\{A^+_\bullet\},\{A^\times_\bullet\},R_\bullet\right)=\begin{cases}\theta_{Parents(X)} & \text{if}\left(\bigvee_i A^+_i\right)\wedge\left(\bigwedge_i A^\times_i\right)\wedge\bigwedge_i\neg R_i\\ 1-\theta_{Parents(X)} & \text{otherwise}\end{cases}\tag{8.1}$$

For each setting of the parent nodes the parameter $\theta_{Parents(X)}$ takes on a different value in the range $(0.5, 1.0)$. If a gene X has no regulators in the network, then its CPT is uniform over the two states.

In causal networks, perturbing a variable removes other causes. We model this with a separate network for each perturbation, with the appropriate edges removed from the graph. The CPTs that are unaffected by perturbation share parameters between the various perturbed networks.

Our observations are from gene expression microarrays that compare the state of two networks, so we construct observation nodes comparing the state of two different networks. The observation variables are ternary, corresponding to expression change up, down, or no change. Alternatively, we could use continuous observations with a mixture of Gaussian distributions, as in §7.1. In the discrete case that we are using, the CPTs are deterministic. We discretize our microarray expression using a cutoff of ± 0.3 For the comparison

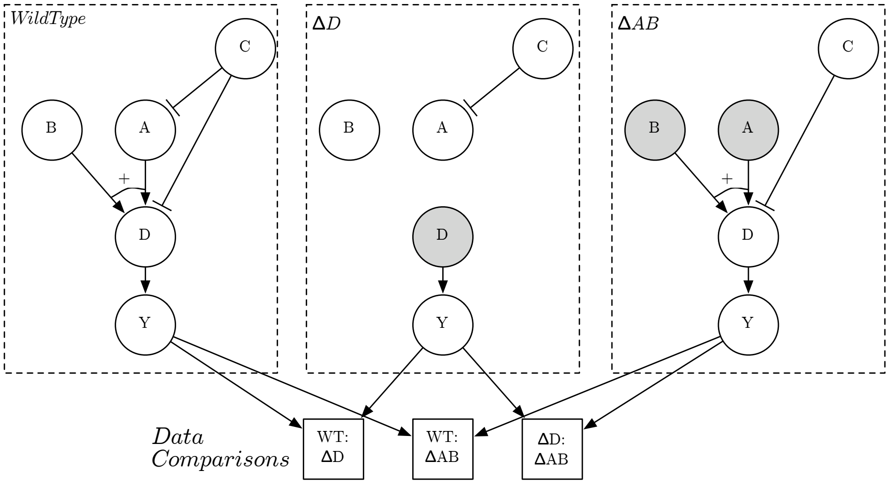


Figure 8.1: Example joint intervention network.

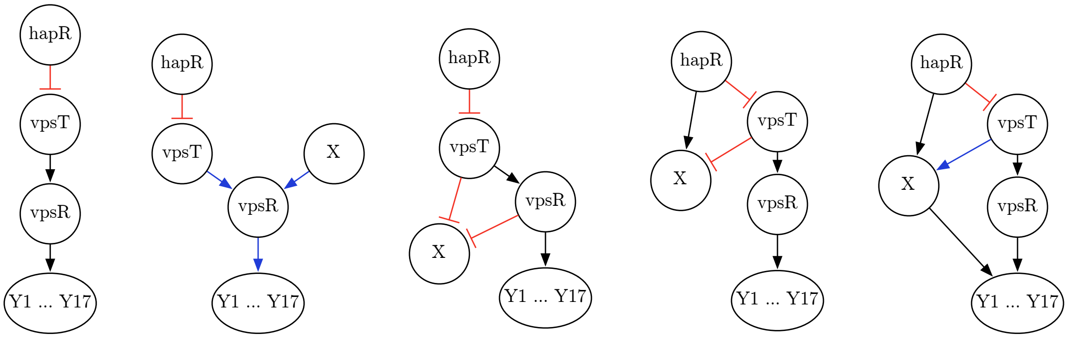
of response gene Y in perturbations 1 and 2, labeled D_{12}, the CPT would be:

			D_{12}	
Y_1	Y_2	up	no change	down
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

(8.2)

In addition to the direct observation of every mutant strain versus wild type, we construct synthetic observations of all pairs of mutants by subtracting out the difference. This emphasizes the difference between, for example, every double knockout and the single knockout, in order to help distinguish any additive regulation.

Figure 8.1 shows an example of the data structure used for learning a network from joint intervention data.. In that network, there is a single deletion strain ΔD, a double deletion strain ΔAB, and a wild type strain. Perturbed nodes are shaded, and any parents are disconnected. There are three observation nodes, for every possible comparison between perturbations.



(a) Expected network

(b) Highest scoring networks

Figure 8.2: Predicted *V. cholerae* biofilm networks.

8.3 Structure Learning in *V. cholerae* Using Known Effect Genes

Our data consisted of a fairly small set of genes: *hapR*, *vpsT*, and *vpsR*. All possible single- and multiple-knockdowns were performed, and microarrays measured gene expression under each knockdown vs. the wildtype strain. Due to previous studies, 17 genes involved in biofilm formation are already known, and can be used as response genes. We label these 17 genes collectively as Y in the following figures.

Since there were a small number of regulatory genes, we used a hill climbing strategy as the search space was likely small enough. We searched over network structures, where a structure is defined as a particular set of edges and the assignment of each edge to a regulation category, repressor, additive activator, or multiplicative activator. To score a network structure, we generate 200 networks with CPTs according to equation 8.1, and θs selected uniformly from 0.5 to 1.0. We then evaluated the likelihood of the data according to the joint intervention network, and chose the best likelihood from the sampling of 200 CPTs as the representative likelihood for the entire data structure. From a random starting network, we tried all possible new networks structures that would result from adding an edge, deleting an edge, or changing the class of an edge.

We also added a hidden variable, X, for two reasons. First, if our assumptions on possible regulation is poor for a gene, X can provide some slack in the CPTs. If placing X in a particular position is strongly favorable in the likelihood, then the hidden variable may explain hidden dependencies in the data. In this case, X may help us locate where to look for another regulator to add to the network.

Figure 8.2 summarizes the predicted networks. All four of the highest scoring networks are compatible with the manually inferred regulation network. The hidden variable, X, was not confidently placed at any

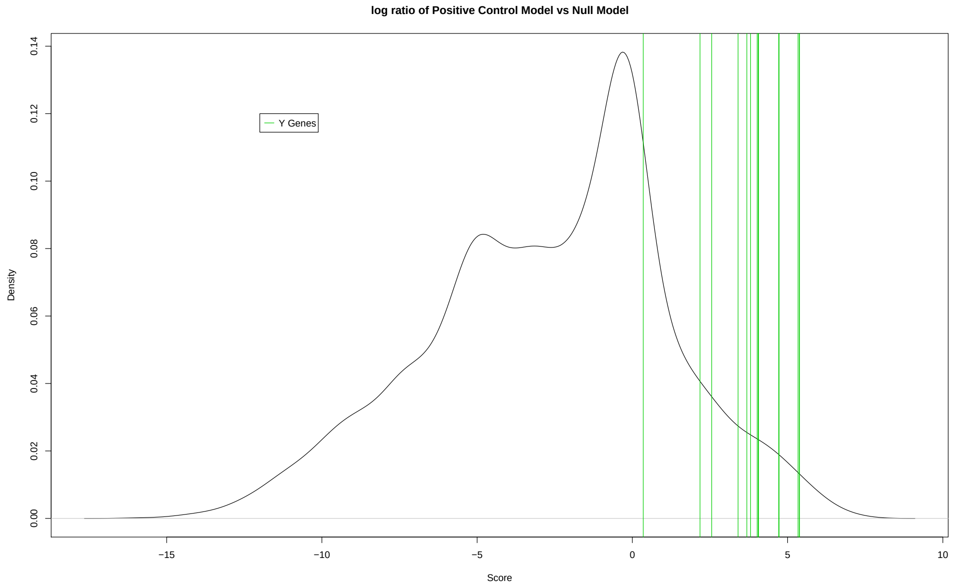


Figure 8.3: Log odds of *V. cholerae* genes to be under the biofilm regulation program.

particular place in the network, and only in the third best model does it have any outgoing influence.

8.4 Predicting new biofilm genes

Given a network model that accurately reflects the regulation program, the rest of the genome can be scored for compatibility. It is simple to calculate every gene's likelihood under the model. However, certain gene profiles score well under all models, such as a flat gene profile that remains at 0 under all perturbations. To correct for this, we evaluate the likelihood under the true model divided by the likelihood of a null model. We use the fully unconnected network model as the null model. This corrects for biases towards genes profiles that fall entirely in the "no change" range.

Figure 8.3 shows a density estimate of the log odds of every gene expression profile in the *V. cholerae* dataset. The green lines indicate the log odds of the known biofilm genes. Among the top-ranking predictions, many are known to participate in the biofilm that were not in the original list 17 genes. Table 8.1 summarizes the top new predictions.

Score	Accession	Annotation
7.192	VC1888	hemolysin-related protein
7.024	VCA0864	methyl-accepting chemotaxis protein
6.936	VC0483	hypothetical protein
6.859	VC2445	general secretion pathway protein A
6.527	VC2730	general secretion pathway protein G
6.464	VC2732	general secretion pathway protein E
6.335	VCA0811	chitinase, putative
6.329	VC1320	DNA-binding response regulator
6.272	VC1195	lipoprotein, putative
6.258	VCA0657	aerobic glycerol-3-phosphate dehydrogenase [EC:1.1.99.5]
6.187	VC1064	lipoprotein-related protein
6.124	VCA0570	Sui1 family protein
6.023	VC2483	acetolactate synthase large subunit [EC:2.2.1.6]
6.016	VC2529	RNA polymerase sigma-54 factor
6.016	VC2731	general secretion pathway protein F
5.995	VC0933	hypothetical protein

Table 8.1: New biofilm gene predictions

Chapter 9

Emerging web technologies allow network visualization

Due to their complex graphical nature visualizing biological networks can be difficult. Quite often the density of edges in a graph makes it nearly impossible to see which nodes connect to what other nodes, resulting in difficulty interpreting the graph. Being able to manipulate the graph, by moving nodes and seeing the corresponding movement of edges, can be a great help in understanding the graph. Thus, interactive graphs are a near necessity for interpreting networks of even moderate size.

New web technologies are allowing such visualizations in a web browser without the installation of any software. This allows the creation of a public compendium of many data sources which can be examined without any startup time. We have begun to implement a web-based network visualization, to assist me in finding supporting data for my predicted networks, named the Interaction Browser.

9.1 Interactive Graph Visualization

Though little used, all current browsers include vector graphics support. Since the vector graphics are drawn entirely by the browser, it is possible to display graphs that the user can manipulate, without having to send bitmaps over the network. Through the use of the Document Object Model and the XMLHTTPRequest object, it is also possible to add new information to the page without losing the current configuration. This allows a user to arrange a network in a convenient way, and then add additional nodes without losing familiarity with the network. I have implemented such a network visualization tool, which can dynamically add and remove nodes, and lay them out on the screen. New gene nodes can be searched for by name or annotation.

The user interface currently has a large network view area in the middle. On the left is a bar for controlling the view: adding and removing nodes and interaction data. On the right is an information panel for viewing detailed annotation and information.

9.2 Multiple Simultaneous Interaction Data Sources

I have also added several interaction sources to the browser, referred to as "tracks" in reference to the UCSC Genome Browser. A track can be any pairwise data, *i.e.* any set of edges. Currently, we have protein-protein interaction data, protein-DNA interaction data, co-expression data, and co-phenotype data. Each track is color coded, and when multiple tracks have data for a gene pair, the edges are stacked next to each other.

Part III

Research Design and Methods

Chapter 10

Aim 1: Extend single-gene perturbation models for colon cancer invasiveness

I began my network predictions with the human colon cancer invasiveness pathway, and am continuing my attempts to successfully predict a confident network. In addition to its medical interest, the dataset has several attributes which make it a difficult but desirable target for a network prediction method. First, the data is of single gene perturbations, which are the most easily performed and therefore the most common case. Second, the data set is in a eukaryotic system, which has a large repertoire of regulation mechanisms, all of which a network prediction method should be able to handle. Also, this dataset is spread over two different array platforms, and a method able to unify predictions across platforms will enable broad discovery using existing perturbation data. Finally, this dataset consists of impulse perturbations via RNAi, rather than steady state gene deletions, which will be a common case for eukaryotic network perturbations.

Predicting networks on this dataset aims to not only understand the genes currently known to be involved with invasiveness, but also aims to help predict new cancer invasiveness genes. Previously, genes have been predicted by using simple correlation measures. I hypothesize that identifying accurate computational network models of genes known to be involved with invasiveness will allow better predictions of new genes involved in the process. Knocking down these new genes may speed the rate of successful discovery by focusing resources on more informative perturbations.

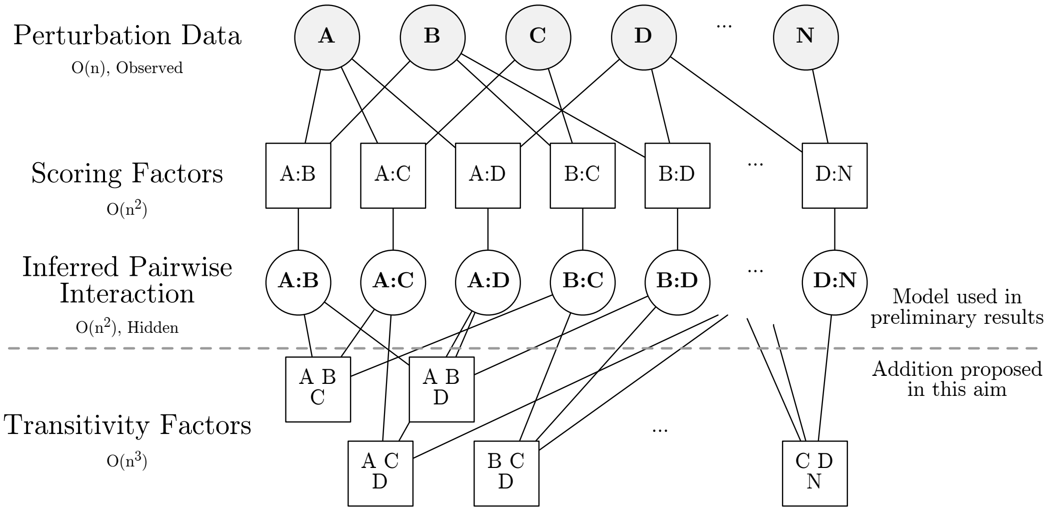


Figure 10.1: Transitivity factor graph

10.1 Predict colon cancer networks

My first attempt at predicting a network in colon cancer, model averaging, failed due difficulties with the data model and also due to the differences between the platforms. In my preliminary results, I came up with methods that are better able to handle noisy data. I will apply the pairwise scoring methods from Chapter 7 to the colon cancer data. If, as evaluated by the methods in §10.2, pairwise scoring does not work, I outline several extensions in the following subsections to try to overcome any difficulties that I encounter. Should all those methods fail, I will conclude that there is not enough signal in the data, and attempt to get more replicates of the microarrays, or attempt an entirely different dataset.

10.1.1 Construct transitivity factor graph

In §7.4 I applied a pairwise scoring method to construct a full network by simply taking the local maximum likelihood interaction for each pair of genes. Note that transitivity was not enforced, and therefore pairwise scores are not guaranteed to be consistent with one another. My predictions in the *S. cerevisiae* pheromone response pathway in §7.4 contained many correctly predicted links. In addition, there were intransitive, incorrect links with poorer scores. This suggests that looking for transitive consistency when choosing a local interaction may result in better predictions where the maximum likelihood is weak.

I propose to construct a factor graph to resolve these inconsistencies. Figure 10.1 shows an example formulation. The previous pairwise method consisted of the first three rows of the figure. The inferred pairwise interactions are discrete variables with a domain of activation forward and backward, inhibition forward and backward, non-interaction, and equivalence. Represented symbolically, the domain is $\mathcal{I} = \{\rightarrow, \leftarrow, \dashv, \vdash, \neq, \leftrightarrow\}$, respectively. Each scoring factor is the functional representation of equation 7.1. That is, it's a function with a domain of $\mathcal{I} \times \Re^2$ and a range of $\Re$.

The new elements to the method are transitivity factors which are shown as the bottom level of Figure 10.1. These transitivity factors are functions that have value 0 when the three pairwise choices are transitively consistent, and have value $-\infty$ when an edge triple is intransitive. For example, if the values of the variable nodes A:B, B:C, and A:C in Figure 10.1 are $\rightarrow$, $\rightarrow$, and $\neq$ respectively, then the factor has value $-\infty$ since A:B $= \rightarrow$ and B:C $= \rightarrow$ imply that A:C $= \rightarrow$.

Standard message passing can be used to solve the factor graph for the global maximum likelihood assignment to all pairwise interactions variables. Since all the factors are completely specified, there are no parameters to learn. My initial attempts at message passing schedules will alternate between passing all possible messages to factor nodes and then passing all possible messages to variable nodes. I will perform message damping to prevent cycles. To determine convergence, I will track the change in the likelihoods of the pairwise interactions after every cycle of message passing, terminating when 10 consecutive cycles have a norm difference of less than 1%.

10.1.2 Choosing informative response genes

Choice of response genes can significantly influence model induction. For example, if the data is very noisy, and all genes are used as response genes, then noise from observations unrelated to cancer invasiveness could confound true changes in gene response. By limiting the data to genes that are more likely to participate in the invasive phenotype, the network predictions may become more robust. I will attempt to improve network predictions by filtering response genes with co-expression data, protein-protein interaction data, or both.

When determining the first round of genes that were essential to invasiveness, gene expression was profiled for many colon cancer cell tissues. The cell lines included both invasive and non-invasive cultures. Our collaborators previously looked for genes that had higher expression in the invasive cultures and lower expression in the non-invasive cultures, in order to choose genes that could be knocked down and disrupt invasiveness. I propose to use this data set to find genes that are likely involved with invasiveness. Since I am not planning to know these gene downs, but just use their responses to predict a network, I can use genes that either correlate or anti-correlate with the phenotype.

In addition to gene expression, I will limit response gene selection by looking for genes that are connected to invasiveness genes by protein-protein interaction and protein-DNA interaction data. This will narrow my response gene selection to genes for which we already have data for physical connection. Since responses may be several links away from direct connection, I will look for genes that are linked by more than a single edge, but no more than a cutoff of n genes away. I will estimate n by finding the median number of protein-protein or protein-DNA interactions necessary in my data sets in order to connect directly interacting genes from the Reactome pathway database.

10.2 Evaluate predicted networks

Possibly the best evaluation is to test links or newly predicted signaling genes is to use wet-lab techniques. However, wet-lab tests can be expensive and time-consuming. Therefore I propose the following methods as a way to increase confidence in the network predictions before wet-lab experimental evaluation needs to be conducted. I will assess the computational predictions' robustness to added noise, and the significance of the predictions under permutation. I will also use known biological information: the tiers of the network genes and plausibility of connections from high-throughput interaction data. These are discussed in more detail below.

10.2.1 Assess robustness

Given the difficulty of choosing significantly perturbed response genes with the model averaging method, determining robustness to noise in the response gene expression data is a promising starting point for evaluation. I will repeatedly add Gaussian noise to the response gene expression data of a particular standard deviation and make predictions of the network. I will define an edge as robust at a particular noise strength if it is predicted in 95% of the trials. I expect no network predictions to be significant with Gaussian noise with a standard deviation equal to the signal-to-noise ratio of my estimated observation distributions. However, noise with a standard deviation of 0 will predict all edges as robust. I will perform a binary search to assess the robustness of each predicted interaction pair in the network. I will consider the prediction method robust if at least a quarter of the network's interactions are robust with noise equal to 25% of the observation distribution signal-to-noise ratio.

To see if the transitivity constraints are helping robustness, I will perform the robustness analysis both on the raw pair scores and those that result from the transitivity message passing. I will plot robustness of an edge against the score of the edge both for the raw pairwise scores and the transitively resolved scores.

If the transitively resolved scores are a better predictor of robustness, then the slope of regression should be higher in that plot.

10.2.2 Assess significance

To assess significance of the predictions, I will predict networks on data which has had observations permuted within each set of knockdown responses, *i.e.* the data within a single node of the top row of Figure 10.1 will be scrambled. This will allow me to establish a background distribution for every pairwise interaction by taking the likelihood of the best pairwise model from each run on permuted data. I will assign a p-value to the prediction learned from the original dataset by establishing its location in this empirical distribution.

10.2.3 Compare topology to known ordering

In the colon cancer dataset, the order of discovery of the perturbed genes provides some prior information about likely network structures. The first tier of three genes were discovered to be essential to the invasive phenotype in a first round of knockdowns. Based on the gene expression profiles of the these genes, the second set of genes were chosen for knockdown, and found to all be essential for invasiveness. Thus, we expect to see some signs of this tiered discovery in the resulting network. In particular, there should be directed links from at least some of the first tier genes to the second tier genes, but not from the second tier to the first. If this is not the case, then I must either reject the prediction method or conclude that the patterns in the data are too weak.

However, we do not necessarily expect that all the first tier genes should be above the second tier genes. Feedback regulation is likely, in which case we could see some directed links from the second tier to the first tier.

10.2.4 Compare topology to high-throughput interaction data

I will evaluate predicted links by looking for support in high-throughput data. There are many databases of protein-protein interaction data, genetic interaction data, and ChIP-chip experiments. If a predicted link is supported by a chain of links in these other datasets, it provides additional support for the signaling relationship predicted by our model since, for example, the signal colud be carried via a series of known physical events. To see that such a chain is significant, I will first compute the all-pairs distance from the supporting data, and show that the chain for a particular predicted interaction is smaller than a chains between arbitrarily chosen genes. I will use the interaction browser, explored further in Chapter 12, for theses investigations.

10.3 Recommending new knockdowns

An important goal of my proposal is to discover a method for directing future investigations. With this cancer dataset, template matching has been used to discover both the first tier and the second tier of genes involved in invasiveness. I will discuss how template matching was used in the past, and how it can be used again, and then discuss a new method which I hope will provide more informative perturbations.

10.3.1 Template matching

Template matching [65] is a straightforward process, where the expression profile of a gene in compared to a template profile, and then assigned a p-value based on a t-test. The genes with the highest correlation to this template profile are then selected as the most likely to be involved in the phenotype. For the cancer invasiveness phenotype, the template expression profile was 1 in the perturbations that were invasive, and 0 in the perturbations that were not invasive. Pearson correlation of each response gene to this profile will rank them according to template matching.

10.3.2 Uncertainty reduction

As an alternative way to choose new knockdowns, I propose an information theoretic method inspired by Yeang *et al.* [84]. The idea is that there is uncertainty in our belief about the network, and we wish to choose the knockdown which will decrease this uncertainty the most. So far the factor graph in Figure 10.1 was used to determine the most likely network, but it also defines a distribution over all possible networks, $\mathrm{Prob}\left(M|D\right)$, and therefore three is entropy over the model space. Upon knocking down response gene j and observing data $\hat{X}$ the reduction in entropy is $\mathrm{H}\left(M|D\right) - \mathrm{H}\left(M|D, X(\Delta j) = \hat{X}\right)$. Though we can not know what data we will observe, our best estimate is the expected value. Our goal is to choose the j that maximizes the following expression:

$$\begin{aligned}\mathrm{H}\left(M|D\right) - \sum_{X'} \mathrm{Prob}\left(X(\Delta j) = X'\right) \mathrm{H}\left(M|D, X(\Delta j) = X'\right) &\equiv \mathrm{H}\left(M|D\right) - \mathrm{H}\left(M|D, X(\Delta j)\right) \\ &\equiv \mathrm{I}\left(M; X(\Delta j)|D\right)\end{aligned}$$

$$\equiv \mathrm{H}\left(X(\Delta j)|D\right) - \mathrm{H}\left(X(\Delta j)|D, M\right) \qquad (10.1)$$

In order to calculate this quantity, I must first define a model for predicting the response under a new knockdown. Given a particular model, we consider the possibility that one of the response genes is "on" a link connecting two network genes. n Being "on a link means that perturbing that gene would have the same effect as cutting that link, *i.e.* the removal of other causes. Figure 10.2 visually shows the meaning of "on the path."

Let β_j refer to the link in M which response gene j intervenes. In addition to placement on a link in the network, we add the possibility that response gene j does not lie on any path in the network, which I denote by $\downarrow$. Recall that for gene j, θ_j is the predicted upstream network gene, of which there can be only one. β_j is therefore related to θ_j in that if θ_j is known to be a particular network gene i, then β_j can only be one of the outgoing edges from i in the model, or the null value. The likelihood of the data given a model M and a value for β_j is the same as that for θ_j, *i.e.* the observation distribution likelihood given in §7.1:

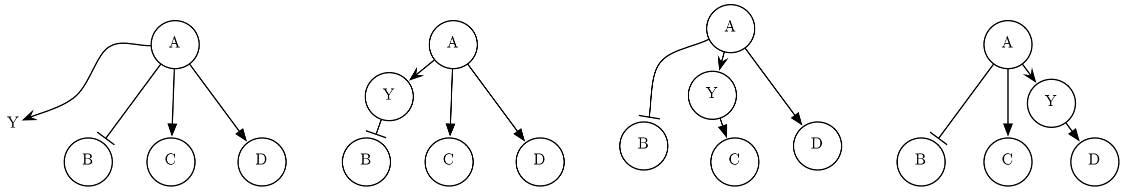
$$\mathrm{Prob}\left(\beta_j = e|D, M\right) = \frac{\mathrm{Prob}\left(D|M, \beta_j = e\right)\mathrm{Prob}\left(\beta_j = e|M\right)}{\sum_{e' \in M \cup \downarrow} \mathrm{Prob}\left(D|M, \beta_j = e'\right)\mathrm{Prob}\left(\beta_j = e'\right)} \qquad (10.2)$$

However, β_j can range over more values, so we must give it different prior and posterior distributions I will use a uniform prior on β_j over all model edges and $\downarrow$.

Given a model M and the location β_j, the predicted responses from knocking down gene j, $X(\Delta j)$, are independent of each other:

$$\mathrm{Prob}\left(X(\Delta j)|M, D, \beta_j\right) = \prod_{j' \in \text{response}} \mathrm{Prob}\left(x_{j'}(\Delta j)|M, D, \beta_j = e\right) \qquad (10.3)$$

I will use the same observation distributions for predicted responses as I have for the observed data, a mixture of Gaussian distributions. The response of single gene j', under the perturbation j, can be calculated



(a) $\beta_Y = \downarrow$ (b) $\beta_Y = A \dashv B$ (c) $\beta_Y = A \rightarrow C$ (d) $\beta_Y = A \rightarrow D$

Figure 10.2: Example of possible values for β_Y if θ_Y is fixed to A.

by summing out the possible attachment points for j', $\theta_{j'}$. Recall that $\theta_{j'}$ can range from $-n$ to n, where n is the number of network genes, negative values refer to inhibitive attachment, and 0 refers to being disconnected from the network. A single response is then calculated by:

$$\begin{aligned}\mathrm{Prob}\left(x_{j'}(\Delta j)|M, D, \beta_j = e\right) &= \sum_{i'=-n}^{n} \mathrm{Prob}\left(x_{j'}(\Delta j)|M, D, \beta_j = e, \theta_{j'} = i'\right) \mathrm{Prob}\left(\theta_{j'} = i'|M, D\right) \\ &= \sum_{i'=-n}^{n} \mathrm{Prob}\left(x_{j'}(\Delta j)|M, \beta_j = e, \theta_{j'} = i'\right) \mathrm{Prob}\left(\theta_{j'} = i'|M, D\right) \qquad (10.4)\end{aligned}$$

The conditions M, $\beta_j = e$, and $\theta_{j'} = i'$ select which Gaussian distribution, up, down, or no change, the predicted response will come from. This results in the predicted distribution being another mixture of Gaussing distributions, but with different mixture coefficients

I have now described the predicted response in a particular model with a specified knockdown. I will now describe three approximations so that the expression 10.1 can be calculated for each potential knockdown. First, I will describe an approximation for $\mathrm{Prob}\left(M|D\right)$, and then approximations for each of the entropies in expression 10.1.

Though the full model space is intractably large, I will sample the most likely models from the factor graph, using a particle-based method. This will result in the most likely network models, $M_1, M_2, \ldots, M_l$, which will account for most of the posterior probability. I will re-evaluate their likelihood with equation 7.2, which scores an entire network, rather than doing all pairwise interactions and double counting the data.. Assuming a uniform prior over these models and using Bayes rule we obtain an approximation on the posterior model space:

$$\mathrm{Prob}\left(M_i|D\right) \approx \frac{\mathrm{Prob}\left(D|M_i\right) \mathrm{Prob}\left(M_i\right)}{\sum_j^l \mathrm{Prob}\left(D|M_j\right) \mathrm{Prob}\left(M_j\right)} \qquad (10.5)$$

I will use the same argument for both terms of expression. 10.1. Because we assume that the each response gene is conditionally independent of each other, we will approximate the total entropy by summing over the entropies of individual response genes, $\mathrm{H}\left(x_{j'}(\Delta j)|D\right)$ and $\mathrm{H}\left(x_{j'}(\Delta j)|D, M\right)$ respectively. We calculate this marginalized probability $\mathrm{Prob}\left(x_{j'}(\Delta j)|D\right)$ for the first term of equation 10.1 by summing over both the model ensemble and possible locations of the candidate perturbation:

$$\mathrm{Prob}\left(x_{j'}(\Delta j)|D\right) = \sum_{k=1}^{l} \mathrm{Prob}\left(M_k|D\right) \sum_{e \in M_k \cup \downarrow} \mathrm{Prob}\left(\beta_j = e|M_k, D\right) \mathrm{Prob}\left(x_{j'}(\Delta j)|M_k, D, \beta_j = e\right) \tag{10.6}$$

This is a mixture of the mixture of Gaussian distributions that resulted from equation 10.4. I can use standard techniques for approximating the entropy of this mixture of Gaussian distributions.

The approximation components of the second term of equation 10.1, $\mathrm{H}\left(x_{j'}(\Delta j)|D, M\right)$ are by definition the expected entropy $\mathrm{H}\left(x_{j'}(\Delta j)|D, M_k\right)$ weighted by the posterior probability of each model in the ensemble. The probability $\mathrm{Prob}\left(x_i(\Delta g)|D, M_k\right)$ is also a mixture of Gaussian distributions and can be calculated by multiplying equation 10.4 by the posterior probabilities of the placement of the candidate perturbation.

10.4 Evaluate perturbation recommendations

I will evaluate the perturbation recommendations by performing the top ranking perturbations from both methods via collaboration. Additionally, a few of the least recommended perturbations will be performed, as a negative control. The final criterion will be which method finds more genes that are essential to the invasiveness phenotype.

Chapter 11

Aim 2: Extend methods for multiple-gene perturbations in *V. cholerae* biofilm formation

Multiple-gene knockdowns permit the discovery of broader regulatory functions than do single-gene knockdowns. The simplest example is the additive regulation of a gene by two activators. If the presence of either regulator is sufficient for activation, than a single-gene knockdown of either regulator will be insufficient to cause a response in the regulated gene. However, a double-gene perturbation of both regulators would discover the links between the three genes.

In Chapter 8 I predicted a biofilm network in a dataset with multiple-gene perturbations. I aim to extend the methods in Chapter 8 to permit discovery of larger networks, allowing for both additive and multiplicative regulation programs. I then aim to use this network to recommend new gene perturbations which will enable further network discovery.

11.1 Predict a *V. cholerae* biofilm network with *cdgA*

We have developed models to successfully predict a *V. cholerae* network. Though not shown in the preliminary results, we applied the same methods to attempted to build a network with additional data from *cdgA* deletion strains. The most likely networks all had very similar likelihoods, and *cdgA* was placed in a different

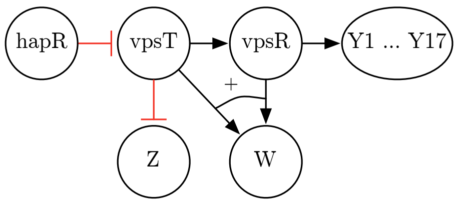


Figure 11.1: Network model for looking for genes that are inhibited by *vpsT* (node Z), or additively activated by *vpsT* and *vpsR* (node W).

location in each.

We will expand upon the previous methods to in two ways First, we will use more response in addition to the 17 known biofilm genes. Second, we will start the structure from better initial conditions, allowing for faster search.

11.1.1 Using more response genes to disambiguate networks

Our previous methods used 17 genes that were known to be the downstream regulatees of the biofilm regulators. There were originally ≈ 4000 genes that were expression profiled. Clustering analysis of the perturbation data found that all 17 biofilm genes were placed in a single cluster. The clustering also found several more clusters that are differentially regulated under perturbation. As network size grows, so does the number of models that will equivalently explain the data, if we look at only the furthest downstream effects. By using only one cluster for network prediction we may be limiting network discovery. In this section I will describe a way to incorporate more clusters into the network prediction.

In §8.4 we found new genes involved in biofilm response by using the predicted biofilm network to rank all genes on the microarray. This same technique can be used to find genes that fit other locations in the network. Suppose we wish to find genes that are repressed by *vpsT*. We place a new node Z in the network, as in Figure 11.1. Then, as in §8.4 we evaluate the likelihood of every gene on the microarray as compared to a disconnected null model. This will provide a ranking over all genes for a particular regulation program by the network.

I will map each cluster to its most likely regulation program in the predicted network. First, I will calculate the centroid of each cluster. For each cluster I will construct a node, similar to the Y node, using the 5% of the cluster that has the highest Pearson correlation to the cluster centroid. I will then attempt to place each cluster in the network independently, using hill climbing as before.

I expect that some clusters will be regulated by the biofilm network, and that some will not be regulated by the biofilm network. For example, a "junk" cluster of genes that do not respond to any of the perturbations will not be informative, and will be likely under the null model of a disconnected network. Conversely, we already know that one cluster, the one that contains the 17 known biofilm genes, is regulated by the network.

To distinguish between informative and non-informative clusters, I will use each cluster's regulatory program to rank all genes in the genome, as in §8.4. For one cluster I will first rank every gene on the microarray using the most likely attachment found above. As before, the ranking will use the ratio of the likelihood under the biofilm network to the likelihood under the null model. With this ranking I will perform a Wilcoxon rank-sum test on all the genes in the cluster versus all the genes outside the cluster. I will keep the clusters that have a significant p-value according to this test.

These observation nodes will then be used in future network searches, along with the previously known biofilm genes.

11.1.2 Seed structure search with single-gene knockdown pairwise scoring

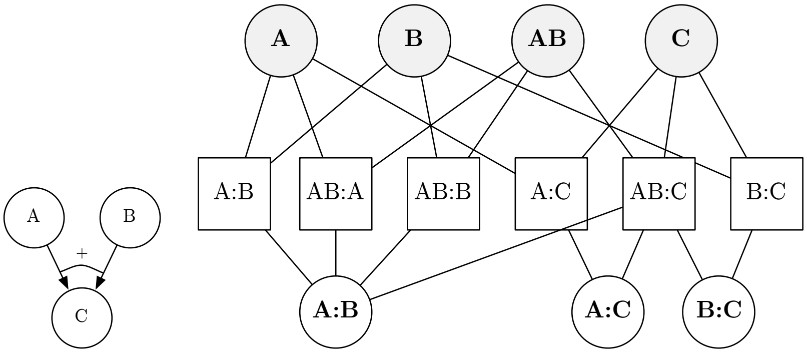
One difficulty of structure search over larger networks is that it can be difficult to obtain a good starting point. Previously, our hill-climbing structure search would make a mere 4-10 modifications to the network before finding a local maximum. The hill climb had to be restarted hundreds of times in order to find the top scoring networks. A good initial network can greatly reduce the search time.

The network learning methods in §10.1 can predict multiplicative networks, and will therefore be able to capture these parts of more complex networks. I will construct a factor graph over just the single-gene perturbations in the *V. cholerae* dataset, and learn the hidden pairwise interactions. As in §10.3.2 I will sample an ensemble of the most likely networks, and use each as a starting point for a hill-climb search.

11.1.3 Seed structure search from a factor graph with additive and multiplicative regulation

If other methods fail, I will adapt pairwise scoring to the multiple-gene perturbation data. One assumption of the pairwise scoring method is that response genes will be connected at most one location in the network. The results of 11.1.1 will indicate the validity of this assumption. If there is a cluster of response genes is already known to be under complex regulation, I will exclude this cluster. Otherwise, I will use all the genes from the above analysis for the pairwise scoring.

Figure 11.2 shows an example factor graph for pairwise scoring with a single double knockout. Pairwise interactions now contain two more possible additive activation interactions: $\rightarrow^{+}$ and $^{+}\leftarrow$. The full domain



(a) Additive regulation

(b) Exapmle factor graph

Figure 11.2: Example of a small factor graph for multiple-gene perturbations with both multiplicative and additive regulation.

of the interaction variables is now $\mathcal{I} = \{\rightarrow, \leftarrow, \dashv, \vdash, \neq, \leftrightarrow \rightarrow^{+}, ^{+}\leftarrow\}$ The previous scoring factors for single perturbation comparisons will remain unchanged. The score for the additive activation model is identical to the score for non-interacting model, as both models result in the same observations.

When including data from double-gene perturbations, there are two new types of factors: double perturbation vs. unrelated single perturbation, and double perturbation vs. contained single perturbation. I will additionally have to consider double-gene perturbation vs. double-gene perturbation, as well as triple-gene perturbations vs. all other perturbations.

The first new type of scoring factor, double perturbation vs. component single perturbation, is very similar to the previous scoring factors. The allowed regions are easy to work through, following the example of Figure 7.1. As in Figure 7.1, each possible response attachment results in a different observation distribution.

The second type of scoring factor, double perturbation vs. independent single perturbation, presents new complications. The score depends not only on the observed data, but also on the interaction between the genes of the double-gene perturbation. This adds one more dimension to the factor, what was previously $\mathcal{I} \times \Re^2$ is now $\mathcal{I}^2 \times \Re^2$. The allowed regions in each of the 36 different combinations of pairwise interactions can again be worked out as in the previous two types of scoring factors.

Due to this additional edge in the factor graph, the structure is no longer tree-like. Solving the factor graph will now require message passing, and summarizing over this new type of factor. When the factor node AB:C sends a message to A:B, it must incorporate the incoming message from B:C, by summing out the appropriate dimension.

11.2 Evaluating network

In the previous section I outlined extensions to our multiple-gene perturbation learning methods. I plan to evaluate each method's prediction efficacy on the dataset with *cdgA* using the following methods.

11.2.1 Confident placement of *cdgA*

In order to see if *cdgA* is confidently placed, I will examine its placement in the most likely models. First, among the top five most likely models I will ensure that *cdgA* has a majority vote for it's regulation program and how it regulates other genes. Second I will use model averaging as in §6.1 to determine the confidence of each of these links that include *cdgA*. The networks explored in the hill climbing will provide the sample of the network subspace. I will ensure that the regulatory links of *cdgA* have a posterior probability at least 1.5 that of the prior, based on the finding in the *D. melanogaster* pathway.

11.2.2 Verify known network components

The predicted network should retain most of the known network structure, as shown and predicted in §8.2. The new data associated with *cdgA* may change some of the predictions, but a complete reinterpretation of the existing data seems very unlikely.

11.2.3 Assess significance and robustness

I will use methods in §10.2.2 and §10.2.1 to assess the significance and robustness of the predicted network.

11.3 Recommending new deletions

In §10.3.2 I outlined an active learning framework for choosing new perturbations. If the response genes in this framework are interpreted to be response gene with a fixed θ_j, then the equations from 10.5 to 10.6 still hold. However, the network modeling is different in two ways between the methods: predicted responses from perturbations are different, and the observations are discrete rather than continuous.

The solution to these difficulties is straightforward in the joint intervention model. First, for a particular setting of β_j, we add a new model interventional model to the set of existing perturbation Bayes nets. This means adding one more box in Figure 8.1. We also add additional data, $x_{j'}(\Delta j)$, and construct additional observation nodes. Each new observation node compares the new perturbation j to the data from another perturbation. Marginalizing out the rest of the Bayesian network results in the predicted discrete probability distribution over the new perturbation data. Both of the approximations used for equation 10.1 still apply.

Another difference is that most of the genes in a cluster will have identical expression after the quantization, so that each gene in a cluster will result in identical predicted information gain. This method is more informative of which cluster's perturbation is most likely to contain information about the network. Therefore, within the best cluster identified by the entropy reduction, I will choose genes for perturbation that are most likely to be regulators based on their annotation, such as genes that are known to have a DNA or protein binding domain.

Though this dataset already has all possible double deletions, we can easily evaluate the information gain of a deletion by the same method. Instead of adding a single-gene deletion network to the joint intervention network, we simply add the multiple-gene deletion network. Finding the predicted distribution of the data uses identical methods to the single-gene deletion case.

11.4 Evaluate new deletions

To see the effectiveness of the information-based knockdown recommendation algorithm, I will collaborate with the Yildiz lab to test the recommendations. If the recommended knockouts have an effect on biofilm formation, we will perform gene expression profiling on the deletion strains and incorporate the new knockdowns into the biofilm network.

Chapter 12

Aim 3: A public resource for network visualization, prediction, and supporting data

Due to their graphical nature, biological networks are best communicated visually. There are software applications for interactive visualization on a desktop computer, requiring a software install. However, none of these tools make it very easy to locate data. There are also a few interactive tools on the web for visualizing networks, which have the advantage of not requiring a software install and already covering a data source. However, these cover only a small number of data sources and don't provide a way to add additional data sources.

One of the largest difficulties that I have experienced so far is validation of my predicted networks. Ideally, verification should use independent data, so that not only is the network prediction verified, but also the source data which gave rise to the predictions. The best way to do this is to use existing datasets: curated pathways, curated interactions, protein-protein interaction data, and other high-throughput data. Any single dataset is unlikely to provide all possible verification, and it is therefore desirable to show many datasets at once.

To fill both these needs I propose a web based, network visualization with immediate access to all leading interaction datasets and the ability to add one's own datasets, as well as the tools to compare and validate networks. In Chapter 9, I covered the existing features of what we call the Interaction Browser, after the

UCSC Genome Browser [50]. The Interaction Browser can currently display a network representation with genes and with edges. As in the Genome Browser, each independent data source is referred to as a track, however these tracks on the Interaction Browser consist of pairwise data, *i.e.* links between genes.. Each edge between two genes will display all applicable tracks. The graph is interactive in that the genes can be moved, and genes can be added or removed at will. Adding genes to the display automatically pulls in edges from the enabled data sources.

In this chapter I describe new features that will be necessary to make a useful, collaborative tool for the aims previously outlined in this proposal. First, I will describe the features of the visualization that are necessary for working with interaction data and for achieving my previous aims. Next, I will describe the data sources that will be pre-loaded into Interaction Browser and the data import and export features available to users. Finally, I will describe how the methods from my previous aims will be made publicly available.

12.1 Web-based network visualization

Many of the basic features of network visualization are already in place in the Interaction Browser. Users can display networks with arbitrary genes and tracks. Users can search for genes by annotation or name, add the genes to the network, and the appropriate network edges will be fetched from the server and displayed. Users can manually layout the graph by moving genes with the mouse, or apply a spring based network layout that runs in the browser.

I plan three features that I believe are essential for my research aims and to make the visualization collaborative. First, in order to enable multi-session use and collaboration, users should be able to save the current state of their browsing and share it with others. Second, the Interaction Browser should be able to perform transformational operations on the tracks to assist in combining information from multiple data sources and Finally, the Interaction Browser should show the data that connects to genes not in the current working set, both to allow easy exploration and prevent bias.

12.1.1 Saving and sharing network visualizations

Choosing genes, selecting networks, and laying them out on the screen can represent a considerable investment of time and thought. Saving and sharing the current state of the visualization so that it can be retrieved and used as a starting point at a different time or on a different computer is essential to making the Interaction

Browser useful. Since the Interaction Browser is a web application, the most appropriate way to maintain state is in the URL of the current visualization.

There are three primary components to the state: the current working set of genes, the 2D coordinates of each gene, and the set of enabled tracks. I will implement two ways of encoding this into the URL.

First, the encoding can be completely transparent, using standard HTML form URL encoding. The advantage of this is that it provides an extremely simple way for other tools to link to the interaction browser in an automated fashion. Nearly every database supports links in this manner. Each component of the data will have a different parameter: `g` for the current working set, `t` for enabled tracks, and (optionally) `p` for a list of the positions of the current working set. An example might be:

```
http://ib.ucsc.edu/celegans?g=let-1,unc-42&t=StuartPheno,CompendiumCoexpression&p=10,20,-30,40
```

URLs like this will grow with the size of the network. There are practical limits on the use of very long URLs, both in what the browser can handle and in what a person can easily share with another. Therefore, when using the interaction browser there will be the option to save the current state to a short identifier, with the full state being saved on the server indefinitely. Such a URL might look like:

```
http://ib.ucsc.edu/net?s=q93VoZT
```

The advantage of this scheme is that it is easily communicated through email and possibly even verbally.

12.1.2 Track operations

A very successful component of the UCSC Genome Browser has been the Table Browser [49]. The Table Browser allows the creation of custom tracks that can then be immediately viewed in the browser. Such tracks can summarize or expand upon the information in the browser. I have already identified the need for such operations in §10.1.2, and I believe that will these operations will allow investigation in ways that surpass any current network visualization tool.

Track intersection and union

Intersection and union both operate on two tracks and result in one track. They operate like the set operations: the result of intersection contains a link if and only if that link is in both of the operand tracks, and the result of union contains a link if and only if the link is in either of the operand tracks. A use for intersection would be to narrow the results of two noisy data-sources into a more confident one. A use for union would be to obtain a single track that has more coverage than the source tracks.

Track transitive closure

Transitive closure operates on a single track and expands upon the track by following links within the track. The result contains all of the original links, and all links such that if there is a path between a pair of genes in the original track, then that link is in the result. To make this more useful, there will be an optional parameter k that limits the lengths of paths that are included in the result. For example, with $k = 2$ the resulting track will contain all links that result by following a link once. See §10.1.2 for an example use of this operation.

Track transitive reduction

Transitive reduction is an operation on a single track which removes edges. In some sense, it is the inverse of transitive closure, in that it removes an edge if there exists a longer path in the network to explain it. However, there is not always a unique transitive reduction of a graph, and arbitrary decisions must sometimes be made. For example, in undirected tracks, a triple of genes all connected to each other can be transitively reduced by removing any of the three links. Despite these difficulties, there are cases where transitive reduction will be very useful, such as when all the edges have unique weights or in directed graphs, where it will commonly be desired to remove extra edges that can be explained by another path.

12.1.3 Neighborhood views

The Interaction Browser currently displays all the edges between a current working set of genes. This eliminates much the visual clutter of looking at the links that do not pertain to the current working set of genes. However, this other data, the links to genes not currently in the working set, can be of great use, and by ignoring it all the time an investigator may come to less complete conclusions about the genes in the current working set or about the tracks that are currently being displayed.

I will add a "neighborhood view" to the Interaction Browser that will display in the periphery of the visualization all genes that are not currently in the working set, but that are directly connected to genes in the working set via an enabled track. Any gene in the periphery can be added to the current working set by dragging it into the main visualization area. This mode will be switchable, to better allow network expansion when turned on and network inspection when turned off.

To make choosing neighborhood genes easier, particularly when there are many neighborhood genes, I will implement Bayesian Iterative Updating [70]. Bayesian Iterative Updating uses multiple networks to find related genes. I will highlight neighborhood genes that Bayesian Iterative Updating recommends, and also highlight genes in the current working set that Bayesian Iterative Updating finds to be unrelated.

12.2 Public data and user data

A primary goal of the Interaction Browser is to easily visualize many different data sources at once. In order to make viewing many data sources easy, they should already be loaded into the browser, and be accessible with a minimum of clicks. Additionally, users should be able to upload their own non-public data as simply as possible. Finally, all data that's in the interaction browser should be easy to download, including the network visualizations.

12.2.1 Organisms, identifiers, and data sources

I have predicted networks on three organisms: human, yeast, and *V. cholerae*, so I will have all of these as target organisms. Greg Dougherty has created an alias mapping tool which we will use in the interaction browser to map the identifiers between various data sources. For human, I will load the identifiers from UniProt, as these are used in the cancer data set. For yeast, I will load the aliases from SGD. For *V. cholerae* I will load the accession identifiers and the common gene names for genes that have such a name.

I will add curated and computationally predicted interaction data from BIND, DIP, Reactome, and BioCarta for these three organisms. I will perform a literature search to identify any protein-protein interaction datasets that cover more than 100 genes in any of these organisms. I will also add any ChIP-chip data sets in these organisms that covers more than 5 transcription factors. If there are few *V. cholerae* datasets, I will map the data from *E. coli* using a best-reciprocal-BLAST-hit mapping.

12.2.2 Track format

The default track format for the Interaction Browser will be a tab-delimited format. There will be a single link per line, and the first entry and second entry on each line will be the genes involved in the link. The next entry on the line will be an optional link score, or weight. The string values `NA`, `.`, and `NaN`, in additon to a zero-length string, will all represent missing data for the link. The fourth and final entry on a line will be an optional directionality, with `>` representing a direction from the first gene to the second and `<` representing a link from the second to the first gene. If both `>` and `<` are present then the link is bidirectional.

In addition, the Interaction Browser will support BioPAX and Cytoscape formats for reading in links.

12.2.3 Custom tracks

As in the Genome Browser, users will be able to submit custom tracks of links. These links can either be uploaded from the users computer, or specified by a URL which the Interaction Browser will download. These tracks will be stored on the server on a temporary basis. They will be identified with a single session, which will be kept in the URL of the page.

12.2.4 Network visualization output

The Interaction Browser will support downloading the current network visualization as raster graphics in SVG, PostScript, and PDF, and in bitmap graphics as PNG.

12.3 Publicly accessible implementation of algorithms

I will provide publicly accessible implementations of the learning algorithms in Chapters 10 and 11 through the Interaction Browser. This will allow users of my algorithms to simultaneously see supporting information for predictions.

12.3.1 Microarray data in the browser

The Interaction Browser currently has some support for matrix based data for microarrays implemented by Greg Dougherty. I will extend this support to allow for annotation of knockdown and knockout columns. I will also extend the matrix support to allow the upload of custom matrices in the same vein as custom tracks.

12.3.2 Prediction outputs

The output of both of my single-gene and multiple-gene perturbation methods are a set of networks. Each of these networks will be available in the Interaction Browser as custom tracks.

12.3.3 Validation of predictions against high-throughput data tracks

I will show validation of the predicted tracks in the browser in two ways. First, I will implement a function to recommend tracks that significantly overlap the current view of the network. This will let users visually inspect data from, for example, protein-protein interaction or curated pathways. Second, I will provide functionality for quantitative comparison of tracks over the current working set of genes. This will include edge overlap counts and percentages.

Bibliography

[1] S. M. Aji and R. J. McEliece. A general algorithm for distributing information in a graph. In *1997 IEEE International Symposium on Information Theory. Proceedings.*, pages 6–6, 1997.

[2] S. M. Aji and R. J. McEliece. The generalized distributive law. *Information Theory, IEEE Transactions on*, 46(2):325–343, 2000.

[3] Randy J. Arnold and James P. Reilly. Observation ofescherichia coliribosomal proteins and their post-translational modifications by mass spectrometry. *Analytical Biochemistry*, 269(1):105–112, 1999.

[4] P L Bartel, J A Roecklein, D SenGupta, and S Fields. A protein linkage map of escherichia coli bacteriophage t7. *Nat Genet*, 12(1):72–77, 1996.

[5] A. Baudin, O. Ozier-Kalogeropoulos, A. Denouel, F. Lacroute, and C. Cullin. A simple and efficient method for direct gene deletion in saccharomyces cerevisiae. *Nucleic Acids Research*, 21(14):3329–3330, 1993.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. In *IEEE International Conference on Communications, 1993. Technical Program, Conference Record,*, volume 2, pages 1064–1070 vol.2, 1993.

[7] Michael Boutros, Herve Agaisse, and Norbert Perrimon. Sequential activation of signaling pathways during innate immune responses in drosophila. *Dev Cell*, 3(5):711–722, 2002.

[8] Catharina Casper-Lindley and Fitnat H. Yildiz. Vpst is a transcriptional regulator required for expression of vps biosynthesis genes and the development of rugose colonial morphology in vibrio cholerae o1 el tor. *The Journal of Bacteriology*, 186(5):1574–1578, 2004.

[9] Ing-Feng Chang. Mass spectrometry-based proteomic analysis of the epitope-tag affinity purified protein complexes in eukaryotes. *Proteomics*, 6(23):6158–6166, 2006.

[10] M H Cobb. Map kinase pathways. *Prog Biophys Mol Biol*, 71(3-4):479–500, 1999.

[11] Autumn A. Cuellar, Catherine M. Lloyd, Poul F. Nielsen, David P. Bullivant, David P. Nickerson, and Peter J. Hunter. An overview of cellml 1.1, a biological model description language. *SIMULATION*, 79(12):740–747, 2003.

[12] Bart Deplancke, Arnab Mukhopadhyay, Wanyuan Ao, Ahmed M Elewa, Christian A Grove, Natalia J Martinez, Reynaldo Sequerra, Lynn Doucette-Stamm, John S Reece-Hoyes, Ian A Hope, Heidi A Tissenbaum, Susan E Mango, and Albertha J M Walhout. A gene-centered c. elegans protein-dna interaction network. *Cell*, 125(6):1193–1205, 2006.

[13] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, July 1999.

[14] A Einhauer and A Jungbauer. The flag peptide, a versatile fusion tag for the purification of recombinant proteins. *J Biochem Biophys Methods*, 49(1-3):455–465, 2001.

[15] Ghia M Euskirchen, Joel S Rozowsky, Chia-Lin Wei, Wah Heng Lee, Zhengdong D Zhang, Stephen Hartman, Olof Emanuelsson, Viktor Stolc, Sherman Weissman, Mark B Gerstein, Yijun Ruan, and Michael Snyder. Mapping of transcription factor binding regions in mammalian cells by chip: comparison of array- and sequencing-based technologies. *Genome Res*, 17(6):898–909, 2007.

[16] S Fields and O Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, 1989.

[17] Stanley Fields and Rolf Sternglanz. The two-hybrid system: an assay for protein-protein interactions. *Trends in Genetics*, 10(8):286–292, August 1994.

[18] A Finney and M Hucka. Systems biology markup language: Level 2 and beyond. *Biochem Soc Trans*, 31(Pt 6):1472–1473, 2003.

[19] A Fire, S Xu, M K Montgomery, S A Kostas, S E Driver, and C C Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998.

[20] Peter Fraser and Wendy Bickmore. Nuclear organization of the genome and the potential for gene regulation. *Nature*, 447(7143):413–417, 2007.

[21] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[22] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[23] Karla Jean Fullner and John J. Mekalanos. Genetic characterization of a new type iv-a pilus gene cluster found in both classical and el tor biotypes of vibrio cholerae. *Infection and Immunity*, 67(3):1393–1404, 1999.

[24] Gallager. *Low Density Parity Check Codes.* M.I.T. Press, 1963.

[25] E Gari, L Piedrafita, M Aldea, and E Herrero. A set of vectors with a tetracycline-regulatable promoter system for modulated gene expression in saccharomyces cerevisiae. *Yeast*, 13(9):837–848, 1997.

[26] Irit Gat-Viks and Ron Shamir. Refinement and expansion of signaling pathways: The osmotic response network in yeast. *Genome Research*, 17(3):358–367, 2007.

[27] Irit Gat-Viks, Amos Tanay, Daniela Raijman, and Ron Shamir. The factor graph network model for biological systems. *: Research in Computational Molecular Biology*, pages 31–47, 2005.

[28] Irit Gat-Viks, Amos Tanay, Daniela Raijman, and Ron Shamir. A probabilistic methodology for integrating knowledge and experiments on biological networks. *Journal of Computational Biology*, 13(2):165–181, March 2006.

[29] Anne-Claude Gavin, Markus Bosche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jorg Schultz, Jens M Rick, Anne-Marie Michon, Cristina-Maria Cruciat, Marita Remor, Christian Hofert, Malgorzata Schelder, Miro Brajenovic, Heinz Ruffner, Alejandro Merino, Karin Klein, Manuela Hudak, David Dickson, Tatjana Rudi, Volker Gnau, Angela Bauch, Sonja Bastuck, Bettina Huhse, Christina Leutwein, Marie-Anne Heurtier, Richard R Copley, Angela Edelmann, Erich Querfurth, Vladimir Rybin, Gerard Drewes, Manfred Raida, Tewis Bouwmeester, Peer Bork, Bertrand Seraphin, Bernhard Kuster, Gitte Neubauer, and Giulio Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.

[30] Guri Giaever, Angela M. Chu, Li Ni, Carla Connelly, Linda Riles, Steeve Veronneau, Sally Dow, Ankuta Lucau-Danila, Keith Anderson, Bruno Andre, Adam P. Arkin, Anna Astromoff, Mohamed El Bakkoury, Rhonda Bangham, Rocio Benito, Sophie Brachat, Stefano Campanaro, Matt Curtiss, Karen Davis, Adam Deutschbauer, Karl-Dieter Entian, Patrick Flaherty, Francoise Foury, David J. Garfinkel, Mark Gerstein, Deanna Gotte, Ulrich Guldener, Johannes H. Hegemann, Svenja Hempel, Zelek Herman, Daniel F. Jaramillo, Diane E. Kelly, Steven L. Kelly, Peter Kotter, Darlene LaBonte, David C. Lamb, Ning Lan, Hong Liang, Hong Liao, Lucy Liu, Chuanyun Luo, Marc Lussier, Rong Mao, Patrice Menard, Siew Loon Ooi, Jose L. Revuelta, Christopher J. Roberts, Matthias Rose, Petra Ross-Macdonald, Bart Scherens, Greg Schimmack, Brenda Shafer, Daniel D. Shoemaker, Sharon Sookhai-Mahadeo, Reginald K. Storms, Jeffrey N. Strathern, Giorgio Valle, Marleen Voet, Guido Volckaert, Ching-yun Wang, Teresa R. Ward, Julie Wilhelmy, Elizabeth A. Winzeler, Yonghong Yang, Grace Yen, Elaine Youngman, Kexin Yu, Howard Bussey, Jef D. Boeke, Michael Snyder, Peter Philippsen, Ronald W. Davis, and Mark Johnston. Functional profiling of the saccharomyces cerevisiae genome. *Nature*, 418(6896):387–391, 2002.

[31] Daniel Gietz, Andrew St. Jean, Robin A. Woods, and Robert H. Schiestl. Improved method for high efficiency transformation of intact yeast cells. *Nucleic Acids Research*, 20(6):1425–, 1992.

[32] Brian K. Hammer and Bonnie L. Bassler. Quorum sensing controls biofilm formation in vibrio cholerae. *Molecular Microbiology*, 50(1):101–104, 2003.

[33] Lin He and Gregory J Hannon. Micrornas: small rnas with a big role in gene regulation. *Nat Rev Genet*, 5(7):522–531, 2004.

[34] Henning Hermjakob, Luisa Montecchi-Palazzi, Gary Bader, Jerome Wojcik, Lukasz Salwinski, Arnaud Ceol, Susan Moore, Sandra Orchard, Ugis Sarkans, Christian von Mering, Bernd Roechert, Sylvain Poux, Eva Jung, Henning Mersch, Paul Kersey, Michael Lappe, Yixue Li, Rong Zeng, Debashis Rana, Macha Nikolski, Holger Husi, Christine Brun, K Shanker, Seth G N Grant, Chris Sander, Peer Bork, Weimin Zhu, Akhilesh Pandey, Alvis Brazma, Bernard Jacq, Marc Vidal, David Sherman, Pierre Legrain, Gianni Cesareni, Ioannis Xenarios, David Eisenberg, Boris Steipe, Chris Hogue, and Rolf Apweiler. The hupo psi's molecular interaction format–a community standard for the representation of protein interaction data. *Nat Biotechnol*, 22(2):177–183, 2004.

[35] Yuen Ho, Albrecht Gruhler, Adrian Heilbut, Gary D Bader, Lynda Moore, Sally-Lin Adams, Anna Millar, Paul Taylor, Keiryn Bennett, Kelly Boutilier, Lingyun Yang, Cheryl Wolting, Ian Donaldson, Soren Schandorff, Juanita Shewnarane, Mai Vo, Joanne Taggart, Marilyn Goudreault, Brenda Muskat, Cris Alfarano, Danielle Dewar, Zhen Lin, Katerina Michalickova, Andrew R Willems, Holly Sassi, Peter A Nielsen, Karina J Rasmussen, Jens R Andersen, Lene E Johansen, Lykke H Hansen, Hans Jespersen, Alexandre Podtelejnikov, Eva Nielsen, Janne Crawford, Vibeke Poulsen, Birgitte D Sorensen, Jesper Matthiesen, Ronald C Hendrickson, Frank Gleeson, Tony Pawson, Michael F Moran, Daniel Durocher, Matthias Mann, Christopher W V Hogue, Daniel Figeys, and Mike Tyers. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–183, 2002.

[36] Christine E Horak and Michael Snyder. Chip-chip: a genomic approach for identifying transcription factor binding sites. *Methods Enzymol*, 350:469–483, 2002.

[37] Zhenjun Hu, Joe Mellor, Jie Wu, Minoru Kanehisa, Joshua M Stuart, and Charles DeLisi. Towards zoomable multidimensional maps of the cell. *Nat Biotech*, 25(5):547–554, 2007.

[38] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novere, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence,

J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[39] Falk Huffner, Sebastian Wernicke, and Thomas Zichner. Faspad: fast signaling pathway detection. *Bioinformatics*, 23(13):1708–1709, 2007.

[40] 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(1):109–26, 2000.

[41] Albert Y Hung and Morgan Sheng. Pdz domains: structural modules for protein complex assembly. *J Biol Chem*, 277(8):5699–5702, 2002.

[42] Gyorgy Hutvagner and Phillip D. Zamore. Rnai: nature abhors a double-strand. *Current Opinion in Genetics & Development*, 12(2):225–232, 2002.

[43] Soren Impey, Sean R McCorkle, Hyunjoo Cha-Molstad, Jami M Dwyer, Gregory S Yochum, Jeremy M Boss, Shannon McWeeney, John J Dunn, Gail Mandel, and Richard H Goodman. Defining the creb regulon: a genome-wide analysis of transcription factor regulatory regions. *Cell*, 119(7):1041–1054, 2004.

[44] T Ito, T Chiba, R Ozawa, M Yoshida, M Hattori, and Y Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A*, 98(8):4569–4574, 2001.

[45] F Jacob and J Monod. Genentic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3(3):318–56, 1961.

[46] P James, J Halladay, and E A Craig. Genomic libraries and a host strain designed for highly efficient two-hybrid selection in yeast. *Genetics*, 144(4):1425–1436, 1996.

[47] Hongkai Ji, Steven A Vokes, and Wing H Wong. A comparative analysis of genome-wide chromatin immunoprecipitation data for mammalian transcription factors. *Nucleic Acids Res*, 34(21):e146, 2006.

[48] J T Kadonaga. Eukaryotic transcription: an interlaced network of transcription factors and chromatin-modifying machines. *Cell*, 92(3):307–313, 1998.

[49] Donna Karolchik, Angela S. Hinrichs, Terrence S. Furey, Krishna M. Roskin, Charles W. Sugnet, David Haussler, and W. James Kent. The ucsc table browser data retrieval tool. *Nucleic Acids Research*, 32(suppl_1):D493–496, 2004.

[50] W. James Kent, Charles W. Sugnet, Terrence S. Furey, Krishna M. Roskin, Tom H. Pringle, Alan M. Zahler, and David Haussler. The human genome browser at ucsc. *Genome Research*, 12(6):996–1006, 2002.

[51] Ingrid M. Keseler, Julio Collado-Vides, Socorro Gama-Castro, John Ingraham, Suzanne Paley, Ian T. Paulsen, Martin Peralta-Gil, and Peter D. Karp. Ecocyc: a comprehensive database resource for escherichia coli. *Nucleic Acids Research*, 33(suppl_1):D334–337, 2005.

[52] M C King and A C Wilson. Evolution at two levels in humans and chimpanzees. *Science*, 188(4184):107–116, 1975.

[53] Kenneth S. Kosik. The neuronal microrna system. *Nat Rev Neurosci*, 7(12):911–920, 2006.

[54] Kschischang, Frey, and Loeliger. Factor graphs and the sum-product algorithm. *IEEETIT: IEEE Transactions on Information Theory*, 47, 2001.

[55] Tong Ihn Lee, Nicola J Rinaldi, Francois Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, Julia Zeitlinger, Ezra G Jennings, Heather L Murray, D Benjamin Gordon, Bing Ren, John J Wyrick, Jean-Bosco Tagne, Thomas L Volkert, Ernest Fraenkel, David K Gifford, and Richard A Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

[56] Ivica Letunic, Richard R Copley, Birgit Pils, Stefan Pinkert, Jorg Schultz, and Peer Bork. Smart 5: domains in the context of genomes and networks. *Nucleic Acids Res*, 34(Database issue):D257–60, 2006.

[57] Benjamin P. Lewis, Richard E. Green, and Steven E. Brenner. Evidence for the widespread coupling of alternative splicing and nonsense-mediated mrna decay in humans. *Proceedings of the National Academy of Sciences*, 100(1):189–192, 2003.

[58] Bentley Lim, Sinem Beyhan, James Meir, and Fitnat H. Yildiz. Cyclic-digmp signal transduction systems in vibrio cholerae: modulation of rugosity and biofilm formation. *Molecular Microbiology*, 60(2):331–348, 2006.

[59] Joanne S. Luciano. Pax of mind for pathway researchers. *Drug Discovery Today*, 10(13):937–942, 2005.

[60] Florian Markowetz, Jacques Bloch, and Rainer Spang. Non-transcriptional pathway features reconstructed from secondary effects of rna interference. *Bioinformatics*, 21(21):4026–4032, 2005.

[61] Florian Markowetz, Dennis Kostka, Olga G. Troyanskaya, and Rainer Spang. Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, 23(13):i305–312, 2007.

[62] Suresh Mathivanan, Balamurugan Periaswamy, T K B Gandhi, Kumaran Kandasamy, Shubha Suresh, Riaz Mohmood, Y L Ramachandra, and Akhilesh Pandey. An evaluation of human protein-protein interaction data in the public domain. *BMC Bioinformatics*, 7 Suppl 5:S19, 2006.

[63] M. Mezard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[64] Oved Ourfali, Tomer Shlomi, Trey Ideker, Eytan Ruppin, and Roded Sharan. Spine: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23(13):i359–366, 2007.

[65] P Pavlidis and W S Noble. Analysis of strain and regional variation in gene expression in mouse brain. *Genome Biol*, 2(10):RESEARCH0042, 2001.

[66] J. Pearl. *Causality: Models, Reasoning, and Inference*. Causality, by Judea Pearl, pp. 400. ISBN 0521773628. Cambridge, UK: Cambridge University Press, March 2000., March 2000.

[67] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[68] Suraj Peri, J Daniel Navarro, Ramars Amanchy, Troels Z Kristiansen, Chandra Kiran Jonnalagadda, Vineeth Surendranath, Vidya Niranjan, Babylakshmi Muthusamy, T K B Gandhi, Mads Gronborg, Nieves Ibarrola, Nandan Deshpande, K Shanker, H N Shivashankar, B P Rashmi, M A Ramya, Zhixing Zhao, K N Chandrika, N Padma, H C Harsha, A J Yatish, M P Kavitha, Minal Menezes, Dipanwita Roy Choudhury, Shubha Suresh, Neelanjana Ghosh, R Saravana, Sreenath Chandran, Subhalakshmi Krishna, Mary Joy, Sanjeev K Anand, V Madavan, Ansamma Joseph, Guang W Wong, William P Schiemann, Stefan N Constantinescu, Lily Huang, Roya Khosravi-Far, Hanno Steen, Muneesh Tewari, Saghi Ghaffari, Gerard C Blobe, Chi V Dang, Joe G N Garcia, Jonathan Pevsner, Ole N Jensen, Peter Roepstorff, Krishna S Deshpande, Arul M Chinnaiyan, Ada Hamosh, Aravinda Chakravarti, and Akhilesh Pandey. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res*, 13(10):2363–2371, 2003.

[69] Jeffrey A. Pleiss, Gregg B. Whitworth, Megan Bergkessel, and Christine Guthrie. Transcript specificity in yeast pre-mrna splicing revealed by mutations in core spliceosomal components. *PLoS Biology*, 5(4), 2007.

[70] Corey Powell. An iterative bayesian updating method for biological pathway predictien. Master's thesis, University of California Santa Cruz, 2004.

[71] W Reik and N D Allen. Genomic imprinting. imprinting with and without methylation. *Curr Biol*, 4(2):145–147, 1994.

[72] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.

[73] Jacob Scott, Trey Ideker, Richard M. Karp, and Roded Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology*, 13(2):133–144, 2006.

[74] Nicola Soranzo, Ginestra Bianconi, and Claudio Altafini. Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics*, 23(13):1640–1647, 2007.

[75] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.

[76] Martin Steffen, Allegra Petti, John Aach, Patrik D'haeseleer, and George Church. Automated modelling of signal transduction networks. *BMC Bioinformatics*, 3(1), 2002.

[77] Lena Stromback and Patrick Lambrix. Representations of molecular pathways: an evaluation of sbml, psi mi and biopax. *Bioinformatics*, 21(24):4401–4407, 2005.

[78] P Uetz, L Giot, G Cagney, T A Mansfield, R S Judson, J R Knight, D Lockshon, V Narayan, M Srinivasan, P Pochart, A Qureshi-Emili, Y Li, B Godwin, D Conover, T Kalbfleisch, G Vijayadamodar, M Yang, M Johnston, S Fields, and J M Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[79] M Vidal and P Legrain. Yeast forward and reverse 'n'-hybrid systems. *Nucleic Acids Res*, 27(4):919–929, 1999.

[80] E A Winzeler, D D Shoemaker, A Astromoff, H Liang, K Anderson, B Andre, R Bangham, R Benito, J D Boeke, H Bussey, A M Chu, C Connelly, K Davis, F Dietrich, S W Dow, M El Bakkoury, F Foury, S H Friend, E Gentalen, G Giaever, J H Hegemann, T Jones, M Laub, H Liao, N Liebundguth, D J Lockhart, A Lucau-Danila, M Lussier, N M'Rabet, P Menard, M Mittmann, C Pai, C Rebischung, J L Revuelta, L Riles, C J Roberts, P Ross-MacDonald, B Scherens, M Snyder, S Sookhai-Mahadeo, R K Storms, S Veronneau, M Voet, G Volckaert, T R Ward, R Wysocki, G S Yen, K Yu, K Zimmermann, P Philippsen, M Johnston, and R W Davis. Functional characterization of the s. cerevisiae genome by gene deletion and parallel analysis. *Science*, 285(5429):901–906, 1999.

[81] Ira G Wool. The structure and function of eukaryotic ribosoms. *Annual Review of Biochemistry*, 48:719–754, 1979.

[82] BioPAX working group. Biopax–biological pathways exchange language. Documentation, 2004.

[83] Chen-Hsiang Yeang and Tommi Jaakkola. Physical network models and multi-source data integration. In *RECOMB*, pages 312–321, 2003.

[84] Chen-Hsiang Yeang, H Craig Mak, Scott McCuine, Christopher Workman, Tommi Jaakkola, and Trey Ideker. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology*, 6(7):R62, 2005.

[85] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 689–695. M.I.T. Press, 2001.

[86] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.

[87] Zhengdong D Zhang, Alberto Paccanaro, Yutao Fu, Sherman Weissman, Zhiping Weng, Joseph Chang, Michael Snyder, and Mark B Gerstein. Statistical analysis of the genomic distribution and correlation of regulatory elements in the encode regions. *Genome Res*, 17(6):787–797, 2007.