Physical Network Models and Multi-source Data Integration

Chen-Hsiang Yeang
MIT AI Lab
200 Technology Square
Cambridge, MA 02139, USA
chyeang@ai.mit.edu

Tommi Jaakkola
MIT AI Lab
200 Technology Square
Cambridge, MA 02139, USA
tommi@ai.mit.edu

ABSTRACT

We develop a new framework for inferring models of transcriptional regulation. The models in this approach, which we call *physical models*, are constructed on the basis of verifiable molecular attributes of the underlying biological system. The attributes include, for example, the existence of protein-protein and protein-DNA interactions in gene regulatory processes, the directionality of signal transduction in protein-protein interactions, as well as the signs of the immediate effects of these interactions (e.g., whether an upstream gen activates or represses the downstream genes). Each attribute is included as a variable in the model, and the variables define a collection of annotated random graphs. Possible configurations of these variables (realizations of the underlying biological system) are constrained by the available data sources. Some of the data sources such as factor-binding data (location data) involve measurements that are directly tied to the variables in the model. Other sources such as gene knock-outs are *functional* in nature and provide only indirect evidence about the (physical) variables. We associate each knock-out effect in the deletion mutant data with a set of causal paths (molecular cascades) that could in principle explain the effect, resulting in aggregate constraints about the physical variables in the model. The most likely setting of all the variables is found by the max-product algorithm. By testing our approach on datasets related to the pheromone response pathway in *S. cerevisiae*, we demonstrate that the resulting transcriptional models are consistent with previous studies about the pathway. Moreover, we show that the approach is capable of predicting gene knock-out effects with high degree of accuracy in a cross-validation setting. The method also implicates likely molecular cascades responsible for each observed knock-out effect. The inference results are robust against variations in the model parameters. We can extend the approach to include other data sources (solve the corresponding data association problems), including, for example, time course expression profiles. We also discuss coordinated regulation and the use of automated experiment design[1].

RECOMB'03, April 10–13, 2003, Berlin, Germany.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences

General Terms

Algorithms.

Keywords

Physical models, Regulatory networks, Data fusion.

1. INTRODUCTION

Understanding transcriptional regulation is a leading problem in contemporary biology. While certain subsystems in model organisms have been studied in great detail (for instance, the galactose metabolism pathway in *S. cerevisiae* [7]), understanding regulatory mechanisms at the genomic and system level remains a grand challenge.

A diverse collection of high-throughput biological data sources are currently available for elucidating transcriptional regulatory mechanisms. This includes, for example, expression microarrays (e.g., [3]), factor-binding profiling or location analysis (e.g., [8, 6]), and yeast two-hybrid experiments (e.g., [10]). We divide the available data sources into *physical* data sources that directly pertain to the underlying molecular interactions (location and two-hybrid) and *functional* data sources that do not directly measure specific molecular events but rather capture cellular responses (expression arrays). A number of computational approaches have been developed for inferring aspects of the underlying biological regulatory system on the basis of such heterogeneous data sources. This includes, for example, Bayesian network models constrained by factor binding data [2], relational probabilistic models built from location, expression, and sequence data [9], as well as methods for finding groups of co-expressed genes sharing common regulators [6].

A computational model can be judged on the basis of its ability to explain or predict consequences of interventions such as gene knock-out effects or predict measurements carried out in the course of the natural operation of the biological system. The models determine what properties/features to explain in addition to how to explain them. The computational approaches listed above fall in the category of dependency models in terms of their treatment of functional data. In other words, they either seek to capture probabilistic dependencies among the available measurements [2, 9] or are directly guided by such dependency assessments [6] in addition to physical measurements. While relying on observed dependencies among expression profiles can be often useful, such dependencies are causally ambiguous as far as their underlying regulatory mechanisms are concerned. For example, co-expression of genes may result from a combination of molecular interactions involving both DNA binding regulators and protein-protein interactions. From biological perspective, we are interested not only in what groups of genes are up or down regulated together in a collection of experiments, but also what mechanisms cause the up or down regulation in certain conditions. Following Ideker et al. [4] we seek to incorporate functional observations by explicitly relying on networks of molecular interactions.

We describe transcriptional regulatory processes in terms of *annotated physical graphs*. Each physical graph articulates a specific hypothesis about the underlying regulatory system on the basis of only verifiable molecular interactions or properties. Since many important data sources available for inferring properties of transcriptional regulation (including gene knock-outs) are functional in nature, it is important to ensure that the molecular properties defined in the graphs maintain their clear meaning when such graphs are constrained also by functional observations. This can be accomplished by explicitly articulating both direct and indirect causal mechanisms underlying the functional observations in terms of the physical quantities (e.g., molecular cascades). We call models following this principle *physical models*. This modeling perspective shifts the computational effort from (largely unautomated) interpretation problems to data association problems – how variables are tied to the observations. The data association problem is largely avoided in dependency models, where the variables are more directly linked to the observations such as expression levels. The data association problem arising in physical models can be cast as a standard graphical model inference problem.

A simple realization of the core physical model we wish to infer is an annotated graph, where the nodes are associated with genes (or their protein products) and edges correspond to types of molecular interactions. We consider here two types of edges: protein-DNA and protein-protein interactions. A protein-DNA interaction has a distinctive direction (from the transcription factor to the DNA it binds to), and the direction of a protein-protein interaction is a priori unspecified (or bi-directional). The directionality of protein-protein edges will be determined on the basis of how the complexes are used in signal transduction pathways. Each type of edge is in addition annotated with a sign (positive or negative), where the sign represents the immediate molecular effect of the interaction. For example, a positive protein-DNA edge signifies that the transcription factor activates the expression of a specific gene, while a negative edge indicates that the factor is a repressor. The core variables in this version of the physical model are the presence/absence of edges and their signs. The physical model will also include additional variables pertaining to the selection of causal explanations (molecular cascades) for indirect observations. In the absence of any observed data, the model is a random

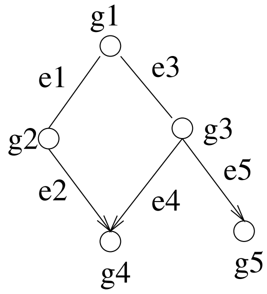

Figure 1: A simple example of physical interaction network

annotated graph without any clear preference over which graph represents a likely interpretation of the biological system. The probabilistic constraints arising from available data sources are incorporated into a factor graph model. The resulting most likely configuration (an annotated physical graph) can be solved with approximate inference methods such as the max-product algorithm (e.g,. [5]) or variants [14, 11]. The bulk of the effort in this paper concerns with establishing the association between the variables specifying the physical model and the available measurements.

We use three types of data to constrain physical models: protein-protein interactions derived from YPD database [2], 106 location (factor binding) profiles of yeast transcription factors [6], as well as 300 genome-wide expression profiles of knock-out experiments [3]. The first two data sources provide direct measurements of the values of the variables in the model. The knock-out data, albeit not directly tied to single types of variables, provide unambiguous measurements on the causal effects of the system. To incorporate the knock-out experiments, we explain a pair of knock-out interaction (from the deleted gene to an up or down regulated gene) with "valid" paths in the physical graph. This association constrains the variables along the selected path based on the evidence in the knock-out measurements.

We begin with an illustrative toy example. This is followed by a more general and formal description of the physical model and our solution to the data association problems. We subsequently describe the inference algorithms for finding the most likely models. Finally, we test our method on real datasets regarding the mating signal transduction pathway. The resulting models are consistent with previous studies and are capable of predicting the effects of gene knock-outs in a cross-validation setting.

2. AN ILLUSTRATIVE EXAMPLE

Figure 1 shows a simple network of protein-DNA and protein-protein interactions of 5 genes. Edges in the graph represent the set of possible interactions and we wish to infer which of these edges are indeed present, and the signs of these interactions. Directed edges denote protein-DNA interactions and undirected edges signify protein-protein interactions. There are possible 5 edges in this network.

For simplicity we assume the directions of protein-protein interactions e_1 and e_3 are known (from g_1 to g_2 and g_3). Here we consider only two types of variables: the presence of physical (protein-DNA and protein-protein) inter-

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

actions and the signs of these interactions (e.g., whether one gene has an immediate positive or negative effect on its downstream gene). More precisely, the variables include $x_1, \cdots, x_5$ and $s_1, \cdots, s_5$, where each x_i is an indicator variable encoding the presence/absence state of the physical interactions e_i and $s_i's$ are ± 1 variables providing the signs of the annotated edges. The values of the variables $x_1, \cdots, x_5$ are constrained directly by protein-protein or protein-DNA measurement(s) as well as indirectly by knock-out observations. In contrast, the signs $s_1, \cdots, s_5$ can be only inferred on the basis of knock-out effects. We must first formally tie the variables to the observed data and subsequently infer the most likely configuration of the variables in light of the available data.

Suppose now that we have protein-DNA and protein-protein interaction measurements pertaining to all the edges along with the error models characterizing the noise in the measurements. Furthermore, suppose we have observed that g_4 is down-regulated by knocking out g_1, and that there are no other significant knock-out effects among any pair of genes (nodes) in this network.

How can we constrain the values of $x_1, \cdots, x_5$, $s_1, \cdots, s_5$ from the data? If there are no errors in measurements then these three types of data are transformed into hard constraints of the configurations of variables: all the pairwise interactions (edges) are present. By hypothesizing that genes are regulated through cascades of protein-protein and protein-DNA interactions [4] we associate functional (knock-out) observations with constraints pertaining to sets of variables along potential regulatory pathways. Again assuming no errors in the measurements, the aggregate sign along path (e_1, e_2) or path (e_3, e_4) has to be positive. Note that the knock-out effect of a downstream gene is inverted from the observed effect.

The hard constraints fail to incorporate information about the confidence of measurements. Such confidence judgement can be derived from error models (if available) and represented as potential functions. In other words, hard constraints are replaced with potential functions (soft constraints) that capture the probabilistic nature of the observations. As a result, the joint distribution over the values of the variables is defined as a product of potential functions associated with the different observations. Finding the most likely setting of the variables reduces to an inference problem in graphical models (factor graphs). The inference problem is analogous to the problem of decoding error correcting codes [1], where approximate local inference algorithms have been quite successful.

More precisely, the error models governing protein-DNA and protein-protein interaction data give rise to potential functions $\phi_i(x_i)$ that bias the presence/absence of the edges in light of such measurements. Assume for simplicity that all x_i's are paired with identical potential functions: $\phi_i(0) = 1$ and $\phi_i(1) = 0.9$ (note that the potential functions need not be normalized). The potential functions indicate a slight bias against including edges in the graph. More generally $\frac{\phi_i(1)}{\phi_i(0)}$ should correspond to the ratio of the likelihoods under the hypotheses that the interaction exists or is absent. A more formal mapping is given later in the paper.

The constraint imposed by the knock-out effect (g_4 is down-regulated by g_1 knockout) is encoded into the following potential function:

$$\psi_1(x_1, \ldots, x_4, s_1, \ldots, s_4) = \begin{cases} 1.00 & \text{if } (x_1 = x_2 = 1, s_1 \cdot s_2 = +1) \\ & \vee(x_3 = x_4 = 1, s_3 \cdot s_4 = +1). \\ 0.01 & \text{otherwise.} \end{cases}$$

The value of the potential function is one if there is a valid explanation in terms of molecular cascades (paths). In order for a path to explain the knock-out effect, all interactions along the path must exist, and the aggregate of the sign must be consistent with the observed sign of the knock-out effect. The small value 0.01 represents the probability that the knock-out observation was due to causes other than those that can be captured in the model. Note that x_5 and s_5 are not involved in $\psi_1(.)$ because e_5 is not along any partially directed path from g_1 and g_4.

The potential functions $\phi_i's$ and ψ_1 now define a joint distribution over the variables:

$$P(X, S) \propto \left[\prod_{i=1}^{5} \phi_i(x_i)\right] \cdot \psi_1(x_1, \ldots, x_4, s_1, \ldots, s_4).$$

where $X = \{x_1, \ldots, x_5\}$ and $S = \{s_1, \ldots, s_5\}$. This probability model is naturally viewed as a *factor graph* and efficient algorithms are available for finding the most likely configurations. Details of the inference algorithms such as max product will be discussed in section 4. In this example the most likely configurations are

$$\begin{aligned} (x_1, \ldots, x_5, s_1, \ldots, s_5) \quad = \quad & (1, 1, 0, 0, 0, +1, +1, *, *, *) \\ & (1, 1, 0, 0, 0, -1, -1, *, *, *) \\ & (0, 0, 1, 1, 0, *, *, +1, +1, *) \\ & (0, 0, 1, 1, 0, *, *, -1, -1, *) \end{aligned}$$

where $*$ indicates that either value is acceptable. The configurations represent the fact that either (e_1, e_2) or (e_3, e_4) must exist with consistent aggregate signs. In this case the configurations corresponding to the situations that both paths explain the knock-out effect (i.e., when $x_1 = x_2 = x_3 = x_4 = 1$) have lower probabilities because of the slight biases arising from the individual potential functions pertaining to the presence of edges: $\phi_i(1) < \phi_i(0)$.

It is worth noting that albeit each edge has the same confidence value from the protein-DNA and protein-protein interaction data, their resulting probabilities can be different. In this example, $x_5 = 0$ in all the most likely configurations. This is because e_5 is not necessary to explain the knock-out effect. More data are available, the more constraints are imposed on the possible configurations. For example, if we conduct the experiment of deleting gene g_3 and find g_4 is down-regulated. This extra evidence can reduce the most likely configurations to

$$(x_1, \ldots, x_5, s_1, \ldots, s_5) \quad = \quad (0, 0, 1, 1, 0, *, *, +1, +1, *)$$

3. PHYSICAL MODELS

Our physical model can be represented as a collection of attributes or variables pertaining to verifiable molecular properties of the biological system such as protein-DNA binding events and formation of protein complexes. The variables need not to be (currently) directly observable and may involve collective properties such as signal transduction pathways. The main requirement is that the variables have to be tied to (in principle) verifiable properties. Any particular setting of such variables gives rise to an annotated physical graph representing interactions that are present. In contrast to dependency models, physical models not only explain observed dependencies but also articulate clear hypotheses about the underlying biological mechanisms.

Our model comprises three parts: an annotated graph representing the set of possible physical interactions, the set of variables whose values determine a physical model, and the construction of a joint distribution over the variables by incorporating observed measurements as evidence. We describe each part in detail with the emphasis on data association.

3.1 Graph representation

Graphs provide a natural representation of possible physical interactions. Here $G = (V, \vec{E}_G \cup \bar{E}_G)$ defines as a directed (possibly cyclic) graph with two types of edges. V is the set of vertices corresponding to genes or their protein products, $\vec{E}_G$ is the set of edges corresponding to possible protein-DNA interactions, and $\bar{E}_G$ is the set of edges denoting possible protein-protein interactions. The directionality of an edge denotes the causal direction along a regulatory pathway. Therefore, the direction of a protein-DNA edge is determined a priori, while the direction of edges in $\bar{E}_G$ cannot be directly determined from protein-protein interaction data. We also allow the possibility of a bi-directional protein-protein edge. In this simple representation we do not distinguish between the DNA sequence, mRNA template, or the protein product of a gene. Two genes g_1 and g_2 can be therefore linked by protein-protein and protein-DNA edges (in either or both directions). In the former case, vertices play the role of protein products whereas in the latter case we refer to the protein binding to the promoter region of the corresponding gene. Note that the interpretation of the pathways is still valid in this collapsed notation so long as we use the implied meaning for the nodes along the pathway.

Without any data all interactions are possible to occur, thus the graph G containing all possible physical interactions should be a complete graph (where there are three edges connecting each pair of vertices). Allowing all possible interactions is computationally burdensome in larger systems. We will restrict the set of possible interactions *a priori*, e.g., by excluding physical interactions without sufficient support in the available data. We demonstrate in section 5 that the modeling results are robust against thresholds used to filter out unlikely protein-DNA interactions.

3.2 Variables

We can annotate the graph G with various biologically meaningful attributes, for example, causal or interaction (sign) directions of edges, delays of edges, latent protein levels of nodes, etc. It is sensible to include only those variables that stand to receive some support either directly or indirectly from the available data. For example, we include activation delays only when relevant time course profiles are available. The variables are associated with features in the physical graphs such as vertices, edges, hyper-edges, paths, or clusters.

Another type of variables are tied to functional processes of the system, for example, whether the expression level changes in a deletion experiment. These variables may not be reduced to individual molecular interactions (they represent active molecular cascades), but are essential for linking physical properties with functional data.

We focus here on a model which incorporates three types of data: location data (protein-DNA interactions), protein-protein interactions, and the mRNA expression levels of gene knock-out experiments. The relevant variables of the regulatory model are in this case:

- $X_{\vec{E}_G} = \{x_{\vec{e}_i} : \vec{e}_i \in \vec{E}_G\}$, a collection of binary (0/1) variables pertaining to the presence or absence of protein-DNA interactions.
- $X_{\bar{E}_G} = \{x_{\bar{e}_j} : \bar{e}_j \in \bar{E}_G\}$, an analogous collection of binary variables denoting whether protein-protein interactions are present
- $S_{\vec{E}_G} = \{s_{\vec{e}_i} : \vec{e}_i \in \vec{E}_G\}$ and $S_{\bar{E}_G} = \{s_{\bar{e}_i} : \bar{e}_i \in \bar{E}_G\}$ which provide the signs $(+1/-1)$ of the interactions represented by the edges.
- $D_{\bar{E}_G} = \{d_{\bar{e}_i} : \bar{e}_i \in \bar{E}_G\}$, a collection of binary variables denoting the directions of protein-protein interactions. The direction of an edge denotes the direction of the corresponding interaction in a signal transduction cascade. For simplicity, we assume here that each edge in $\bar{E}$ has only one direction, essentially reducing the protein-protein interactions to directed edges.
- $K = \{k_{ij} : g_i, g_j \in V\}$ is a collection of the discrete variables of pairwise single knock-out effects whose domains are $\{-1, 0, +1\}$. k_{ij} denotes the effect of knocking out gene g_i on gene g_j. $k_{ij} = -1$ if g_j is down-regulated, $+1$ if g_j is up-regulated, and 0 if g_j is unaffected by the knock-out.
- $\Sigma = \{\sigma_{ija} : k_{ij} \in K, \pi_a \in \Pi\}$ is a collection of binary (0/1) path selection variables, where Π is the set of all valid paths in G (the notion of a valid path will be clarified later). σ_{ija} denotes whether path π_a is active and is the causal explanation of knock-out effect k_{ij}.
- $Y_{\vec{E}_G} = \{y_{\vec{e}_i} : \vec{e}_i \in \vec{E}_G\}$, $Y_{\bar{E}} = \{y_{\bar{e}_j} : \bar{e}_j \in \bar{E}_G\}$ and $O = \{o_{k_{ij}} : k_{ij} \in K\}$ denote the measurement variables of factor binding affinities, protein-protein interactions affinities and gene expression levels of knock-out experiments. Their domains are real numbers and the values are fixed by the data.

3.3 Potential functions

We formalize here how the variables can be tied to the observations through potential functions. The joint distribution over all the variables can be then defined as a product of the potential functions similarly to the toy example.

3.3.1 Potential functions for physical data

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

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

where $\phi_{\vec{e}_i}(x_{\vec{e}_i}; y_{\vec{e}_i})$ is a function of $x_{\vec{e}_i}$ only since the value of $y_{\vec{e}_i}$ remains fixed. The potential function $\phi_{\bar{e}_i}(x_{\bar{e}_i}; y_{\bar{e}_i})$ associated with an undirected edge can be defined analogously. The local nature of these potentials implies that the location measurements are considered to be independent. The form of the conditional probabilities $P(y_{\vec{e}_i}|x_{\vec{e}_i} = 1)$

and $P(y_{\vec{e}_i}|x_{\vec{e}_i}=0)$ permits sensible error models to be used. However, these probabilities are not directly available in current error models and we have to resort to more heuristic transforms of the available p-values.

3.3.2 Potential functions for knock-out data

The knock-out effect k_{ij} is tied to the observed measurement $o_{k_{ij}}$ via a potential function ϕ_{ij} analogously to the protein-DNA and protein-protein interaction data:

$$\phi_{ij}(k_{ij};o_{k_{ij}}) \propto \left[\frac{P(o_{k_{ij}}|k_{ij})}{P(o_{k_{ij}}|k_{ij}=0)}\right]. \quad (2)$$

Each knock-out effect is associated with multiple core attributes in the model. This association amounts to explaining each knock-out effect through a cascade of physical interactions available in the model (here signed protein-DNA and protein-protein interactions). We must first decide what aspects of the knock-outs we attempt to capture. While any significant knock-out effect (a gene is up or down-regulated) can be easily attributed to a cascade of physical interactions, unaffected genes are much more difficult to explain as other causes may be at play. To a first approximation we attempt to explain only likely up/down regulations.

The potential function associated with a knock-out effect k_{ij} reflects the constraint that a cascade in the physical model has to explain k_{ij}. For a path in G to qualify for explaining k_{ij}, the path, denoted here as π, must satisfy:

1. The end nodes of π are g_i and g_j.
2. The last edge in π is a protein-DNA interaction.
3. All the edges in π are in the forward direction (from g_i to g_j).
4. The signs of the edges along π are consistent with the sign of the knock-out effect.
5. The length of π is less than a pre-defined upper bound.
6. If intermediate genes along π have been knocked out, they also exhibit a knock-out effect on g_j.

The first condition manifests the assumption of using a cascade of physical interactions to explain gene regulation. The second condition is based on the accepted assumption that the last step of gene regulation is transcriptional control. The third condition ensures that the path has a causal interpretation. The fourth condition is evident as stated and the fifth one excludes unreasonably long cascades. The last condition requires that each interaction along a path is a necessary component for gene regulation with the exception of missing data. A path which satisfies these conditions is able to explain the knock-out effect k_{ij}. k_{ij} is explained by the physical model if there exists at least one path which satisfies these conditions. These conditions would have to be modified slightly to incorporate the notion of coordinate regulation.

The above conditions impose constraints on the presence and signs of edges as well as the (latent) directions of protein-protein edges. Let $\Pi_{ij}=\{\pi_1,\cdots,\pi_n\}$ denote the paths in G connecting g_i and g_j which satisfy conditions 1, 2, 5, 6 and their protein-DNA edge directions satisfy condition 3. Π_{ij} contain all possible candidate paths which can explain the knock-out effect k_{ij}. Other conditions are encoded into potential functions of variables. Let $\pi_a \in \Pi_{ij}$ be a candidate explanatory path of knock-out effect k_{ij}, $E_a=\{e_a \in \pi_a\}=\vec{E}_a \cup \bar{E}_a$ denote the protein-DNA and protein-protein edges along π_a, $X_a=\{x_e : e \in E_a\}$, $S_a=\{s_e : e \in E_a\}$ be the presence and sign variables of edges along π_a, and $D_a=\{d_e : e \in \bar{E}_a\}$ be the direction variables of protein-protein edges along π_a. Then π_a explains k_{ij} if the following conditions hold:

- $\forall e \in E_a$, $x_e=1$.
- $\prod_{e\in E_a} s_e = -k_{ij}$.
- $\forall e \in \bar{E}_a$, $d_e=\hat{d}_e$ ($\hat{d}_e$ is determined by our definition of path direction values).

The potential function encoding these conditions can be expressed as follows:

$$\psi_{ija}(X_a,S_a,k_{ij}) = \begin{cases} 1 & \text{if } (\bigwedge_{e\in E_a} x_e) \wedge I(\prod_{e\in E_a} s_e = -k_{ij}) \\ & \wedge(\bigwedge_{e\in \bar{E}_a} d_e=\hat{d}_e), \\ \epsilon & \text{otherwise.} \end{cases} \quad (3)$$

where $\wedge$ denotes logical AND and $I(.)$ is the indicator function. The potential function does not vanish even when the constraints are violated so as to allow other causes (those not included in the model) to explain the knock-out effect. Our experimental results are not sensitive to the value of ϵ.

When there are multiple candidate paths connecting g_i and g_j, we require that the conditions along at least one of the paths suffices to explain k_{ij}. Encoding these OR-like constraints in a single potential function is cumbersome since it contains many variables. Alternatively, we introduce auxiliary path selection variables and factorize the potential function into terms corresponding to single paths. Recall that σ_{ija} is the selection variable of path π_a, $\sigma_{ija}=1$ if we use π_a to explain k_{ij}, and $\sigma_{ija}=0$ otherwise. Thus the potential function ψ_{ija} in equation 3 is augmented with variable σ_{ija}:

$$\psi_{ija}(X_a,S_a,k_{ij},\sigma_{ija}) = \begin{cases} 1 & \text{if } (\bigwedge_{e\in E_a} x_e) \wedge I(\prod_{e\in E_a} s_e = -k_{ij}) \\ & \wedge(\bigwedge_{e\in \bar{E}_a} d_e=\hat{d}_e) \wedge I(\sigma_{ija}=1), \\ \epsilon_1 & \text{if } I(\sigma_{ija}=0), \\ \epsilon_2 & \text{otherwise.} \end{cases} \quad (4)$$

We require that $1>\epsilon_1 \gg \epsilon_2$ so that selecting a path that explains the knock-out pair is the most desirable outcome. The value of ϵ_2 is immaterial so long as it is sufficiently small. ϵ_1 can be conceived as the probability that a valid path π_a in general fails to explain the knock-out pair.

We construct a potential function term ψ_{ij}^{OR} to specify the condition that at least one candidate path is selected to explain k_{ij} if k_{ij} is explained. Similar to other potential functions, ψ_{ij}^{OR} is a "soft" or "noisy" version of logical OR:

$$\psi_{ij}^{OR}(\sigma_{ij1},\cdots,\sigma_{ij|\Pi_{ij}|}) = \begin{cases} 1 & \text{if } \bigvee_a \sigma_{ija}=1, \\ \epsilon & \text{otherwise.} \end{cases} \quad (5)$$

where $\vee$ denotes logical OR. Combining equations 3 and 4, the potential function associated with a pairwise knock-out effect is as follows. Denote $E_{ij}=\cup_{\pi_a\in\Pi_{ij}} E_a$, $X_{ij}=\cup_{\pi_a\in\Pi_{ij}} X_a$, $S_{ij}=\cup_{\pi_a\in\Pi_{ij}} S_a$, $D_{ij}=\cup_{\pi_a\in\Pi_{ij}} D_d$, and $\Sigma_{ij}=\{\sigma_{ija} : \pi_a \in \Pi_{ij}\}$, then

$$\begin{aligned} \psi_{ij}^0(X_{ij},S_{ij},D_{ij},\Sigma_{ij},k_{ij}) &= \psi^{OR}(\sigma_{ij1},\cdots,\sigma_{ij|\Pi_{ij}|}) \cdot \\ & \prod_a \psi_{ija}(X_a,S_a,D_a,\sigma_{ija},k_{ij}). \end{aligned} \quad (6)$$

ψ^0_{ij} returns a relatively high value if there exists at least a path which can explain k_{ij} provided selected paths all satisfy the conditions of explanation. Moreover, the returned value is higher if there are more paths which explain the knock-out effect.

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

$$\psi_{ij}(X_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}) = I(k_{ij} \neq 0)\psi^0_{ij}(X_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}) + I(k_{ij} = 0). \tag{7}$$

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

4. INFERENCE OF MODEL ATTRIBUTES

We can combine all the potential functions to define a joint distribution over the core and auxiliary variables in the physical model

$$\begin{array}{l} P(X_{\vec{E}_G}, S_{\vec{E}_G}, X_{\bar{E}_G}, S_{\bar{E}_G}, K, \Sigma; Y_{\vec{E}_G}, Y_{\bar{E}_G}, O_K) \propto \\ \prod_{\vec{e}_i \in \vec{E}_G} \phi_{\vec{e}_i}(x_{\vec{e}_i}; y_{\vec{e}_i}) \cdot \prod_{\bar{e}_j \in \bar{E}_G} \phi_{\bar{e}_j}(x_{\bar{e}_j}; y_{\bar{e}_j}) \cdot \\ \prod_{k_{ij} \in K} \phi_{ij}(k_{ij}; o_{k_{ij}}) \cdot \prod_{k_{ij} \in K} \psi_{ij}(X_{ij}, S_{ij}, \Sigma_{ij}, k_{ij}). \end{array} \tag{8}$$

The potential functions appear as a product in the joint distribution since we assume that individual measurements of physical interactions and knock-out effects (pairs) are independent. This assumption is not always realistic; for example, the readings of adjacent spots on microarrays may be correlated. This is, however, a sensible simplification to start with since adequate models for dependencies in the measurements are not yet available. The potential functions corresponding to different valid paths also appear in a product form. This corresponds to the bias we have introduced to preferably use all the valid paths as explanations for the knock-out effects. Such an "AND bias" corresponds to a product of constraints.

This joint distribution can be viewed as a graphical models such as a Markov random field. It is, however, most naturally viewed as a factor graph [5] similarly to coding problems where the potential function encode (for example) parity check constraints among the bits in the code word. In a factor graph the joint distribution is defined as a product of potential functions (also called factors). For visualization and for structuring inference calculations, a factor graph can be represented as an undirected, bi-partite graph with two types of nodes: variables whose values we are interested in determining, and potential functions (factors) which correspond to (here soft) constraints between the variables. A factor/potential node is adjacent to the variable nodes that are used as arguments in the potential function.

The remaining problem is to find the most likely (MAP) configuration of the values of the variables in the factor graph. The MAP configuration can be computed approximately by the *max-product* algorithm or its various refinements [5, 14, 11]. The properties and optimality guarantees accompanying such algorithms are under active study (see, e.g., [13, 12, 11]).

In brief, the max-product algorithm is a local propagation algorithm analogous to the standard belief propagation algorithm for inference in graphical models. The only difference is that the max-product algorithm seeks to find the most likely configuration of the variables rather than marginal posterior probabilities. The algorithm operates as follows: each node iteratively passes messages to its neighbors, where a message contains the information local to the variable and all previous messages coming to this node (except the one that came previously from the destination node of the message). The algorithm stops when all messages in the graph converge. The message-updating rules are different for variable $\rightarrow$ factor and factor $\rightarrow$ variable messages:

$$\begin{array}{rcl} m_{x \rightarrow f}(x) & = & \prod_{f' \in N(x) \backslash \{f\}} m_{f' \rightarrow x}(x), \\ m_{f \rightarrow x}(x) & = & \max_{\mathbf{X}' = N(f) \backslash \{x\}} [\psi_f(x, \mathbf{X}') \prod_{x' \in \mathbf{X}'} m_{x' \rightarrow f}(x')], \end{array} \tag{9}$$

where $N(.)$ denote the neighbors of a node in the factor graph, f is a factor node, x a variable node, and $\psi_f(.)$ is the potential function corresponding to f. Maximization is carried out over all configurations of variables in $\mathbf{X}'$. The message from a variable node to a factor node is simply the product of other messages incident to the variable (other than the one coming from the factor), while the message from a factor to a variable node is the product of the potential function and incoming messages marginalized in the maximizing sense over all other variables of this factor. The algorithm returns approximate *max-marginals* for all variables which are the products of converged messages incident to each variable node:

$$b_x(x) = \prod_{f \in N(x)} m_{f \rightarrow x}(x). \tag{10}$$

$b_x(x)$ approximates the true *max-marginal probability* given by $P^{\max}_x(x)$ of x:

$$P^{\max}_x(x) = \max_{\mathbf{U} \backslash \{x\}} P(x, \mathbf{U} \backslash \{x\}), \tag{11}$$

where $\mathbf{U}$ stands for all variables in the model. Clearly, $P^{\max}_x(x)$ has a unique maximizing argument iff the MAP configuration is unique, and the MAP configuration is simply found by collecting the maximizing arguments of each max-marginal. The approximate MAP configuration in the max-product algorithm is found analogously by interpreting $b_x(x)$ as true max-marginals.

One possible bottleneck in the max-product algorithm is the evaluation of factor $\rightarrow$ variable messages (equation 8.2). The structure of the potential functions in our setting, however, permits efficient evaluation of these messages. For example, suppose we want to evaluate a message from a single path factor f to an edge presence variable x:

$$m_{f \rightarrow x}(x) = \max_{\mathbf{U} \backslash \{x\}} [f(x, \mathbf{U} \backslash \{x\}) \prod_{y \in \mathbf{U} \backslash \{x\}} m_{y \rightarrow f}(y)]. \tag{12}$$

For $x = 0$, the best scenario is either this path is not selected ($\sigma = 0$), the knock-out effect is not significant ($k_{ij} = 0$), or both conditions hold. The max configurations of other variables are determined by their incident messages but are not dependent on the potential function. For $x = 1$, we need to consider two cases: either the path is not used to explain k_{ij} (the same as $x = 0$ case), or the path is selected and all other variables satisfy the constraints of explanation. The max configuration for $x = 1$ is the supremum of these two cases. This simple deduction greatly reduces the number of enumerations to consider. Moreover, instead of storing the whole lookup table we only need to store distinct return values, since the configurations leading to a particular return value can be deduced from the constraint. This simplification applies to all potential functions composed of simple logical rules such as path explanation and noisy-OR.

In case of more general potential functions, the algorithm may require $O(m \cdot 2^n)$ running time and space, where m is the number of factors and n is the maximum number of arguments in potential functions. A naive implementation of checking each configuration in all potential functions is not possible since a regulatory network can induce relatively large factor graphs. For example, in the subnetwork of pheromone response network given in section 5, there are 46 genes and 67 physical interactions, but its factor graph contains 840 variables, 879 factors and 7436 edges. However, due to the type of constraints we are interested in imposing in the regulatory models, we do not envision this to become a problem in extending the framework.

Another problem arises when constraints do not suffice to uniquely determine all variables. In this case there are multiple MAP configurations which yield equal (or approximately equal) joint probabilities. The max probabilities on the variables on which those MAP configurations differ have degenerate arg max values[3]. For example, if the MAP configurations of (x_1, x_2) are $(0, 1)$ and $(1, 0)$, then both $P^{max}_{x_1}(x_1)$ and $P^{max}_{x_2}(x_2)$ have equal values on $x = 1$ and $x = 0$. We perform an additional recursive search by fixing some of the undetermined variables and running the max-product again over the remaining variables. In a relatively well-constrained case (for example, the phermone response network in section 5), the recursive search is capable of identifying all MAP configurations.

Table 1: Physical interactions on the mating response dataset

protein-DNA:		
(STE12 FUS3)	(STE12 FIG1)	(STE12 TEC1)
(STE12 FUS1)	(STE12 KAR4)	(STE12 GIC2)
(STE12 MFA1)	(STE12 BEM2)	(STE12 STE2)
(STE12 AGA2)	(STE12 MSB2)	(STE12 GPA1)
(STE12 BAR1)	(STE12 YIL169C)	(STE12 FAR1)
(STE12 ASG7)	(STE12 STE6)	(STE12 PRY2)
(STE12 SST2)	(STE12 YMR046C)	(STE12 KAR5)
(STE12 SCW10)	(STE12 MFA2)	(STE12 YNL279W)
(STE12 AGA1)	(STE12 SRL1)	(MCM1 MFA1)
(MCM1 STE2)	(MCM1 AGA2)	(MCM1 GPA1)
(MCM1 BAR1)	(MCM1 YIL169C)	(MCM1 FAR1)
(MCM1 STE6)	(MCM1 MFA2)	(MCM1 AGA1)
(MCM1 SRL1)		
protein-protein:		
(STE2 MFα1)	(STE2 MFα2)	(STE2 STE4)
(STE2 GPA1)	(GPA1 STE4)	(GPA1 SST2)
(GPA1 STE11)	(STE4 STE18)	(STE4 STE5)
(STE4 STE11)	(STE4 FAR1)	(FUS3 STE7)
(FUS3 STE11)	(FUS3 STE5)	(FUS3 STE12)
(STE7 KSS1)	(STE7 STE5)	(STE7 STE11)
(STE11 KSS1)	(STE11 STE5)	(STE11 STE50)
(STE5 KSS1)	(STE12 MCM1)	(STE12 KSS1)
(STE20 STE4)	(STE50 STE5)	(STE50 STE4)
(AGA1 AGA2)	(AGA2 SAG1)	(YIL169C FUS3)

5. EMPIRICAL RESULTS

5.1 Datasets

We evaluated the framework using three datasets in budding yeasts: location analysis data about protein-DNA interactions [6], protein-protein interaction data manually pulled out from the YPD database, and mRNA expression of knockout experiments from the Rosetta compendium data [3]. To simplify the task we focused on genes involved in the pheromone response pathway. We selected protein-DNA interactions and pairwise knock-out effects whose p-values ≤ 0.001. As discussed below, thresholds ranging from 10^{-5} to 0.1 have little effect on the predictive accuracy of the model. When edges with p-values higher than 0.001 are excluded from consideration, the resulting model contains 46 genes, 37 protein-DNA edges, 30 protein-protein edges and 164 pairwise knock-out effects. Tables 1 lists the interactions of the subsystem. To save space we put the data of pairwise knock-out effects in our webpage [4].

The heuristic error model developed by Hughes et al. [3] was applied to location and knock-out data. False positive p-values were derived according to this error model. As a simple (and incorrect) use of the error model, we took the p-values to represent the probabilities P(measurement | interaction does not exist) in location and knock-out datasets. The datasets do not provide information about P(measurement | interaction exists). In this preliminary evaluation, we set these to arbitrary fixed values (0.02) for all confident edges. More reasonable methods of constructing the potential functions for physical evidence may be applied in the future. For instance, Segal et al. constructed P(measurement | interaction exists) from the available error models and used a uniform distribution for P(measurement |interaction does not exist) [9].

Protein-protein interaction data was obtained from the YPD database [5]. The degree of confidence in each interaction is not provided in the database. Here we set the potential functions of all implicated protein-protein interactions to $\phi_i(x) = 2.0I(x = 1) + I(x = 0)$ to reflect the high degree of false positives in the dataset. A slightly more systematic way of setting the potential functions is to incorporate the number of previously verified experiments for each interaction, which is provided in some large-scale protein-protein database such as DIP [6].

5.2 Inferred models

Potential functions for explaining knock-out effects and the joint probability function over these variables were constructed as described previously. Here we restricted the path length ≤ 5. The max-product algorithm was applied to obtain the approximate max-marginal probabilities for each variable. Figure 2 shows the physical subnetwork annotated with attributes which are uniquely determined by these max-marginal probabilities. It is visualized using *cytoscape*, a freeware developed by Ideker et al. [7]. Solid lines correspond to protein-DNA and dash lines represent protein-protein interactions. The direction of protein-DNA arrows

[3]We are assuming here for simplicity that the max marginals are correct.

[4]http://www.ai.mit.edu/people/chyeang/koeffects.txt

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

[6]http://dip.doe-mbi.ucla.edu/

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

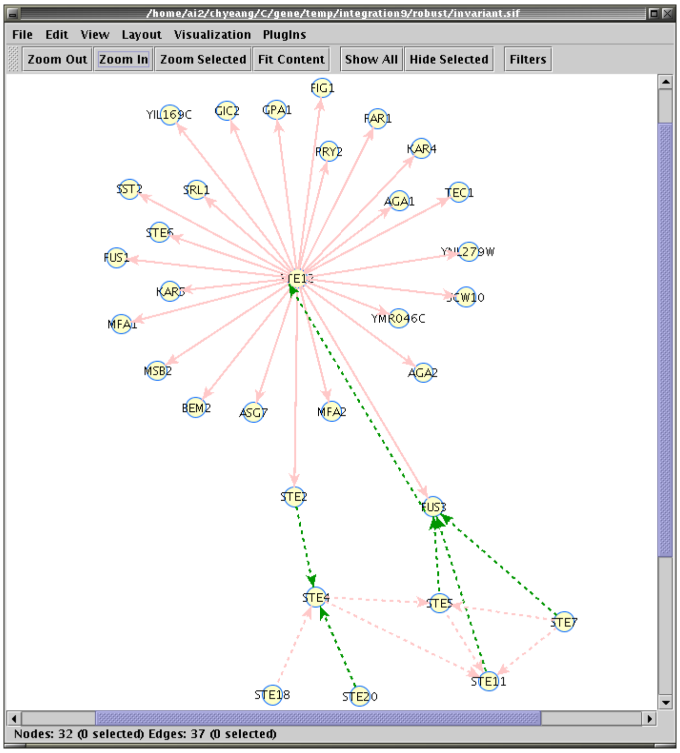

Figure 2: Uniquely determined network attributes

are given in the data, while the arrows (and the existence) of protein-protein edges are inferred from the model. Edge signs are color-coded with light red (positive) and dark green (negative).

It can be seen first that most protein-DNA edges emanating from STE12 have positive signs. This is consistent with previous studies that STE12 is the activator of mating response genes. Second, the inferred directions of protein-protein interactions (STE5,STE11), (STE7,STE11), (STE11,FUS3), (STE11,KSS1), (FUS3, STE12) and (KSS1,STE12) agree with the direction of signal transduction pathway of pheromone response[8]: STE12 is a transcription factor, FUS3 and KSS1 are MAP kinases, STE11 MAP kinase kinase, etc. We note that these ordering relations cannot be retrieved by inspecting the expression levels of kinases or transcription factors, since knocking out an upstream gene changes their protein modification states rather than mRNA or protein abundance.

One incongruence with previous studies about the mating pathway is the sign of (FUS3,STE12). Our model claims that the edge should be negative but previous studies indicate that FUS3 activates STE12 by phosphorylation [?]. Our prediction is based on the fact that many genes down-regulated in ΔSTE12 experiment are up-regulated in ΔFUS3 experiment. Therefore, if we use the path (FUS3,STE12) (STE12,g) to explain these effects the sign of (FUS3,STE12) must be negative. It is certainly possible that such up-regulation is mediated via other pathways outside the sub-network being considered here. In the absence of any additional data our prediction for the sign of the edge (FUS3,STE12) remains incorrect.

[8]http://mips.gsf.de/proj/yeast/CYGD/db/index.html

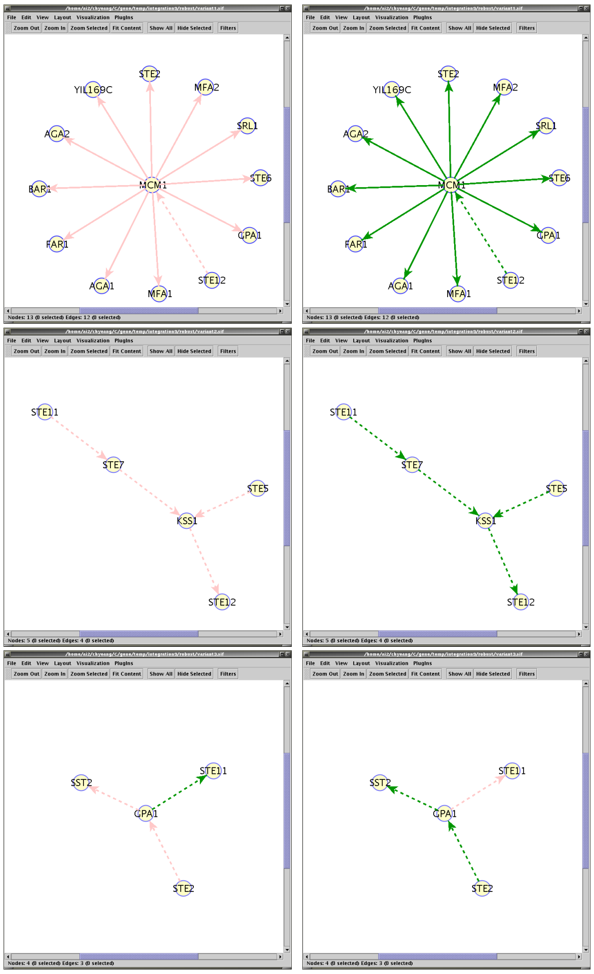

Figure 3: Degenerate MAP configurations on sub-networks

To instantiate other variables without unique maxima in their max-marginals, we then apply the recursive search procedure described earlier. In the pheromone response subnetwork, the data constrain the model sufficiently well so that we can enumerate all remaining MAP configurations from the recursive search. There are only 8 MAP configurations in our example. The degeneracy occurs only at the edge sign variables, and these configurations can be expressed as products of the subconfigurations of three small networks shown in Figure 3. Each small network has two subconfiguartions (corresponding to the possible overall signs) and we can pick up the subconfigurations of each small network independently.

Subnetwork 1 reflects the ambiguity of the sign of protein-protein interaction (STE12,MCM1). Many genes are bound jointly by STE12 and MCM1. Since ΔMCM1 experiment is unavailable (in fact deleting MCM1 is lethal for yeast), we speculate that both paths (STE12,g) and (STE12,MCM1)(MCM1,g) are active regulatory pathways. Thus the product of signs of (STE12,MCM1) and (MCM1,g) is fixed while the individual signs are not.

5.3 Explaining knock-out data

How well can our model predict knock-out data? We start

Table 2: Cross validation on knock-out pairs

# hold-outs	# trials	% error
1	136	0.74 %
5	500	0.68 %
20	200	12.22 %

by checking whether the model is sufficiently flexible to explain the available knock-out data. There are 136 knock-out pairs which are connected via candidate paths in the physical network. It turns out all the 136 pairs are captured by all the 8 MAP configurations obtained from the max-product algorithm and the iterative search. By explanation we mean for each path which is selected according to the MAP configuration ($\sigma_a = 1$), the variables satisfy conditions 1-6 in section 3.4.

We use cross-validation to evaluate the predictive accuracy of the model. In other words, we randomly hold out a fixed number of knock-out pairs when constructing the joint distribution and running the inference algorithm. The resulting MAP configurations are used to gauge whether the model explains the held-out knock-out pairs. For the model to "explain" a held-out knock-out effect, each MAP configuration (a realization of the physical graph) must have all the signs in the valid paths consistent with the knock-out effect. This measure is conservative. Table 2 shows the results of leave-n-out cross validation, where n equals to 1, 5, and 20. The results indicate that the algorithm can predict the knock-out effects with high degree of accuracy. This is to be expected since the information about a knock-out interaction is distributed among multiple interactions along pathways. In contrast, if we systematically hide all effects regarding a particular knock-out experiment (which there are relatively few), then the small number of other available knock-out experiments no longer suffices to constrain the variables enough to predict the effects.

5.4 Sensitivity analysis

The cross validation results are encouraging in terms of being able to predict pairwise knock-out effects given the constraints from other knock-out interactions. However, this outcome might be an artifact of a particular setting of the model parameters/thresholds. We provide here a sensitivity analysis to illustrate that this is not the case.

We consider the following adjustable parameters: the maximum length of candidate paths, thresholds on p-values of location and knock-out expression data for selecting possible protein-DNA interactions and knock-out pairs to explain, and the error probabilities used as soft constraints in the potential functions (ϵ_1 in the definition of ψ_{ija}). Figure 4 shows the leave-one-out test accuracy rates across a wide range of these parameters. The test accuracy here is normalized by the number of knock-out effects that the inferred model can in principle explain (this is a function of the number of edges they contain). The default values of these parameters are: location and knock-out p-value thresholds $= 10^{-3}$, $\epsilon_1 = 0.7$, $\epsilon_2 = 0.299$, and the maximum path length $= 5$. Robustness tests are carried out by varying one parameter and fixing all others at their default values. Accuracy rates are evaluated by dividing the number of correct predictions by the number of knock-out pairs connected via valid paths. It is clear that test errors are very robust against

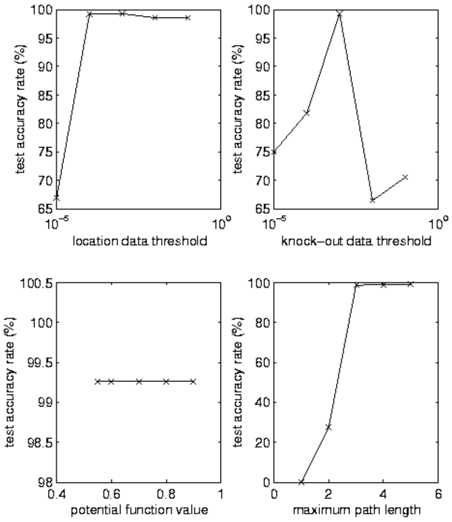

Figure 4: Sensitivity analysis on test accuracy

the location p-value threshold and returned potential function values, and moderately robust against the knock-out p-value threshold. If path length < 3 then the model can hardly predict knock-out effects accurately. This is because short paths can receive very few (or no) constraints from other knock-out pairs. Test errors become robust when the maximum path length ≥ 3.

6. DISCUSSION

There are several ways in which we will extend our framework. We provide here a few examples. Biological experiments are typically costly and time consuming. Systematizing the experimental effort with the help of computational techniques is important. In the framework of physical models, new experiments can serve two major purposes. The existing datasets are unlikely to impose sufficient constraints to yield a unique physical interpretation. New experiments become necessary in order to further distinguish between degenerate models. We can view a model as a system which responds to inputs (environmental or internal perturbations) by producing a set of observable outputs. We would like to perturb the system to make the predicted responses of alternative interpretations of the physical graphs as different as possible. In addition, new experiments can also verify or falsify existing (unique) models. Any inferred interactions not yet known to exist can be verified experimentally. Moreover, we can use gene knock-outs to fill in information along each explaining cascade about the knock-out effects of the intermediate genes.

All the physical interactions in the examples given above have been pairwise interactions. In an actual biological system, interactions may involve multi-protein complexes (e.g., holoenzyme in yeast) or coordinated binding of multiple proteins. We can use a hyper-graph to represent the regulatory network involving this type of multi-way interactions. As before we can incorporate variables specifying the presence or absence of hyper-edges. We can also generalize the notion of the edge sign to a hyper-edge. Here the "sign" specifies instead a combinatorial (logic) function (e.g., AND) of how coordination is required for a regulatory effect. This approach can be particularly useful in interpreting double knock-out experiments.

We can also incorporate other types of functional data to further constrain the model. Unlike knock-out expression data in which causes (the deleted genes) and effects (the affected genes) are clear, causal relations are often difficult to resolve in most expression datasets. In time course profiles, however, the order of the measurements does restrict possible causal interpretations. We can incorporate time course profiles as evidence in our framework both in terms of trying to infer additional attributes (time lags of interactions) as well as to explain appropriately chosen time lag correlations on the basis of common ancestors in the physical graph. Such association of observations to sets of variables is analogous to the knock-out case.

7. CONCLUSION

We have developed a new framework for inferring genetic regulatory networks from multiple sources of data. Our approach differs from many previous methods (statistical dependency models) in terms of requiring readily interpretable and verifiable models of underlying biological mechanisms. Our experimental results are encouraging. Inferred models on a subnetwork of yeast mating pathway are shown to conform previous studies in several aspects. Cross validation experiments on a reduced regulatory subsystem indicate that the presence, direction and sign of protein-DNA and protein-protein interactions can be accurately predicted under this framework. Sensitivity analysis on several free parameters also suggests inferred models are robust against particular settings of these parameters. The framework can be naturally extended to model other characteristics of the regulatory network such as coordinated effect of multiple transcription factors or even to resolve hidden causes of responses to environmental perturbations.

8. ACKNOWLEDGEMENT

The authors are grateful for the discussions with our colleagues from MIT Whitehead Institute and Artificial Intelligence Laboratory: Dr. Trey Ideker, Owen Ozier, and Professor Richard Young, Professors David Gifford and Tomas Lozano-Perez. We also thank to professor Richard Young's lab at Whitehead Institute for providing the location analysis data. Tommi Jaakkola acknowledges support from the Sloan foundation in the form of the Sloan Research Fellowship. The work was also partially funded by grants from DARPA and NIH.

9. REFERENCES

[1] B. Frey and D. MacKay. A Revolution : Belief Propagation in Graphs with Cycles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 479–485, 1997.

[2] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Combining Location and Expression Data for Principled Discovery of Genetic Regulatory Network Models. In *PSB Proceedings*, 2002.

[3] T. Hughes, M. Marton, A. Jones, C. Roberts, R. Stoughton, C. Armour, H. Bennett, E. Coffey, H. Dai, Y. He, M. Kidd, A. King, M. Meyer, D. Stade, P. Lum, S. Stepaniants, D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. Friend. Functional Discovery via a Compendium of Expression Profiles. *Cell*, 102:109–126, 2000.

[4] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, J. Buhler, J. Eng, R. Bumgarner, D. Goodlett, R. Aebersold, and L. Hood. Integrated Genomic and Proteomic Analysis of a Systematically Perturbed Metabolic Network. *Science*, 292:929–934, 2001.

[5] F. Kschischang, B. Frey, and H. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[6] T. Lee, N. Rinaldi, F. Robert, D. Odom, Z. Bar-Joseph, G. Gerber, N. Hannett, C. Harbison, C. Thompson, I. Simon, J. Zeitlinger, E. Jennings, H. Murray, D. Gordon, B. Ren, J. Wyrick, J. Tagne, T. Volkert, D. G. E. Fraenkel and, and R. Young. Transcriptional Regulatory Networks in *Saccharomyces Cerevisiae*. *Science*, 298:799–804, 2002.

[7] D. Lohr, P. Venkov, and J. Zlatanova. Transcriptional Regulation in Yeast GAL Gene Family: a Complex Genetic Network. *FASEB Journal*, 9:777–787, 1995.

[8] B. Ren, F. Robert, J. Wyrick, O. Aparicio, E. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. Volkert, C. Wilson, S. Bell, and R. Young. Genome-wide Location and Function of DNA-binding Proteins. *Science*, 290:2306–2309, 2000.

[9] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From Promoter Sequence to Expression: A Probabilistic Framework. In *RECOMB Proceedings*, pages 263–272, 2002.

[10] P. Uetz, L. Giot, G. Cagney, T. Mansfield, R. Judson, J. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J.M.Rothberg. A Comprehensive Analysis of Protein-Protein Interactions in *Saccharomyces Cerevisiae*. *Nature*, 403:623–627, 2000.

[11] M. Wainwright, T. Jaakkola, and A. Wilsky. Exact MAP Estimates by (Hyper)Tree Agreement. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[12] Y. Weiss and W. Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Computation*, 13:2173–2200, 2001.

[13] Y. Weiss and W. Freeman. On the Optimality Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs. *IEEE Transactions on Information Theory*, 47:736–744, 2001.

[14] J. Yedidia, W. Freeman, and Y. Weiss. Generalized Belief Propagation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, 2000.