

Understanding Protein-protein Interaction Networks

Thesis submitted for the degree of
“Doctor of Philosophy”

by
Ariel Jaimovich

Submitted to the Senate of the Hebrew University
September 2010

Supervised by
Prof. Hanah Margalit and Prof. Nir Friedman

Abstract

All living organisms consist of living cells and share basic cellular mechanisms. Amazingly, all those cells, whether from a bacterium or a human being, although different in their structure and complexity, comprise the same building blocks of macro molecules: DNA, RNA, and proteins. Proteins play major roles in all cellular processes: they create signaling cascades, regulate almost every process in the cell, act as selective porters on the cell membrane, accelerate chemical reactions, and many more. In most of these tasks, proteins work in concert, by creating complexes of varying sizes, modifying one another and transporting other proteins. These interactions vary in many aspects: they might take place under specific conditions, have different biophysical properties, different functional roles, etc. Identifying and characterizing the full repertoire of interacting protein pairs are of crucial importance for understanding the functionality of a living cell. In the last decade, development of new technologies allowed large-scale measurements of interaction networks. In turn, many studies used the results of such assays to gain functional insights into specific proteins and specific pathways as well as learn about the more global characteristics of the interaction network. Unfortunately, the experimental noise in the large-scale assays makes such analyses hard, challenging the development of advanced computational approaches towards this goal.

In the first part of my PhD work I used the language of relational graphical models to suggest a novel statistical framework for representing interaction networks. This framework enables taking into account uncertainty about the observed large-scale measurements, while investigating the interaction network properties. Specifically, it allows simultaneous prediction of all interactions given the results of large-scale experimental assays. I applied this model to noisy observations of protein-protein interactions and showed how such simultaneous predictions enable intricate information flow between the interactions, allowing for better prediction of missing information. However, application of this model to the entire interaction network would require creation of a huge model over millions

of interactions. Thus, I turned to develop tools that will allow realistic representation of such models over very large interaction networks and also enable efficient computation of approximate answers to probabilistic queries. Such tools facilitate learning the properties of these models from experimental data, while taking into account the uncertainty arising from experimental noise. Importantly, I created a code framework to allow efficient implementation of these (and other) algorithms, and devoted a special effort to provide an implementation of such ideas to general models. To date this library has been used in a wide variety of applications, such as protein design algorithms and object localization in cluttered images. The last part of my PhD research addressed genetic interaction networks. In this work, together with Ruty Rinott, we showed how analysis of the network properties leads to novel biological insights. We devised an algorithm that used data from genetic interactions to create an automatic organization of the genes into functionally coherent modules. Next, we showed how using additional information on genetic screens performed under a range of chemical perturbations sheds light on the cellular function of specific modules. As large-scale screens of genetic interactions are becoming a widely used tool, our method should be a valuable tool for extracting insights from these results.

Contents

1	Introduction	1
1.1	Protein-protein interactions	1
1.1.1	Large-scale identification of protein-protein interactions .	2
1.1.2	Genetic interactions as a tool to decipher protein function .	5
1.1.3	From interactions to networks	7
1.1.4	Uncertainty in interaction networks	9
1.2	Probabilistic graphical models	9
1.2.1	Markov networks	10
1.2.2	Approximate inference in Markov networks	12
1.2.3	Relational graphical models	14
1.3	Research Goals	15
2	Paper chapter: Towards an integrated protein-protein interaction network: a relational Markov network approach	17
3	Paper chapter: Template based inference in symmetric relational Markov random fields	38
4	Paper chapter: FastInf - an efficient approximate inference library	48
5	Paper chapter: Modularity and directionality in genetic interaction maps	53
6	Discussion and conclusions	63
6.1	Learning relational graphical models of interaction networks	63
6.2	Lifted inference in models of interaction networks	66
6.3	<i>In-vivo</i> measurements of protein-protein interactions	68
6.4	Analysis of genetic interaction maps	68
6.5	Implications of our methodology for analysis of networks	72
6.6	Integration of interaction networks with other data sources	74

6.7	Open source software	74
6.8	Concluding remarks	75

1 Introduction

In the last couple of decades, large-scale data have accumulated for many types of interactions, varying from social interactions through links between pages of the world wide web and to various types of biological relations between proteins. Visualization of such data as networks and analysis of the properties of these networks has proven useful to explore these complex systems (Alon, 2003; Yamada and Bork, 2009; Boone et al., 2007; Handcock and Gile, 2010). In this thesis I will concentrate on analysis of protein-protein interaction networks, introducing novel methods that should be valuable also for the analysis of different kinds of networks.

1.1 Protein-protein interactions

All living organisms consist of living cells and share basic cellular mechanisms. Amazingly, all those cells, whether from a bacterium or a human being, although different in their structure and complexity, comprise the same building blocks of macromolecules: DNA, RNA, and proteins.

Protein sequences are encoded in DNA and synthesized by a well known pathway that is often referred to as the *central dogma of molecular biology* (Figure 1). It states that the blueprint of each cell is encoded in its DNA sequence. The DNA is replicated (and so the blueprint is passed on to the cell's offsprings), and part of it is transcribed to messenger RNA (mRNA) which is, in turn, translated into proteins. After translation of a mRNA to a protein there are still many processes the protein has to undergo before it is functional. First it has to fold into the correct three dimensional structure, then it has to be transported to a specific cellular localization, and often it has to undergoes specific modifications. There are many types of proteins in a cell (e.g., 6000 proteins types in the budding yeast *Saccharomyces cerevisiae*) and each of them can be expressed in one copy or in thousands of copies (Ghaemmaghami et al., 2003). The expression levels are tightly regulated, and thus similar cells can have drastically different sets of expressed

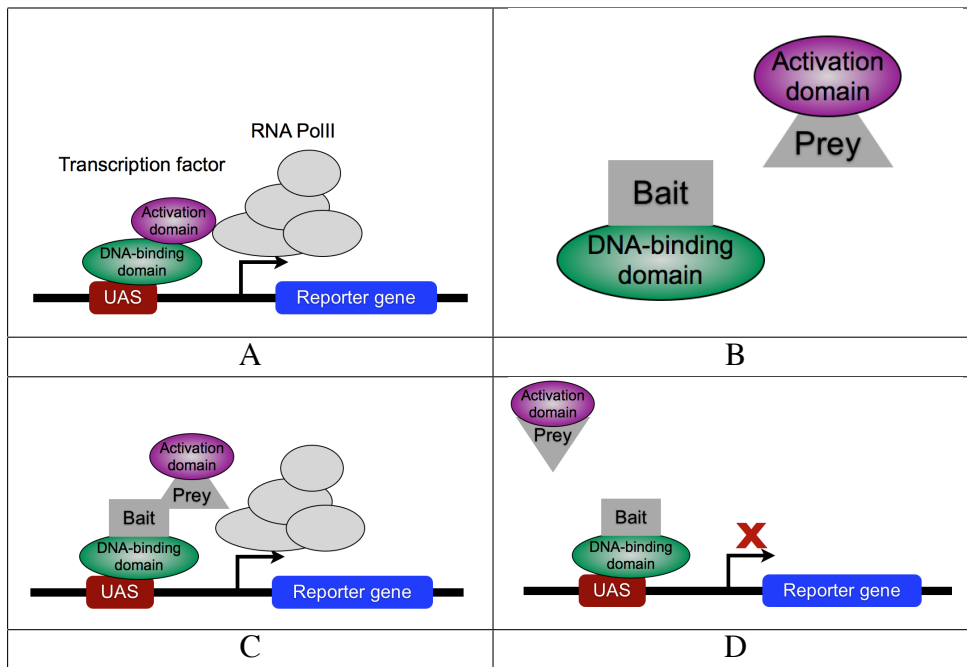


Figure 2: Identifying protein-protein interactions using the yeast two-hybrid method: (A) A Transcription factor has a binding domain and an activation domain. (B) The bait protein is fused to the DNA binding domain and the prey protein is fused to the transcription activation domain (C) If the proteins interact, RNA polymerase is recruited to the promoter and the reporter gene is transcribed (D) If there is no physical interaction, the reporter gene is not transcribed.

manner was a high throughput adaptation of the yeast two-hybrid method (Uetz et al., 2000; Ito et al., 2001). In this method a DNA binding domain is fused to a 'bait' protein and a matching transcription activation domain is fused to a library of 'prey' proteins (Figure 2). Each time one pair of specific bait and prey proteins are introduced into a yeast cell using standard yeast genetics techniques. In turn, if the bait and prey proteins physically interact, they enable transcription of a reporter gene. By using a selection marker as a reporter gene, this method can be used to easily identify interacting proteins by searching for combinations of bait and prey proteins whose introduction to a yeast strain results in a viable yeast colony.

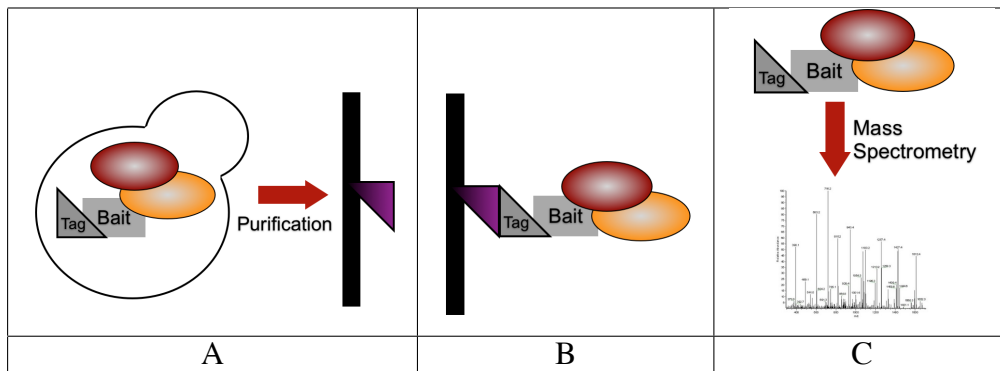


Figure 3: Identifying protein-protein interactions using affinity purification: (A) A bait protein (grey rectangle) is attached to a tag (grey triangle) and introduced into a yeast cell. (B) The tag is used to capture the protein. (C) Mass spectrometry is used to identify the proteins that were captured with the bait proteins (orange and red ellipses).

Although small scale applications of this method yielded functional insights into specific biological mechanisms (Rodal et al., 1999; Jensen et al., 2000), the small overlap between the results of the first large-scale screens that used this approach raised questions regarding the accuracy and sensitivity of this method (von Mering et al., 2002; Sprinzak et al., 2003). However, more recent works claim that the small overlap in the results is due to the low sensitivity of the method, and that comparable results can be obtained when each bait protein is tested a number of times against the prey library (Yu et al., 2008).

Another type of large-scale assay measuring protein-protein interactions uses an affinity purification approach. In this method each prey protein is fused to a Tandem Affinity Purification (TAP) tag and introduced to the yeast cell. In turn, the tag is used to purify the protein with all its interaction partners. Finally, mass spectrometry is used to identify the interaction partners that were captured with the bait protein (Figure 3 ; Rigaut et al. (1999)). In contrast to the yeast two-hybrid method, which measures binary interactions that do not necessarily create stable complexes, this method is aimed towards measurement of stable protein complexes. This approach was carried out initially on a relatively small set of

tagged proteins (Gavin et al., 2002; Ho et al., 2002) and later on a much broader set covering almost the entire yeast proteome (Gavin et al., 2006; Krogan et al., 2006). Again, small overlap between the results of such screens raised concerns regarding the reliability of their experimental results. However, later studies showed that improved analysis can reduce some of this noise and integrated the results of both screens to yield a predicted set of yeast complexes (Collins et al., 2007a). In the discussion section I will describe newer methods that assay protein-protein interaction at a proteomic scale and their implications.

With the accumulation of large-scale information on protein-protein interactions, many studies tried to infer the function of unannotated proteins using a "guilt by association" approach (Galperin and Koonin, 2000; Marcotte et al., 1999b; Deng et al., 2004; Zhao et al., 2008). The basic logic behind these methods is that we can learn about the function of an uncharacterized protein by looking at the known functions of its interaction partners.

1.1.2 Genetic interactions as a tool to decipher protein function

Another, indirect, type of information regarding functional relations between proteins arises from genetic screens. In such screens the phenotypic effect of knocking out target genes is measured under the genetic background of the knockout of a query gene. In turn, if the mutation suppresses or amplifies the effect of the query gene knockout, these two genes are deduced to be functionally related (Boone et al., 2007).

The first large-scale genetic interactions screens focused on searching *synthetic lethal* interactions. Those are the most drastic manifestation of genetic interactions that occur when both single perturbations yield viable strains but the double knockout is lethal (Tong et al., 2004). More recently, a more quantitative measure was developed to determine the sign and strength of the genetic interaction (Schuldiner et al., 2005; Collins et al., 2006; Costanzo et al., 2010). In these methods the growth rate of the double knockout strain is compared to the expected growth rate assuming an additive effect of both single knockouts. If the growth

	w.t	ΔX	ΔY	ΔXY	Possible pathway	Observation	Genetic interaction
a						$\Delta XY > \Delta X \Delta Y$	Alleviating genetic interaction
b						$\Delta XY = \Delta X \Delta Y$	No genetic interaction
c						$\Delta XY < \Delta X \Delta Y$	Aggravating genetic interaction

Figure 4: **Schematic illustration of the possible results of a genetic interaction assay:** In all panels the first four columns illustrate a yeast colony (in grey) on a petri dish for different genetic backgrounds. The size of the yeast colony is a proxy to its growth rate. The fifth column describes possible pathways that can result in such observations. The sixth column shows the formulation of each type of interaction and the seventh column shows the resulting annotation. (a) An example for an alleviating genetic interaction. (b) An example of two genes that do not have a genetic interaction. (c) An example of two genes that have an aggravating genetic interactions.

rate is faster than expected given the two single knockouts the genetic interaction is termed an alleviating interaction. This effect can be caused by two genes that belong to the same pathway (Figure 4 a). If the growth rate is slower than expected then the genetic interaction is termed an aggravating interaction. This effect can be caused, for example, by two proteins participating in two parallel pathways that lead to the same product (Figure 4 c). Here each single knockout has almost no effect since the alternative pathway compensates for the missing protein. However, a double knockout that eliminates both pathways will create a growth defect that could not be expected given the two single knockouts.

1.1.3 From interactions to networks

In addition to providing many insights into the function of specific proteins, the accumulation of such large-scale data on cellular interactions raised more global questions regarding the *network* of interactions. The seminal work of Erdos and Renyi (1960) laid the basis for many of the modern works that model large-scale interaction networks in many fields, suggesting a random graph model in which each two nodes are connected with an edge with probability p . Barabasi and Albert (1999) showed that the degree distribution of many biological networks (including the protein-protein interaction network) does not fit a Poisson distribution, as would be expected for the random network model of Erdos and Renyi. Furthermore, they showed that the degree distribution in such biological networks does fit a power law distribution, where most of the proteins interact with a small number of partners, and a few proteins (termed *hubs*) interact with a large number of partners. Such networks are called *scale free* networks because of the behavior of their degree distribution. Another property that was observed in biological networks is that any two nodes in the network are connected through a very short path (Cohen and Havlin, 2003). This network property is called *small world*, originating from its implication on social networks. Finally, Barabasi and Oltvai (2004) showed how evolutionary principles of gene duplication and preferential attachment can result in networks with these properties.

These network properties have also biological implications, as they characterize networks that are more robust to random perturbations. That is, a random attack is more probable to perturb a node with a small number of interactions, and might have little effect on the performance of the entire network. In support of this theory many works show that the essential proteins are *hubs* in the network (Jeong et al., 2001). Later, Han et al. (2004) showed that these *hub* proteins can be divided to two major types according to the expression patterns of their interaction partners. The first type (termed *date hubs*) corresponds to proteins whose interaction partners are expressed in different times, and thus are assumed to create many pairwise interactions, each time with a different partner. On the other

hand, proteins hubs which are expressed simultaneously with all their interaction partners (termed *party hubs*) are assumed to create complexes that act together to perform certain tasks in the cell.

In recent years, concerns were raised regarding this type of network analysis based on both the quality of the data used in the analysis and also on the quality of the statistical validations. For example, Lima-Mendez and van Helden (2009) claim that the good fits of the scale free and small world models result from sampling artifacts or improper data representation. Concerns were raised also regarding the distinction between date and party hubs. Batada et al. (2006) claim that this distinction might be a reflection of the small datasets used in the analysis of Han et al. (2004). This topic is still under scrutiny, as Bertin et al. (2007) claims that repeating the same analysis on larger datasets validates the distinction between the two type of hubs while Batada et al. (2007) claims that this newer analysis is not controlled for the proper confounding factors.

Another work that tried to infer biological insights from the network properties looked for recurring connected patterns that appear in the network more than expected at random, termed *network motifs* (Shen-Orr et al., 2002; Milo et al., 2002). This method was initially applied to transcription regulation networks in bacteria, reporting on the feed forward loop as a predominant network motif in this transcription regulation network (Shen-Orr et al., 2002). Although concerns were raised regarding the limitations of this identification method (Artzy-Randrup et al., 2004), further analysis demonstrated both experimentally and computationally the functional and biological meaning of these motifs in specific cellular pathways (Mangan et al., 2003, 2006; Kaplan et al., 2008). Furthermore, Milo et al. (2004) showed that different networks have different sets of overrepresented motifs, which can result in different global properties. In addition, similar strategies applied to networks combining various types of interactions (Yeager-Lotem et al., 2004; Zhang et al., 2005) showed how such analysis can be used to provide a useful simplification of complex biological relationships.

1.1.4 Uncertainty in interaction networks

Identifying and characterizing the full repertoire of protein-protein interactions is one of the main challenges in this field. However, it was shown that since most of the large-scale assays contain noisy observations, one has to integrate information from a number of screens in order to produce reliable predictions of such interactions (von Mering et al., 2002; Sprinzak et al., 2003). As a result, many methods have tried to take into account the results of the experimental results described above, along with those of computational assays (Sprinzak and Margalit, 2001; Pellegrini et al., 1999) into one integrated prediction (Jansen et al., 2003; Bock and Gough, 2003; Zhang et al., 2004; Jensen et al., 2009). Most of these methods predict each of the protein-protein interactions independently of the others. However, the analysis of network properties makes it clear that such independent prediction ignores information arising from the properties of the network. For example, the over-representation of 3-cliques in the protein-protein interaction network (Yeager-Lotem et al., 2004) means that if we know that a protein x physically interacts with proteins y and z , this raises our prior belief regarding an interaction between y and z .

On the other hand, the majority of the studies that analyze the network properties ignore the uncertainty regarding the interaction data by trying to use one relatively reliable source of information. This usually results in a compromise between the coverage of the data and its reliability. In my PhD work, using the language of probabilistic graphical models, I tried to bridge this gap by dealing with uncertainty in the data *while* learning about the properties of the network.

1.2 Probabilistic graphical models

Probabilistic graphical models provide a framework for representing a complex joint distribution over a set of n random variables $\mathcal{X} = \{X_1 \dots X_n\}$. Even in the case of discrete binary random variables, naive representation of such a distribution requires specification of the probability for each of the 2^n assignments.

The main premise of these models is that we can take advantage of conditional independence properties of the joint distribution to yield an efficient representation that will enable efficient performance of various tasks, such as probabilistic inference and learning.

In the early 1970's using the Hammersley-Clifford theorem, Besag (1974) introduced the notion of *Markov random fields* that related between the conditional independence properties and the graph structure for lattice models. Later, Frank and Strauss (1986) applied similar ideas to general log-linear models which they called *Markov networks*. The seminal work of Pearl (1988) laid the foundation for the generalization of directed versions of these models, termed *Bayesian networks*. The common idea in all these models is that a qualitative graph structure, in which each random variable is denoted by a node, specifies the set of conditional independencies assumed by the model. In addition, a set of quantitative parameters specifies the actual distribution.

In the last 20 years many works suggested new forms of models (Buntine, 1995; Murphy, 2002; Lafferty et al., 2001), improved exact and approximate inference techniques (Yuille and Rangarajan, 2002; Wierginck and Heskes, 2003; Wainwright et al., 2005b; Yedidia et al., 2005; Chavira et al., 2006), and developed methods to estimate the parameters and structure of the model from noisy observations. Graphical models have been successfully used in many types of applications, ranging from medical expert systems (Heckerman and Nathwani, 1992) through error correcting codes (Kschischang et al., 2001) and analysis of gene expression data (Friedman et al., 2000; Segal et al., 2004) to image analysis (Shotton et al., 2006).

1.2.1 Markov networks

In this dissertation I use the undirected version of graphical models, called Markov networks (or sometimes Markov Random Fields). One standard way to describe such models, is using Factor Graphs (Figure 5; Kschischang et al. (2001)) that contain a bipartite graph between two kinds of nodes:

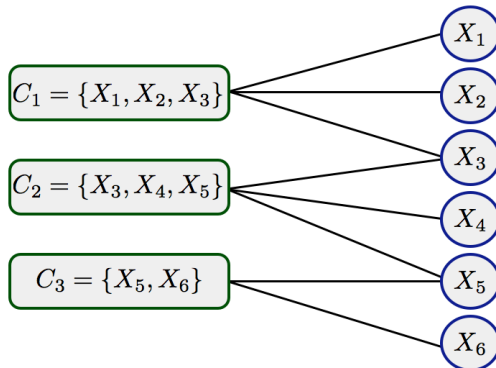


Figure 5: **Factor graph**: factor nodes depicted by squares shown on the left. Variable nodes, depicted as circles, shown on the right.

- *Variable nodes* that depict random variables, shown as circles.
- *Factor nodes* that define groups of variables, shown as rectangles.

Given a joint distribution over n random variables ($\mathcal{X} = \{X_1, \dots, X_n\}$) this qualitative graph implies a decomposition of the joint distribution into a product of local terms:

$$P(\mathcal{X}; \Theta) = \frac{1}{Z(\Theta)} \exp \left(\sum_i \theta_i(X_i) + \sum_\alpha \theta_\alpha(X_\alpha) \right), \quad (1)$$

where X_i and X_α are subsets of \mathcal{X} defined by the variable and factor nodes in the graph, and Θ is the set of quantitative parameters (potential functions) that specify the distribution. We denote by $\theta_i(X_i)$ and $\theta_\alpha(X_\alpha)$ the specific parameter matching the assignment of X_i and X_α , respectively. The normalization term, $Z(\Theta)$, is called the *partition function*:

$$Z(\Theta) = \sum_{\mathcal{X}} \exp \left(\sum_i \theta_i(X_i) + \sum_\alpha \theta_\alpha(X_\alpha) \right).$$

1.2.2 Approximate inference in Markov networks

Inferring the *marginal probabilities* and *likelihood* in graphical models are critical tasks needed both for making predictions and for facilitating learning. The *marginal distribution* of a group of k variables X_{i_1}, \dots, X_{i_k} is denoted by μ_{i_1, \dots, i_k} and defined as:

$$\begin{aligned}\mu_{i_1, \dots, i_k} &= p(X_{i_1}, \dots, X_{i_k}) \\ &= \sum_{\mathcal{X} \setminus \{X_{i_1}, \dots, X_{i_k}\}} p(\mathcal{X}).\end{aligned}$$

The *likelihood function* of M independent observations over \mathcal{X} (denoted as $\mathcal{X}[1], \dots, \mathcal{X}[M]$) for a given parameterization Θ is given by:

$$\begin{aligned}p(\mathcal{X}[1], \dots, \mathcal{X}[M]; \Theta) &= \prod_{m=1}^M p(\mathcal{X}[m]; \Theta) \\ &= \prod_{m=1}^M \frac{1}{Z(\Theta)} \exp\left(\sum_i \theta_i(X_i[m]) + \sum_\alpha \theta_\alpha(X_\alpha[m])\right).\end{aligned}$$

Where $X_i[m]$ and $X_\alpha[m]$ are the assignments for the appropriate variables in the m 'th observation.

Computing exact answers to these inference queries is often infeasible even for relatively modest problems. Thus, there is a growing need for inference methods that are both efficient and can provide reasonable approximate computations. The Loopy Belief Propagation (LBP, Pearl, 1988) algorithm has gained substantial popularity in the last two decades due to its impressive empirical success, and is now being used in a wide range of applications ranging from transmission decoding to image segmentation (Murphy and Weiss, 1999; McEliece et al., 1998; Shental et al., 2003). The seminal work of Yedidia et al. (2005) has added theoretical support to the loopy belief propagation algorithm by posing it as a variational inference method. The general variation principle phrases the inference problem

as an optimization task:

$$\log Z(\Theta) = \max_{\mu \in \mathcal{M}(G)} \left\{ \Theta^T \mu + H(\mu) \right\} \quad (2)$$

Where:

- $\Theta^T \mu$ is a vector notation that implies the multiplication of all marginal distributions ($\mu_i(X_i)$ and $\mu_\alpha(X_\alpha)$) with their corresponding parameters ($\theta_i(X_i)$ and $\theta_\alpha(X_\alpha)$) for all variables and factors.
- $\mathcal{M}(G)$ is the *marginal polytope* associated with a graph G (Wainwright and Jordan, 2008). A vector μ is in $\mathcal{M}(G)$ if it corresponds to the marginals of *some* distribution $p(\mathcal{X})$:

$$\mathcal{M}(G) = \left\{ \mu \mid \exists p(\mathcal{X}) \text{ s.t. } \begin{array}{l} \mu_i(X_i) = p(X_i) \\ \mu_\alpha(X_\alpha) = p(X_\alpha) \end{array} \right\}$$

- $H(\mu)$ is defined as the entropy of the unique exponential distribution p^* of the form in Eq. (1) consistent with the marginals μ :

$$H(\mu) = - \sum_{\mathcal{X}} p^*(\mathcal{X}) \log p^*(\mathcal{X}).$$

The objective in Eq. (2) is the negative of the *free energy functional*, denoted $F[\mu, \Theta]$. Elegantly, solving this optimization problem will result in a solution to the exponential summation needed in order to compute the partition function. Furthermore, the set μ that results in the optimum also provides the marginal probabilities for each variable and factor in the model.

In itself, this observation is not sufficient to provide an efficient algorithm, since the maximization in Eq. (2) is as hard as the original inference task. Specifically, $\mathcal{M}(G)$ is difficult to characterize and the computation of $H(\mu)$ is also intractable, so both need to be approximated. First, one can relax the optimization problem to be over an outer bound on the marginal polytope. In particular, it is

natural to require that the resulting *pseudo-marginals* obey some local normalization and marginalization constraints. These constraints define the *local polytope*:

$$L(G) = \left\{ \mu \geq 0 \mid \begin{array}{l} \sum_{x_i} \mu_i(x_i) = 1 \\ \sum_{x_\alpha \setminus x_i} \mu_\alpha(x_\alpha) = \mu_i(x_i) \end{array} \right\}.$$

Obviously, any set $\mu \in \mathcal{M}(G)$, that corresponds to some legal distribution, will obey these constraints. Moreover, one can define sets of pseudo-marginals that obey these local constraints but do not correspond to the marginal probabilities of any legal distribution. Thus, $L(G)$ defines an outer bound over $\mathcal{M}(G)$.

As for the entropy term, a family of entropy approximations with a long history in statistical physics is based on a weighted sum of local entropies:

$$H_{\mathbf{c}}(\mu) = \sum_r c_r H_r(\mu_r),$$

where r are subsets of variables (regions) and the coefficients c_r are called *counting numbers* (Yedidia et al., 2005). The approximate optimization problem then takes the form:

$$\log \tilde{Z}(\Theta) = \max_{\mu \in L(G)} \left\{ \Theta^T \mu + H_{\mathbf{c}}(\mu) \right\} \quad (3)$$

These insights led to an explosion of practical and theoretical interest in propagation based inference methods, and a range of improvements to the convergence behavior and approximation quality of the basic algorithms have been suggested (Wainwright et al., 2003; Wiergerinck and Heskes, 2003; Elidan et al., 06; Meshi et al., 2009).

1.2.3 Relational graphical models

As discussed in Section 1.1.3, in many domains, including protein-protein interaction networks, local patterns can recur many times in the network. Relational probabilistic models (Friedman et al., 1999; Getoor et al., 2001; Taskar et al., 2002) provide a language for constructing models from such reoccurring sub-

components. In these models, we distinguish the *template-level* model that describes the types of objects and components of the model and how they can be applied, from the *instantiation-level* that describes a particular model that is an instantiation of the template to a specific set of entities. Depending on the specific *instantiation*, these sub-components are duplicated to create the actual probabilistic model.

A set of template parameters $\Psi = \{\psi_1, \dots, \psi_T\}$ are used to parametrize all cliques using

$$P(\mathcal{X}; \Psi) = \frac{1}{Z(\Psi)} \exp \left(\sum_t \left(\sum_{i \in I(t)} \psi_t(X_i) + \sum_{\alpha \in I(t)} \psi_t(X_\alpha) \right) \right),$$

where $I(t)$ is the set of ground features that are mapped to the t 'th parameter. This template based representation allows the definition of large-scale models using a relatively small number of parameters.

This type of models has a couple of advantages. First, it enables instantiating models over very large data-sets while using a relatively small number of parameters. Second, it enables implementation that formalizes the relational nature of many real-life models. That is, it relates multiple observations of a local pattern to each other as manifestations of the same template rule in various instantiations. These two advantages are important in order to enable efficient and robust parameter estimation and structure learning.

1.3 Research Goals

The first paper in this thesis (Jaimovich et al., 2006) presents the foundations for formalizing probabilistic models over an interaction network. This model enables taking into account noisy observations from large-scale measurements while providing a simultaneous prediction over all interactions. I implement this model on a set of protein-protein interactions and show how such simultaneous prediction enables intricate information flow between the interactions, allowing for better

prediction of protein-protein interactions from noisy observations.

Application of our model to the entire protein-protein interaction network would require creation of a huge network over millions of interaction variables. In order to learn the properties of such models from data one should be able to answer queries regarding the marginal distributions of sets of variables. As exact inference is not feasible in such settings, the second paper in this thesis (Jaimovich et al., 2007) presents an algorithm for computing approximate answers to such queries with a cost that scales only with the size of the template model (and not with the size of the instantiation). We show that these approximations are equivalent to those that can be computed by standard belief propagation on the fully instantiated model.

The third paper in this thesis (Jaimovich et al., 2010b) presents a code framework, created in order to enable efficient implementation of various approximate inference algorithms to graphical models. Although this library was created in order to implement our ideas towards analysis of the protein-protein interaction network, a large effort was invested in implementation of such algorithms on general models. In addition, we made an effort to document the code infrastructure, and offer it as a free resource to the scientific community, hoping that it would serve other groups to use such models in their analysis.

The last paper in this thesis (Jaimovich et al., 2010a) presents analysis of genetic interaction networks. In this work we show how network analysis of such networks can provide an automatic division of the genes into functionally coherent modules. Furthermore, we show how using additional information on genetic screens performed under a range of chemical perturbations can shed light on the cellular function of specific modules. With the recent advances in technology, large-scale screens of genetic interactions are becoming a very popular assay. I believe that the methodology developed in this work will serve as a valuable tool for extracting biological insights from the results of these assays.

2 Paper chapter: Towards an integrated protein-protein interaction network: a relational Markov network approach

Ariel Jaimovich, Gal Elidan, Hanah Margalit and Nir Friedman.
Journal of Computational biology 13(2):145-65, 2006.

Towards an Integrated Protein–Protein Interaction Network: A Relational Markov Network Approach

ARIEL JAIMOVICH,^{1,2} GAL ELIDAN,³ HANAH MARGALIT,² and NIR FRIEDMAN¹

ABSTRACT

Protein–protein interactions play a major role in most cellular processes. Thus, the challenge of identifying the full repertoire of interacting proteins in the cell is of great importance and has been addressed both experimentally and computationally. Today, large scale experimental studies of protein interactions, while partial and noisy, allow us to characterize properties of interacting proteins and develop predictive algorithms. Most existing algorithms, however, ignore possible dependencies between interacting pairs and predict them independently of one another. In this study, we present a computational approach that overcomes this drawback by predicting protein–protein interactions simultaneously. In addition, our approach allows us to integrate various protein attributes and explicitly account for uncertainty of assay measurements. Using the language of *relational Markov networks*, we build a unified probabilistic model that includes all of these elements. We show how we can learn our model properties and then use it to predict all unobserved interactions simultaneously. Our results show that by modeling dependencies between interactions, as well as by taking into account protein attributes and measurement noise, we achieve a more accurate description of the protein interaction network. Furthermore, our approach allows us to gain new insights into the properties of interacting proteins.

Key words: Markov networks, probabilistic graphical models, protein–protein interaction networks.

1. INTRODUCTION

ONE OF THE MAIN GOALS OF MOLECULAR BIOLOGY is to reveal the cellular networks underlying the functioning of a living cell. Proteins play a central role in these networks, mostly by interacting with other proteins. Deciphering the protein–protein interaction network is a crucial step in understanding the structure, function, and dynamics of cellular networks. The challenge of charting these protein–protein interactions is complicated by several factors. Foremost is the sheer number of interactions that have to be considered. In the budding yeast, for example, there are approximately 18,000,000 potential interactions between the roughly 6,000 proteins encoded in its genome. Of these, only a relatively small fraction occur

¹School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel.

²Hadassah Medical School, The Hebrew University, Jerusalem, Israel.

³Computer Science Department, Stanford University, Stanford, CA.

in the cell (von Mering *et al.*, 2002; Sprinzak *et al.*, 2003). Another complication is due to the large variety of interaction types. These range from stable complexes that are present in most cellular states to transient interactions that occur only under specific conditions (e.g., phosphorylation in response to an external stimulus).

Many studies in recent years address the challenge of constructing protein–protein interaction networks. Several experimental assays, such as *yeast two-hybrid* (Uetz *et al.*, 2000; Ito *et al.*, 2001) and *tandem affinity purification* (Rigaut *et al.*, 1999) have facilitated high-throughput studies of protein–protein interactions on a genomic scale. Some computational approaches aim to detect functional relations between proteins, based on various data sources such as phylogenetic profiles (Pellegrini *et al.*, 1999) or mRNA expression (Eisen *et al.*, 1998). Other computational assays try to detect physical protein–protein interactions by, for example, evaluating different combinations of specific domains in the sequences of the interacting proteins (Sprinzak and Margalit, 2001).

The various experimental and computational screens described above have different sources of error and often identify markedly different subsets of the full interaction network. The small overlap between the interacting pairs identified by the different methods raises serious concerns about their robustness. Recently, in two separate works, von Mering *et al.* (2002) and Sprinzak *et al.* (2003) conducted a detailed analysis of the reliability of existing methods, only to discover that no single method provides a reasonable combination of sensitivity and recall. However, both studies suggest that interactions detected by two (or more) methods are much more reliable. This motivated later “meta” approaches that hypothesize about interactions by combining the predictions of computational methods, the observations of experimental assays, and other correlating information sources, such as that of localization assays. These approaches use a variety of machine learning methods to provide a combined prediction, including *support vector machines* (Bock and Gough, 2001), *naive Bayesian classifiers* (Jansen *et al.*, 2003), and *decision trees* (Zhang *et al.*, 2004).

While the above combined approaches lead to an improvement in prediction, they are still inherently limited by the treatment of each interaction independently of other interactions. In this paper, we argue that by explicitly modeling such dependencies, we can leverage observations from varied sources to produce better *joint* predictions of the protein interaction network as a whole. As a concrete example, consider the budding yeast proteins Pre7 and Pre9. These proteins were predicted to be interacting by a computational assay (Sprinzak and Margalit, 2001). However, according to a large-scale localization assay (Huh *et al.*, 2003), the two proteins are *not* co-localized; Pre9 is observed in the cytoplasm and in the nucleus (light gray) and Pre7 is not annotated to be in either one of those. This interaction was predicted by a computational assay (Sprinzak and Margalit, 2001) (dashed line). This evidence alone provides weak support for an interaction between the two proteins. (b) Two additional proteins Pre5 and Pup3. These were found to interact with Pre9 and Pre7 either by a computation assay (Sprinzak and Margalit, 2001) (dashed line) or experimental assays (Mewes *et al.*, 1998) (solid line). The combined evidence gives more support to the hypothesis that Pre7 and Pre9 interact.

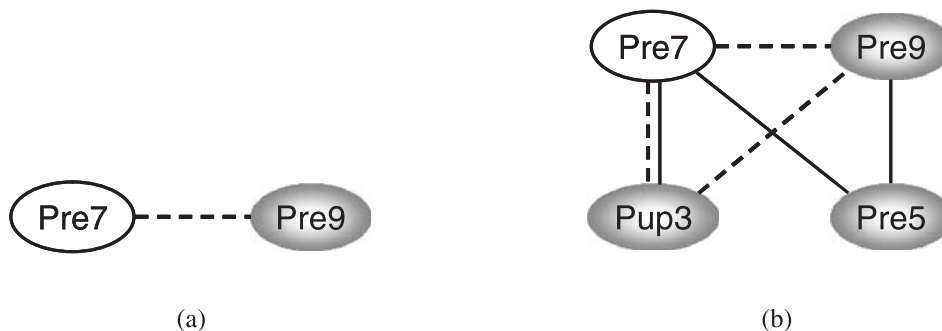


FIG. 1. Dependencies between interactions can be used to improve predictions. (a) A possible interaction of two proteins (Pre7 and Pre9). Pre9 is localized in the cytoplasm and in the nucleus (light gray) and Pre7 is not annotated to be in either one of those. This interaction was predicted by a computational assay (Sprinzak and Margalit, 2001) (dashed line). This evidence alone provides weak support for an interaction between the two proteins. (b) Two additional proteins Pre5 and Pup3. These were found to interact with Pre9 and Pre7 either by a computation assay (Sprinzak and Margalit, 2001) (dashed line) or experimental assays (Mewes *et al.*, 1998) (solid line). The combined evidence gives more support to the hypothesis that Pre7 and Pre9 interact.

see Fig. 1b. These observations suggest that these proteins might form a complex. Moreover, as both Pre5 and Pup3 were found to be localized both in the nucleus and in the cytoplasm, we may infer that Pre7 is also localized in these compartments. This in turn increases our belief that Pre7 and Pre9 interact. Indeed, this inference is confirmed by other interaction (Gavin *et al.*, 2002) and localization (Kumar, 2002) assays. This example illustrates two reasoning patterns that we would like to allow in our model. First, we would like to encode that certain patterns of interactions (e.g., within complexes) are more probable than others. Second, an observation relating to one interaction should be able to influence the attributes of a protein (e.g., localization), which in turn will influence the probability of other related interactions.

We present unified probabilistic models for encoding such reasoning and for learning an effective protein-protein interaction network. We build on the language of relational probabilistic models (Friedman *et al.*, 1999; Taskar *et al.*, 2002) to explicitly define probabilistic dependencies between related protein-protein interactions, protein attributes, and observations regarding these entities. The use of probabilistic models also allows us to explicitly account for measurement noise of different assays. Propagation of evidence in our model allows interactions to influence one another as well as related protein attributes in complex ways. This in turn leads to better and more confident overall predictions. Using various proteomic data sources for the yeast *Saccharomyces cerevisiae*, we show how our method can build on multiple weak observations to better predict the protein-protein interaction network.

2. A PROBABILISTIC PROTEIN-PROTEIN INTERACTION MODEL

Our goal is to build a unified probabilistic model that can capture the integrative properties of protein-protein interactions as exemplified in Fig. 1. We represent protein-protein interactions, interaction assays readout, and other protein attributes as random variables. We model the dependencies between these entities (e.g., the relation between an interaction and an assay result) by a joint distribution over these variables. Using such a joint distribution, we can answer queries such as What is the most likely interaction map given an experimental evidence? However, a naive representation of the joint distribution requires a huge number of parameters. To avoid this problem, we rely on the language of *relational Markov networks* to compactly represent the joint distribution. We now review relational Markov network models and the specific models we construct for modeling protein-protein interaction networks.

2.1. Markov networks for interaction models

Markov networks belong to the family of probabilistic graphical models. These models take advantage of conditional independence properties that are inherent in many real world situations to enable representation and investigation of complex stochastic systems. Formally, let $\mathcal{X} = \{X_1, \dots, X_N\}$ be a finite set of random variables. A *Markov network* over \mathcal{X} describes a joint distribution by a set of potentials Ψ . Each potential $\psi_c \in \Psi$ defines a measure over a set of variables $\mathbf{X}_c \subseteq \mathcal{X}$. We call \mathbf{X}_c the *scope* of ψ_c . The potential ψ_c quantifies local preferences about the joint behavior of the variables in \mathbf{X}_c by assigning a numerical value to each joint assignment of \mathbf{X}_c . Intuitively, the larger the value, the more likely the assignment. The joint distribution is defined by combining the preferences of all potentials

$$P(\mathcal{X} = \mathbf{x}) = \frac{1}{Z} \prod_c \psi_c(\mathbf{x}_c) \quad (1)$$

where \mathbf{x}_c refers to the projection of \mathbf{x} onto the subset \mathbf{X}_c and Z is a normalizing factor, often called the *partition function*, that ensures that P is a valid probability distribution.

The above product form facilitates compact representation of the joint distribution. Thus, we can represent complex distributions over many random variables using a relatively small number of potentials, each with limited scope. Moreover, in some cases the product form facilitates efficient probabilistic computations. Finally, from the above product form, we can read properties of (conditional) independencies between random variables. Namely, two random variables might depend on each other if they are in the scope of a single potential, or if one can link them through a series of intermediate variables that are in a scope of other potentials. We refer the reader to Pearl (1988) for a careful exposition of this subject. Thus, potentials confer dependencies among the variables in their scope, and unobserved random variables can

mediate such dependencies. As we shall see below, this criteria allows us to easily check for conditional independence properties in the models we construct.

Using this language to describe protein–protein interaction networks requires defining the relevant random variables and the potential describing their joint behavior. A distribution over protein–protein interaction networks can be viewed as the joint distribution over binary random variables that denote interactions. Given a set of proteins $\mathcal{P} = \{p_i, \dots, p_k\}$, an interaction network is described by interaction random variables I_{p_i, p_j} for each pair of proteins. The random variable I_{p_i, p_j} takes the value 1 if there is an interaction between the proteins p_i and p_j , and 0 otherwise. Since this relationship is symmetric, we view I_{p_j, p_i} and I_{p_i, p_j} as two ways of naming the same random variable. Clearly, a joint distribution over all these interaction variables is equivalent to a distribution over possible interaction networks.

The simplest Markov network model over the set of interaction variables has a univariate potential $\psi_{i,j}(I_{p_i, p_j})$ for each interaction variable. Each such potential captures the prior (unconditional) preference for an interaction versus a noninteraction by determining the ratio between $\psi_{i,j}(I_{p_i, p_j} = 1)$ and $\psi_{i,j}(I_{p_i, p_j} = 0)$. This model yields the next partition of the joint distribution function:

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{p_i, p_j \in \mathcal{P}} e^{\psi_{i,j}(I_{p_i, p_j})} \quad (2)$$

Figure 2a shows the graphic representation of such a model for three proteins. This model by itself is overly simplistic as it views interactions as independent from one another.

We can extend this oversimplistic model by incorporating protein attributes that influence the probability of interactions. Here we consider cellular localization as an example of such an attribute. The intuition is simple: if two proteins interact, they have to be physically co-localized. As a protein may be present in multiple localizations, we model cellular localization by several indicator variables, L_{l, p_i} , that denote whether the protein p_i is present in the cellular localization $l \in \mathcal{L}$. We can now relate the localization variables for a pair of proteins with the corresponding interaction variable between them by introducing a potential $\psi_{l,i,j}(L_{l, p_i}, L_{l, p_j}, I_{p_i, p_j})$. Such a potential can capture preference for interactions between co-localized proteins. Note that in this case the order of p_i and p_j is not important, and thus we require this potential to be symmetric around the role of p_i and p_j (we return to this issue in the context of learning). As with interaction variables, we might also have univariate potentials on each localization variable L_{l, p_j} that capture preferences over the localizations of specific proteins.

Assuming that \mathcal{X} contains variables $\{I_{p_i, p_j}\}$ and $\{L_{l, p_i}\}$, we now have a Markov network of the form

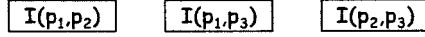
$$P(\mathcal{X}) = \frac{1}{Z} \prod_{p_i, p_j \in \mathcal{P}} e^{\psi_{i,j}(I_{p_i, p_j})} \prod_{l \in \mathcal{L}, p_i \in \mathcal{P}} e^{\psi_{l,i}(L_{l, p_i})} \prod_{l \in \mathcal{L}, p_i, p_j \in \mathcal{P}} e^{\psi_{l,i,j}(I_{p_i, p_j}, L_{l, p_i}, L_{l, p_j})} \quad (3)$$

The graph describing this model can be viewed in Fig. 2b. Here, representations of more complex distributions are possible, as interactions are no longer independent of each other. For example, I_{p_i, p_j} and L_{l, p_i} are co-dependent as they are in the scope of one potential. Similarly, I_{p_i, p_k} and L_{l, p_i} are in the scope of another potential. We conclude that the localization variable L_{l, p_i} mediates dependency between interactions of p_i with other proteins. Applying this argument recursively, we see that all interaction variables are co-dependent on each other. Intuitively, once we observe one interaction variable, we change our beliefs about the localization of the two proteins and in turn revise our belief about their interactions with other proteins.

However, if we observe all the localization variables, then the interaction variables are conditionally independent of each other. That is a result of the fact that if L_{l, p_i} is observed, it cannot function as a dependency mediator. Intuitively, once we observe the localization variables, observing one interaction cannot influence the probability of another interaction.

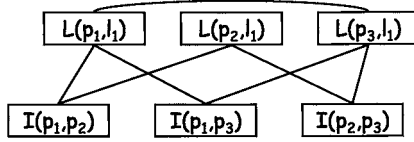
2.2. Noisy sensor models as directed potentials

The models we discussed so far make use of undirected potentials between variables. In many cases, however, a clear directional cause and effect relationship is known. In our domain, we do not observe protein interactions directly, but rather through experimental assays. We can explicitly represent the stochastic



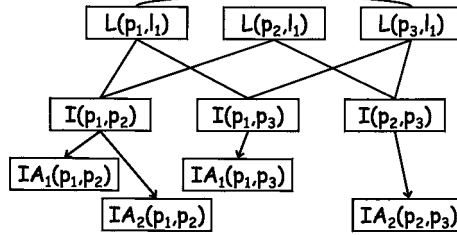
$$P(I_{p_1,p_2}, I_{p_2,p_3}, I_{p_1,p_3}) = \frac{1}{Z} e^{\psi_{1,2}(I_{p_1,p_2})} e^{\psi_{2,3}(I_{p_2,p_3})} e^{\psi_{1,3}(I_{p_1,p_3})}$$

(a)



$$P(I_{p_1,p_2}, I_{p_2,p_3}, I_{p_1,p_3}, L_{l,p_1}, L_{l,p_2}, L_{l,p_3}) = e^{\psi_{l,1}(L_{l,p_1})} e^{\psi_{l,2}(L_{l,p_2})} e^{\psi_{l,3}(L_{l,p_3})} e^{\psi_{l,1,2}(L_{l,p_1}, L_{l,p_2}, I_{p_1,p_2})} e^{\psi_{l,2,3}(L_{l,p_2}, L_{l,p_3}, I_{p_2,p_3})} e^{\psi_{l,1,3}(L_{l,p_1}, L_{l,p_3}, I_{p_1,p_3})}$$

(b)



$$P(I_{p_1,p_2}, I_{p_2,p_3}, I_{p_1,p_3}, L_{l,p_1}, L_{l,p_2}, L_{l,p_3}, IA_{p_1,p_2}^1, IA_{p_1,p_2}^2, IA_{p_2,p_3}^1, IA_{p_1,p_3}^2) = P(I_{p_1,p_2}, I_{p_2,p_3}, I_{p_1,p_3}, L_{l,p_1}, L_{l,p_2}, L_{l,p_3}) P(IA_{p_1,p_2}^1 | I_{p_1,p_2}) P(IA_{p_1,p_2}^2 | I_{p_1,p_2}) P(IA_{p_2,p_3}^1 | I_{p_2,p_3}) P(IA_{p_1,p_3}^2 | I_{p_1,p_3})$$

(c)

FIG. 2. Illustration of different models describing underlying different independence assumptions for a model over three proteins. An undirected arc between variables denotes that the variables coappear in the scope of some potential. A directed arc denotes that the target depends on the source in a conditional distribution. (a) Model shown in Equation (2) that assumes all interactions are independent of each other. (b) Model shown in Equation (3) that introduces dependencies between interactions using their connection with the localization of the proteins. (c) Model described in Equation (4) that adds noisy sensors to the interaction variables.

relation between an interaction and its assay readout within the model. For each interaction assay $a \in \mathcal{A}$ aimed toward evaluating the existence of an interaction between the proteins p_i and p_j , we define a binary random variable IA_{p_i,p_j}^a . Note that this random variable is not necessarily symmetric, since for some assays, such as yeast two hybrid, IA_{p_i,p_j}^a and IA_{p_j,p_i}^a represent the results of two different experiments.

It is natural to view the assay variable IA_{p_i,p_j}^a as a noisy sensor of the real interaction I_{p_i,p_j} . In this case, we can use a *conditional distribution* potential that captures the probability of the observation given

the underlying state of the system:

$$e^{\psi_{i,j}^a(IA_{p_i,p_j}^a, I_{p_i,p_j})} P(IA_{p_i,p_j}^a | I_{p_i,p_j}).$$

Conditional probabilities have several benefits. First, due to local normalization constraints, the number of free parameters of a conditional distribution is smaller (two instead of three in this example). Second, such potentials do not contribute to the global partition function Z , which is typically hard to compute. Finally, the specific use of directed models will allow us to prune unobserved assay variables. Namely, if we do not observe IA_{p_i,p_j}^a , we can remove it from the model without changing the probability over interactions.

Probabilistic graphical models that combine directed and undirected relations are called *chain graphs* (Buntine, 1995). Here we examine a simplified version of chain graphs where a dependent variable associated with a conditional distribution (i.e., IA_{p_i,p_j}^a) is not involved with other potentials or conditional distributions. If we let \mathcal{Y} denote the assay variables, then the joint distribution is factored as

$$P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{X})P(\mathcal{Y}|\mathcal{X}) = P(\mathcal{X}) \prod_{p_i,p_j} P(IA_{p_i,p_j}^a | I_{p_i,p_j}) \quad (4)$$

where $P(\mathcal{X})$ is the Markov network of Equation (3). The graph for this model is described in Fig. 2c.

2.3. Template Markov networks

Our aim is to construct a Markov network over a large-scale protein–protein interaction network. Using the model described above for this task is problematic in several respects. First, for the model with just univariate potentials over interaction variables, there is a unique parameter for each possible assignment of each possible interaction of protein pairs. The number of parameters is thus extremely large even for the simplest possible model (in the order of $\approx \frac{6000^2}{2}$ for the protein–protein interaction network of the budding yeast *S. cerevisiae*). Robustly estimating such a model from finite data is clearly impractical. Second, we want to generalize and learn “rules” (potentials) that are applicable throughout the interaction network, regardless of the specific subset of proteins we happen to concentrate on. For example, we want the probabilistic relation between interaction (I_{p_i,p_j}) and localization (L_{l,p_i}, L_{l,p_j}), to be the same for all values of i and j .

We address these problems by using *template models*. These models are related to relational probabilistic models (Friedman *et al.*, 1999; Taskar *et al.*, 2002) in that they specify a recipe with which a concrete Markov network can be constructed for a specific set of proteins and localizations. This recipe is specified via *template potentials* that supply the numerical values to be reused. For example, rather than using a different potential $\psi_{l,i,j}$ for each protein pair p_i and p_j , we use a single potential ψ_l . This potential is used to relate an interaction variable I_{p_i,p_j} with its corresponding localization variables L_{l,p_i} and L_{l,p_j} , regardless of the specific choice of i and j . Thus, by reusing parameters, a template model facilitates a compact representation and at the same time allows us to apply the same “rule” for similar relations between random variables.

The design of the template model defines the set of potentials that are shared. For example, when considering the univariate potential over interactions, we can have a single template potential for all interactions $\psi(I_{p_i,p_j})$. On the other hand, when looking at the relation between localization and interaction, we can decide that for each localization value l we have a different template potential for $\psi_l(L_{l,p_i})$. Thus, by choosing which templates to create, we encapsulate the complexity of the model.

For the model of Equation (3), we introduce one template potential $\psi(I_{p_i,p_j})$ and one template potential for each localization l that specifies the recipe for potentials of the form $\psi_l(I_{p_i,p_j}, L_{l,p_i}, L_{l,p_j})$. The first template potential has one free parameter, and each of the latter ones have five free parameters (due to symmetry). We see that the number of parameters is a small constant, instead of growing quadratically with the number of proteins.

2.4. Protein–protein interaction models

The discussion so far defined the basis for a simple template Markov network for the protein–protein interaction network. The form given in Equation (4) relates protein interactions with multiple interaction

assays (Fig. 3a) and protein localizations (Fig. 3b). In this model, the observed interaction assays are viewed as noisy sensors of the underlying interactions. Thus, we explicitly model experiment noise and allow the measurement to stochastically differ from the ground truth. For each type of assay, we have a different conditional probability that reflects the particular noise characteristics of that assay. In addition, the basic model contains a univariate template potential $\psi(I_{p_i,p_j})$ that is applied to each interaction variable. This potential captures the prior preferences for interaction (before we make any additional observations).

In this model, if we observe the localization variables, then, as discussed above, interaction variables are conditionally independent. This implies that if we observe both the localization variables and the interaction assay variables, the posterior over interactions can be reformulated as an independent product of terms, each one involving I_{p_i,p_j} , its related assays, and the localization of p_i and p_j . Thus, the joint model can be viewed as a collection of independent models for each interaction. Each of these models is equivalent to a naive Bayes model (see, e.g., Jansen *et al.* [2003]). We call this the *basic* model (see Fig. 3e).

We now consider two extensions to the basic model. The first extension relates to the localization random variables. Instead of using the experimental localization results to assign these variables, we can view these experimental results as noisy sensors of the true localization. To do so, we introduce localization assay random variables $LA_{i,p}$, which are observed, and relate each localization assay variable to its corresponding hidden ground truth variable using a conditional probability (Fig. 3c). The parameters of this conditional probability depend on the type of assay and the specific cellular localization. For example, some localizations, such as “bud,” are harder to detect as they represent a transient part of the cell cycle, while other localizations, such as “cytoplasm,” are easier to detect since they are present in all stages of the cell’s life and many proteins are permanently present in them. As we have seen above, allowing the model to infer the localization of a protein provides a way to create dependencies between interaction variables. For example, an observation of an interaction between p_i and p_j may change the

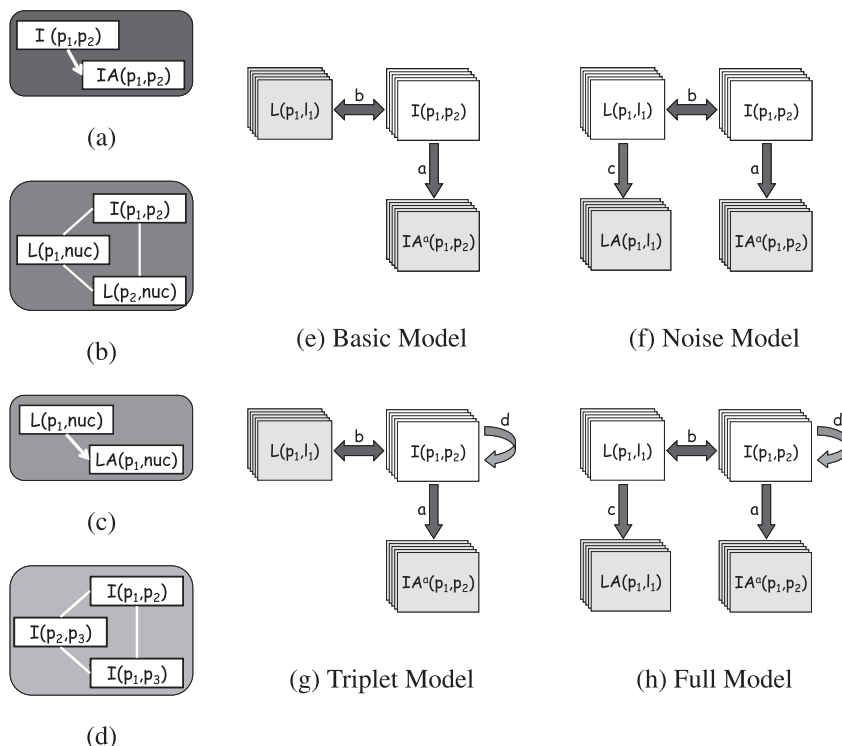


FIG. 3. Protein-protein interaction models. In all models, a plain box stands for a hidden variable, and a shadowed box represents an observed variable. The model consists of four classes of variables and four template potentials that relate them. (a) Conditional probability of an interaction assay given the corresponding interaction; (b) potential between an interaction and the localization of the two proteins; (c) conditional probability of a localization assay given a corresponding localization; (d) potential between three related interacting pairs; (e)–(h) The four models we build and how they hold the variable classes and global relations between them.

belief in the localization of p_i and thereby influence the belief about the interaction between p_i and another protein, p_k , as in the example of Fig. 1. We use the name *noise* model to refer to the basic model extended with localization assay variables (see Fig. 3f). This model allows, albeit indirectly, interactions to influence each other in complex ways via co-related localization variables.

In the second extension, we explicitly introduce direct dependencies between interaction variables by defining potentials over several interaction variables. The challenge is to design a potential that captures relevant dependencies in a concise manner. Here we consider dependencies between the three interactions among a triplet of proteins. More formally, we introduce a three variables potential $\psi_3(I_{p_i,p_j}, I_{p_i,p_k}, I_{p_j,p_k})$ (Fig. 3d). This model is known in the social network literature as the *triad model* (Frank and Strauss, 1986). Such a triplet potential can capture properties such as preferences for (or against) adjacent interactions, as well as transitive closure of adjacent edges. Given our set of proteins \mathcal{P} , the induced Markov network has $\binom{|\mathcal{P}|}{3}$ potentials, all of which replicate the same parameters of the template potential. Note that this requires the potential to be ignorant of the order of its arguments (as we can “present” each triplet of interactions in any order). Thus, the actual number of parameters for ψ_3 is four—one when all three interactions are present, another for the case when two are present, and so on. We use the name *triplet* model to refer to the basic model extended with these potentials (see Fig. 3g). Finally, we use the name *full* model to refer to the basic model with both the extensions of noise and triplet (see Fig. 3h).

3. LEARNING AND INFERENCE

In the previous section, we qualitatively described the design of our model and the role of the template potentials, given the interpretation we assign to the different variables. In this section, we address situations where this qualitative description of the model is given and we need to find an explicit quantification for these potentials. At first sight, it may appear as if we could manually decide, based on expert advice, on the values of this relatively small number of parameters. Such an approach is problematic in several respects. First, a seemingly small difference might have a significant effect on the predictions. This effect is amplified by the numerous times each potential is used within the model. We may not expect an expert to be able to precisely quantify the potentials. Second, even if each potential can be quantified reasonably on its own, our goal is to have the potentials work in concert. Ensuring this is nearly impossible using manual calibration.

To circumvent these problems, we adopt a data-driven approach for estimating the parameters of our model, using real-life evidence. That is, given a dataset \mathcal{D} of protein–protein interactions, as well as localization and interaction assays, we search for potentials that best “explain” the observations. To do so, we use the *maximum likelihood* approach where our goal is to find a parameterization Θ so that the log probability of the data, $\log P(\mathcal{D} | \Theta)$, is maximized. Note that obtaining such a database \mathcal{D} is not always an easy task. In our case, it means we have to find a reliable set of both interacting protein pairs and “noninteracting” protein pairs. Finding such a reliable database is not simple, since we have no evidence for such a “noninteraction.”

3.1. Complete data

We first describe the case where \mathcal{D} is complete, that is, every variable in the model is observed. Recall that our model has both undirected potentials and conditional probabilities. Estimating conditional probabilities from complete data is straightforward and amounts to gathering the relevant *sufficient statistics* counts. For example, for the template parameter corresponding to a positive interaction assay given that the interaction actually exists, we have

$$P(IA_{p_i,p_j}^a = 1 | I_{p_i,p_j} = 1) = \frac{N(IA_{p_i,p_j}^a = 1, I_{p_i,p_j} = 1)}{N(I_{p_i,p_j} = 1)} \quad (5)$$

where $N(IA_{p_i,p_j}^a = 1, I_{p_i,p_j} = 1)$ is the number of times both IA_{p_i,p_j}^a and I_{p_i,p_j} are equal to one in \mathcal{D} and similarly for $N(I_{p_i,p_j} = 1)$ (see, for example, Heckerman [1998]). Note that this simplicity of estimating

conditional probability is an important factor in preferring these to undirected potentials where it is natural to do so.

Finding the maximum likelihood parameters for undirected potentials is more involved. Although the likelihood function is concave, there is no closed-form formula that returns the optimal parameters. This is a direct consequence of the factorization of the joint distribution Equation (1). The different potentials are linked to each other via the partition function, and thus we cannot optimize each of them independently. A common heuristic is a gradient ascent search in the parameter space (e.g., Bishop [1995]). This requires that we repeatedly compute both the likelihood and its partial derivatives with respect to each parameter. It turns out that for a specific entry in a potential $\psi_c(\mathbf{x}_c)$, the gradient is

$$\frac{\partial \log P(\mathcal{D} \mid \Theta)}{\partial \psi_c(\mathbf{x}_c)} = \hat{P}(\mathbf{x}_c) - P(\mathbf{x}_c \mid \Theta) \tag{6}$$

where $\hat{P}(\mathbf{x}_c)$ is the empirical count of \mathbf{x}_c (Della Pietra *et al.*, 1997). Thus, the gradient equals to the difference between the empirical count of an event and the probability of that event $P(\mathbf{x}_c)$ as predicted by the model. This is in accordance with the intuition that at the maximum likelihood parameters, where the gradient is zero, the predictions of the model and the empirical evidence match. Note that this estimation may be significantly more time consuming than in the case of conditional probabilities, and that it is sensitive to the large dimension of the parameter space—the combined number of all values in all the potentials.

3.2. Parameter sharing

In our template model, we use many potentials which share the same parameters. In addition to the conceptual benefits of such a model (as described in Section 2), template potentials can also help us in parameter estimation. In particular, the large reduction of the size of the parameter space significantly speeds up and stabilizes the estimation of undirected potentials. Furthermore, many observations contribute to the estimation of each potential, leading to an estimation that is more robust.

In our specific template model, we also introduce constraints on the template potentials to ensure that the model captures the desired semantics (e.g., invariance to protein order). These constraints are encoded by parameter sharing and parameter fixing (e.g., if two proteins are not in a specific cellular location, the potential value should have no effect on the interaction of these two proteins). This further reduces the size of the parameter space in the model. See Fig. 4 for the design of our potentials.

Learning with shared parameters is essentially similar to simple parameter learning. Concretely, let a set of potentials \mathcal{C} share a common potential parameter θ so that for all $c \in \mathcal{C}$ we have $\psi_c(\mathbf{x}_c) = \theta$. Using the chain rule of partial derivatives, it can be shown that

$$\frac{\partial \log P(\mathbf{e})}{\partial \theta} = \sum_{c \in \mathcal{C}} \frac{\partial \log P(\mathbf{e})}{\partial \psi_c(\mathbf{x}_c)}$$

Thus, the derivatives with respect to the template parameters are aggregates of the derivatives of the corresponding entries in the potentials of the model. Similarly, estimating template parameters for conditional potentials amount to an aggregation of the relevant counts.

It is important to note that evaluating the gradients does not require access to the whole data. As the gradient depends only on the aggregate count associated with each parameter, we need to store only these sufficient statistics.

3.3. Incomplete data

In real life, the data is seldom complete, and some variables in the model are unobserved. In fact, some variables, such as the true location of a protein, are actually hidden variables that are never observed directly. To learn in such a scenario, we use the *expectation maximization* (EM) algorithm (Dempster *et al.*, 1977). The basic intuition is simple. We start with some initial guess for the model’s parameters. We then use the model and the current parameters to “complete” the missing values in \mathcal{D} (see Section 3.4 below).

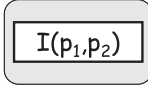
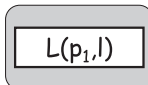
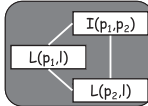
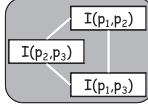
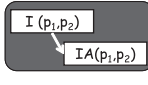
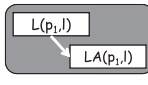
Potentials	Free parameters
Univariate interaction 	$\psi(I_{p_1, p_2} = 1)$
Univariate localization 	For each cellular compartment l : $\psi_l(L_{l, p_1} = 1)$
Colocalization 	For each cellular compartment l : $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 1, L_{l, p_2} = 1)$ $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 1, L_{l, p_2} = 0)$ $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 0, L_{l, p_2} = 0)$ * symmetric in p_1 and p_2
Interaction triplets 	$\psi_3(I_{p_1, p_2} = 1, I_{p_2, p_3} = 1, I_{p_1, p_3} = 1)$ $\psi_3(I_{p_1, p_2} = 1, I_{p_2, p_3} = 1, I_{p_1, p_3} = 0)$ * symmetric in p_1, p_2 and p_3
Conditional probabilities	Free parameters
Interaction assays 	For each interaction assay a : $P(IA_{p_1, p_2}^a = 1 I_{p_1, p_2} = 1)$ $P(IA_{p_1, p_2}^a = 1 I_{p_1, p_2} = 0)$
Localization assays 	For each localization assay a and cellular compartment l : $P(LA_{l, p_1} = 1 L_{l, p_1} = 1)$ $P(LA_{l, p_1} = 1 L_{l, p_1} = 0)$

FIG. 4. A summary of the free parameters that are learned in the model. For each potential/conditional distribution, we show the entries that need to be estimated. The remaining entries are set to 0 in potentials and to the complementary value in conditional distributions.

The parameters are then reestimated based on the “completed” data using the complete data procedure described above, and so on. Concretely, the algorithm has the following two steps:

- **E-step.** Given the observations \mathbf{e} , the model, and the current parameterization Θ , compute the *expected* sufficient statistics counts needed for estimation of the conditional probabilities and the posterior probabilities $P(\mathbf{x}_c | \mathbf{e}, \Theta)$ required for estimation of the undirected potentials.
- **M-step.** Maximize the parameters of the model using the computations of the *E*-step, as if these were computed from complete data.

Iterating these two steps is guaranteed to converge to a local maximum of the likelihood function.

3.4. Inference

The task of inference involves answering probabilistic queries given a model and its parameters. That is, given some evidence \mathbf{e} , we are interested in computing $P(\mathbf{x} \mid \mathbf{e}, \Theta)$ for some (possibly empty) set of variables \mathbf{e} as evidence. Inference is needed both when we want to make predictions about new unobserved entities and when we want to learn from unobserved data. Specifically, we are interested in computation of the likelihood $P(\mathcal{D} \mid \Theta)$ and the probability of the missing observations (true interactions and localization) given the observed assays.

In general, exact inference is computationally intensive (Cooper, 1990) except for a limited classes of structures (e.g., trees). Specifically, in our model that involves tens of thousands of potentials and many undirected cycles, exact inference is simply infeasible. Thus, we need to resort to an approximate method. Of the numerous approximate inference techniques developed in recent years, such as variational methods (e.g., Jordan *et al.* [1998]) and sampling-based methods (e.g., Neal [1993]), propagation based methods (e.g., Murphy and Weiss [1999]) have proved extremely successful and particularly efficient for large-scale models.

In this work, we use the *loopy belief propagation* algorithm (e.g., Pearl [1988]). The intuition behind the algorithm is straightforward. Let $b(\mathbf{x}_c)$ be the belief (current estimate of the marginal probability) of an inference algorithm about the assignment to some set of variables \mathbf{X}_c . When inference is exact $b(\mathbf{x}_c) = P(\mathbf{x}_c)$. Furthermore, beliefs over different subsets of variables are consistent in that they agree on the marginals of variables in their intersection. In belief propagation, we phrase inference as message passing between sets of variables, which are referred to as *cliques*. Each clique has its own potential that forms its initial belief. For example, these potentials can be defined using the same potentials as in the factorization of the joint distribution function in Equation (1). During belief propagation, each clique passes messages to cliques that share some of its variables, conveying its current belief over the variables in the intersection between the two cliques. Each message updates the beliefs of the receiving clique to calibrate the beliefs of the two cliques to be consistent with each other.

Concretely, a message from clique s to clique c that share some common variables is defined recursively as

$$m_{s \rightarrow c}(\mathbf{x}_{s \cap c}) = \prod_{t \in \mathcal{N}_s \setminus c} m_{t \rightarrow s}(\mathbf{x}_s) e^{\psi_s(\mathbf{x}_s)} \tag{7}$$

where $\psi_s(\mathbf{x}_s)$ is s 's potential, $s \cap c$ denotes the variables in the intersection of the two cliques, and \mathcal{N}_s is the set of neighbors (see below for description of the graph construction) of the clique s . The belief over a clique c is then defined as

$$b(\mathbf{x}_c) = e^{\psi_c(\mathbf{x}_c)} \prod_{s \in \mathcal{N}_c} m_{s \rightarrow c}(\mathbf{x}_c).$$

The result of these message propagations depends on the choice of cliques, their potentials, and the neighborhood structure between them. To perform inference in a model, we select cliques that are consistent with the model in the sense that each model potential (that is, every $\psi_c(\mathbf{x}_c)$ from Equation (1)) is absorbed in the potential of exactly one clique. This implies that the initial potentials of the cliques are exactly the potentials of the model. Moreover, we require that all the cliques that contain a particular variable X form one connected component. This implies that beliefs about X will be eventually shared by all cliques that contain it.

Pearl (1988) showed that if these conditions are met and the neighborhood structure is singly connected (that is, there is at most a single path between any two cliques), then this simple and intuitive algorithm is guaranteed to provide the exact marginals for each clique. In fact, using the correct ordering of messages, the algorithm converges to the true answer in just two passes along the tree.

The message defined in Equation (7) can be applied to an arbitrary clique neighborhood structure even if it contains loops. In this case, it is not even guaranteed that the final beliefs have a meaningful interpretation. In fact, in such a situation, the message passing is not guaranteed to converge. Somewhat surprisingly, applying belief propagation to graphs with loops produces good results even when the algorithm does not

converge and is arbitrarily stopped after some predefined time has elapsed (e.g., Murphy and Weiss [1999]). Indeed, the loopy belief propagation algorithm has been used successfully in numerous applications and fields (e.g., Freeman and Pasztor [2000] and McEliece *et al.* [1998]). The empirical success of the algorithm found theoretical basis with recent works and in particular with the work of Yedidia *et al.* (2002) that showed that even when the underlying graph is not a tree the fixed points of the algorithm correspond to local minima of the Bethe free energy.

Here we use the effective variant of loopy belief propagation which involves the construction of a *generalized cluster graph* over which the messages are propagated. The nodes in this graph are the cliques that are part of the model. An edge E_{sc} is created between any two cliques s and c that share common variables. The scope of an edge is the variables $X_{s \cap c}$ that are in the intersection of the scope of the two cliques. To ensure mathematical coherence, each variable X must satisfy the *running intersection property*: there must be one and only one path between any two cliques in which X appears. With the above construction, this amounts to requiring that X does not appear in a loop. We ensure this by constructing a spanning tree over the edges that have X in their scope and then remove it from the scope of all edges that are not part of that tree. We repeat this for all random variables in the graph. Messages are then propagated along the remaining edges and their scope. We note that our representation is only one out of several possible options. Each different representation might produce different propagation schemes and different resulting beliefs. We are guaranteed though that the insights of Yedidia *et al.* (2002) hold in all possible representations, as long as we satisfy the conditions above.

4. EXPERIMENTAL EVALUATION

In Section 2, we discussed a general framework for modeling protein–protein interactions and introduced four specific model variants that combine different aspects of the data. In this section, we evaluate the utility of these models in the context of the budding yeast *S. cerevisiae*. For this purpose, we choose to use four data sources, each with different characteristics. The first is a large-scale experimental assay for identifying interacting proteins by the yeast two hybrid method (Uetz *et al.*, 2000; Ito *et al.*, 2001). The second is a large-scale effort to curate experimental results from the literature about protein complexes (Mewes *et al.*, 1998). The third is a collection of computational predictions based on correlated domain signatures learned from experimentally determined interacting pairs (Sprinzak and Margalit, 2001). The fourth is a large scale experimental assay examining protein localization in the cell using GFP-tagged protein constructs (Huh *et al.*, 2003). Of the latter, we regarded four cellular localizations (nucleus, cytoplasm, mitochondria, and ER).

In our models, we have a random variable for each possible interaction and a random variable for each assay measuring such an interaction. In addition, we have a random variable for each of the four possible localizations of each protein and yet another variable corresponding to each localization assay. A model for all $\approx 6,000$ proteins in the budding yeast includes close to 20,000,000 random variables. Such a model is too large to cope with using our current methods. Thus, we limit ourselves to a subset of the protein pairs, retaining both positive and negative examples. We construct this subset from the study of von Mering *et al.* (2002) who ranked $\approx 80,000$ protein–protein interactions according to their reliability based on multiple sources of evidence (including some that we do not examine here). From this ranking, we consider the 2,000 highest-ranked protein pairs as “true” interactions. These 2,000 interactions involve 867 proteins. The selection of negative (noninteracting) pairs is more complex. There is no clear documentation of failure to find interactions, and so we consider pairs that do not appear in von Mering’s ranking as noninteracting. Since the number of such noninteracting protein pairs is very large, we randomly selected pairs from the 867 proteins and collected 2,000 pairs that do not appear in von Mering’s ranking as “true” noninteracting pairs. Thus, we have 4,000 interactions, of these, half interacting and half noninteracting. For these entities, the full model involves approximately 17,000 variables and 38,000 potentials that share 37 parameters.

The main task is to learn the parameters of the model using the methods described in Section 3. To get an unbiased estimate of the quality of the predictions with these parameters, we test our predictions on interactions that were not used for learning the model parameters. We use a standard four-fold cross validation technique, where in each iteration we learn the parameters using 1,500 positive and 1,500 negative interactions and then test on 500 unseen interactions of each type. Cross validation in the relational setting

is more subtle than learning with standard i.i.d. instances. In particular, when testing the predictions on the 1,000 unseen interactions, we use both the parameters we learned from the interactions in the training set and also the observations on these interactions. This simulates a real world scenario when we are given observations on some set of interactions, and are interested in predicting the remaining interactions, for which we have no direct observations.

To evaluate the performance of the different model elements, we compare the four models described in Section 2 (see Fig. 3). Figure 5 compares the test set performance of these four models. The advantage of using an integrative model that allows propagation of influence between interactions and protein attributes is clear, as all three variants improve significantly over the baseline model. Adding the dependency between different interactions leads to a greater improvement than allowing noise in the localization data. We hypothesize that this potential allows for complex propagation of beliefs beyond the local region of a single protein in the interaction network. When both elements are combined, the full model reaches quite impressive results: close to 85% true positive rate with just a 1% false positive rate. This is in contrast to the baseline model that achieves less than half of the above true-positive rate with the same amount of false positives.

A potential concern is that the parameters we learn are sensitive to the number of proteins and interactions we have. To further evaluate the robustness of the parameters in regard to these aspects, we applied the parameters learned using the 4,000 interactions described above in additional settings. Specifically, we increase the dataset of interaction by adding additional 2,000 positive examples (again from von Mering’s ranking) and 8,000 negative examples (random pairs that do not appear in von Mering’s ranking), resulting in a dataset of 14,000 interactions. We then performed four-fold cross-validation on this dataset, but used the parameters learned in the previous cross-validation trial rather than learning new parameters. The resulting ROC curve was quite similar to Fig. 5 (data not shown). This result indicates that at least in this range of numbers the learned parameters are not specific to a particular number of training interactions.

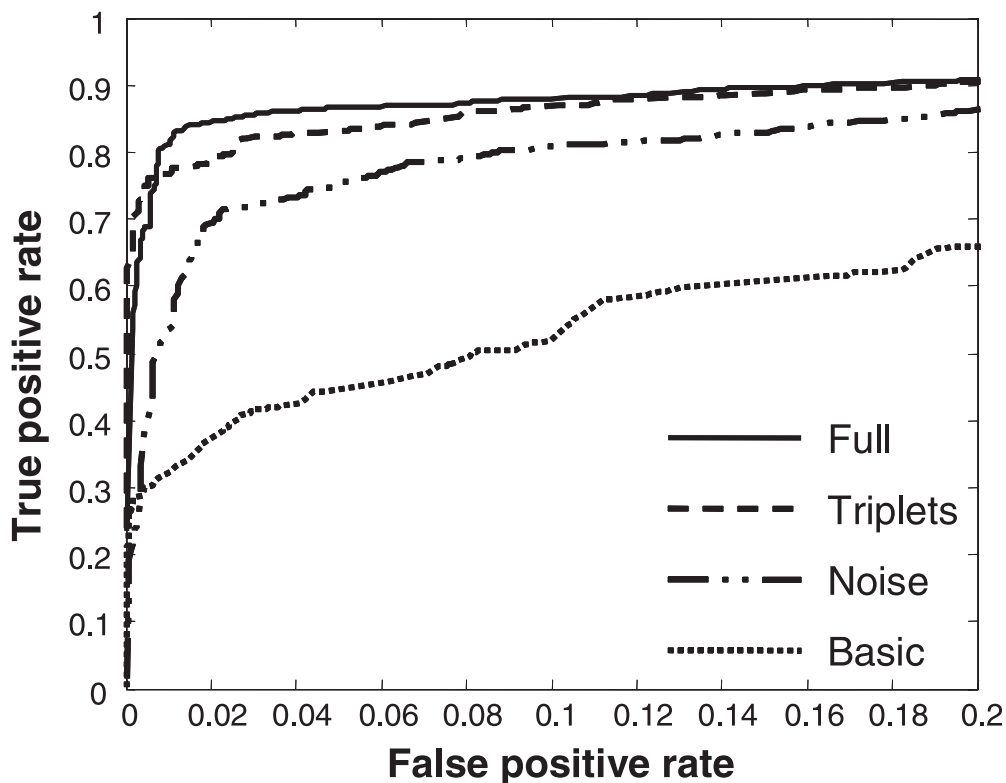


FIG. 5. Test performance (based on 4-fold cross validation) of the different models we evaluate. Shown is the true positive rate vs. the false positive rate for four models: **Basic** with just interaction, interaction assays, and localization variables; **Noise** that adds the localization assay variables; **Triplets** that adds a potential over three interactions; and **Full** that combines both extensions.

Another potential concern is that in real life we might have few observed interactions. In our cross-validation test, we have used the training interactions as observations when making our predictions about the test interactions. A harder task is to infer the test interactions without observing the training interactions. That is, we run prediction using only the observed experimental assays. We evaluated prediction accuracy as before using the same four-fold cross validation training but predicting test interactions without using the training set interactions as evidence. Somewhat surprisingly, the resulting ROC curves are quite similar to Fig. 5 with a slight decrease in sensitivity.

We can gain better insight into the effect of adding a noisy sensor model for localization by examining the estimated parameters (Fig. 6). As a concrete example, consider the potentials relating an interaction variable with the localization of the two relevant proteins in Fig. 6b. In both models, when only one of the proteins is localized in the compartment, noninteraction is preferred, and if both proteins are co-localized, interaction is preferred. We see that smaller compartments, such as the mitochondria, provide stronger support for interaction. Furthermore, we can see that our noise model allows us to be significantly more confident in the localization attributes in the nucleus and in the cytoplasm. This confidence might reveal, by using information from the learned interactions, the missing annotation of the interaction partners of these proteins.

Another way of examining the effect of the noisy sensor is to compare the localization predictions made by our model with the original experimental observations. For example, out of 867 proteins in our experiment, 398 proteins are observed as nuclear (Huh *et al.*, 2003). Our model predicts that 492 proteins are nuclear. Of these, 389 proteins were observed as nuclear, 36 are nuclear according to YPD (Costanzo *et al.*, 2001), 45 have other cellular localizations, and 22 have no known localization. We get similar results for other localizations. These numbers suggest that our model is able to correctly predict the localizations of many proteins, even when the experimental assay misses them.

As an additional test to evaluate the information provided by localization, we repeated the original cross-validation experiments with randomly reshuffled localization data. As expected, the performance of the basic model decreased dramatically. The performance of the full model, however, did not alter significantly. A possible explanation is that the training “adapted” the hidden localization variables to capture dependencies between interactions. Indeed, the learned conditional probabilities in the model capture a weak relationship between the localization variables and the shuffled localization assays. This experiment demonstrates the expressive power of the model in capturing dependencies and shows the ability of the model to use hidden protein attributes (the localization variables in this case) to capture dependencies

	Basic	Noise	Basic model		Noise model		
Interaction	0	-0.02	$L_{l,p_i} = 1$	$L_{l,p_i} = 1$	$L_{l,p_i} = 1$	$L_{l,p_i} = 1$	
Nucleus	-1.13	-0.91	$L_{l,p_j} = 0$	$L_{l,p_j} = 1$	$L_{l,p_j} = 0$	$L_{l,p_j} = 1$	
Cytoplasm	-1.34	-1.13	Nucleus	-0.47	0.66	-0.91	1.15
Mitochondria	-1.96	-2.04	Cytoplasm	-0.66	-0.02	-0.94	1.27
ER	-2.52	-2.52	Mitochondria	-0.71	1.26	-0.99	1.38
			ER	-0.82	1.18	-0.73	1.16

(a) Univariate potentials

(b) Localization to interaction

FIG. 6. Examples of potentials learned using the **Basic** and the **Noise** models. (a) Univariate potentials of interactions and the four localizations. The number shown is the difference between a positive and a negative value so that a larger negative number indicates preference for no interaction or against localization. (b) The four potentials between an interaction I_{p_i,p_j} and localizations of the proteins L_{l,p_i}, L_{l,p_j} for the four different localizations. For each model, the first column corresponds to the case where one protein is observed in the compartment while the other is not. The second column corresponds to the case where both proteins are observed in the compartment. The number shown is the difference between the potential value for interaction and the value for no interaction. As can be seen, co-localization typically increases the probability of interaction, while disagreement on localization reduces it. In the **Noise** model, co-localization provides more support for interaction, especially in the nucleus and cytoplasm.

between interaction variables. This experiment also reinforces the caution needed in interpreting what hidden variables represent. In our previous experiment, the localization assay was informative, and thus the hidden localization variables maintain the intended semantics. In the reshuffled experiment, the localization observations were uninformative, and the learned model in effect ignores them.

To get a better sense of the way in which our model improves predictions, we consider specific examples where the predictions of the full model differ from those of the basic model. Consider the unobserved interaction between the EBP2 and NUG1 proteins. These proteins are part of a large group of proteins involved in rRNA biogenesis and transport. Localization assays identify NUG1 in the nucleus, but do not report any localization for EBP2. The interaction between these two proteins was not observed in any of the three interaction assays included in our experiment and thus was considered unlikely by the basic model. In contrast, propagation of evidence in the full model effectively integrates information about interactions of both proteins with other rRNA processing proteins. We show a small fragment of this network in Fig. 7a. In this example, the model is able to make use of the fact that several nuclear proteins interact with *both* EBP2 and NUG1 and thus predicts that EBP2 is also nuclear and indeed interacts with NUG1. Importantly, these predictions are consistent with the cellular role of these proteins and are supported by independent experimental assays (Costanzo *et al.*, 2001; von Mering *et al.*, 2002).

Another, qualitatively different example involves the interactions between RSM25, MRPS9, and MRPS28. While there is no annotation of RSM25’s cellular role, the other two proteins are known to be components of the mitochondrial ribosomal complex. Localization assays identify RSM25 and MRPS28 in the mitochondria, but do not report any observations about MRPS9. As in the previous example, neither of these interactions was tested by the assays in our experiment. As expected, the baseline model predicts that both interactions do not occur with a high probability. In contrast, by utilizing a fragment of our network shown in Fig. 7b, our model predicts that MRPS9 is mitochondrial and that both interactions occur. Importantly, these predictions are supported by independent results (Costanzo *et al.*, 2001; von Mering *et al.*, 2002). These predictions suggest that RSM25 is related to the ribosomal machinery of the mitochondria. Such an important insight could not be gained without using an integrated model such as the one presented in this work.

Finally, we evaluate our model in a more complex setting. We consider the interactions of various proteins with the mediator complex. This complex has an important role in helping activator transcription factors to recruit the RNA polymerase II. We used the results of Gugliemi *et al.* (2004) as evidence for interactions with the mediator complex. We then applied the parameters previously learned to infer interactions of other proteins with the complex. Specifically, we found a set of 496 proteins that according to the ranking of von Mering *et al.* might be in interaction with proteins in the mediator complex. Among these proteins, there are 7,179 potential interactions according to that ranking. We then applied the inference procedure to the model involving these proteins and potential interactions, using the same assays as above and the same learned parameters, and taking the interactions within the mediator complex to be observed. The predicted



FIG. 7. Two examples demonstrating the difference between the predictions by our **Full** model and those of the **Basic** model. Solid lines denote observed interactions and a dashed line corresponds to an unknown one. Grey colored nodes represent proteins that are localized in the nucleus in Fig. (a) and in the mitochondria in Fig. (b). White colored nodes have no localization evidence. In (a), unlike the **Basic** model, our **Full** model correctly predicts that EBP2 is localized in the nucleus and that it interacts with NUG1. Similarly, in (b) we are able to correctly predict that MRPS9 is localized in the mitochondria and interacts with RSM25, which also interacts with MRPS28.

interaction network is shown in Fig. 8. Our model predicts that only a small set of the 496 proteins interact directly with the mediator complex. Two large complexes could be identified in the network: the proteasome complex and the TFIID complex. In the predicted network, these interact with the mediator complex via Tbf1 and Spt15, respectively, two known DNA binding proteins. Many other DNA binding proteins interact with the complex directly to recruit the RNA polymerase II.

5. DISCUSSION

In this paper we presented a general purpose framework for building integrative models of protein-protein interaction networks. Our main insight is that we should view this problem as a *relational learning problem*, where observations about different entities are not independent. We build on and extend tools from relational probabilistic models to combine multiple types of observations about protein attributes and protein-protein interactions in a unified model. We constructed a concrete model that takes into account interactions, interaction assays, localization of proteins in several compartments, and localization assays, as well as the relations between these entities. Our results demonstrate that modeling the dependencies between interactions leads to significantly better predictions. We have also shown that including observations of protein properties, namely, protein localization, and explicit modeling of noise in such observations, leads to further improvement. Finally, we have shown how evidence can propagate in the model in complex ways leading to novel hypotheses that can be easily interpreted.

Our approach builds on relational graphical models. These models exploit a template level description to induce a concrete model for a given set of entities and relations among these entities (Friedman *et al.*, 1999; Taskar *et al.*, 2002). In particular, our work is related to applications of these models to *link prediction* (Getoor *et al.*, 2001; Taskar *et al.*, 2004b). In contrast to these works, the large number of unobserved random variables in the training data poses significant challenges for the learning algorithm. Our probabilistic model over network topology is also related to models devised in the literature of *social networks* (Frank and Strauss, 1986). Recently, other studies tried to incorporate global views of the interaction network when predicting interactions. For example, Iossifov *et al.* (2004) proposed a method to describe properties of an interaction network topology when combining predictions from literature search and yeast two-hybrid data for a dataset of 83 proteins. Their model is similar to our triplet model in that it combines a model of dependencies between interactions with the likelihood of independent observations about interactions. Their model of dependencies, however, focuses on the global distribution of node degrees in the network, rather than on local patterns of interactions. Similarly, Morris *et al.* (2004) use degree distributions to impose priors on interaction graphs. They decompose the interactions observed by yeast two-hybrid data as a superimposition of several graphs, one representing the true underlying interactions, and another the systematic bias of the measurement technology. Other recent studies employ variants of Markov networks to analyze protein interaction data. In these studies, however, the authors assumed that the interaction network is given and use it for other tasks, e.g., predicting protein function (Deng *et al.*, 2004; Leone and Pagnani, 2005; Letovsky and Kasif, 2003) and clustering interacting co-expressed proteins (Segal *et al.*, 2003). In contrast to our model, these works can exploit the relative sparseness of the given interaction network to perform fast approximate inference.

Our emphasis here was on presenting the methodology and evaluating the utility of integrative models. These models can facilitate incorporation of additional data sources, potentially leading to improved predictions. The modeling framework allows us to easily extend the models to include other properties of both the interactions and the proteins, such as cellular processes or expression profiles, as well as different interaction assays. Moreover, we can consider additional dependencies that impact the global protein-protein interaction network. For example, a yeast two-hybrid experiment might be more successful for nuclear proteins and less successful for mitochondrial proteins. Thus, we would like to relate the cellular localization of a protein and the corresponding observation of a specific type of interaction assay. This can be easily achieved by incorporating a suitable template potential in the model. An exciting challenge is to learn which dependencies actually improve predictions. This can be done by methods of *feature induction* (Della Pietra *et al.*, 1997). Such methods can also allow us to discover high-order dependencies between interactions and protein properties.

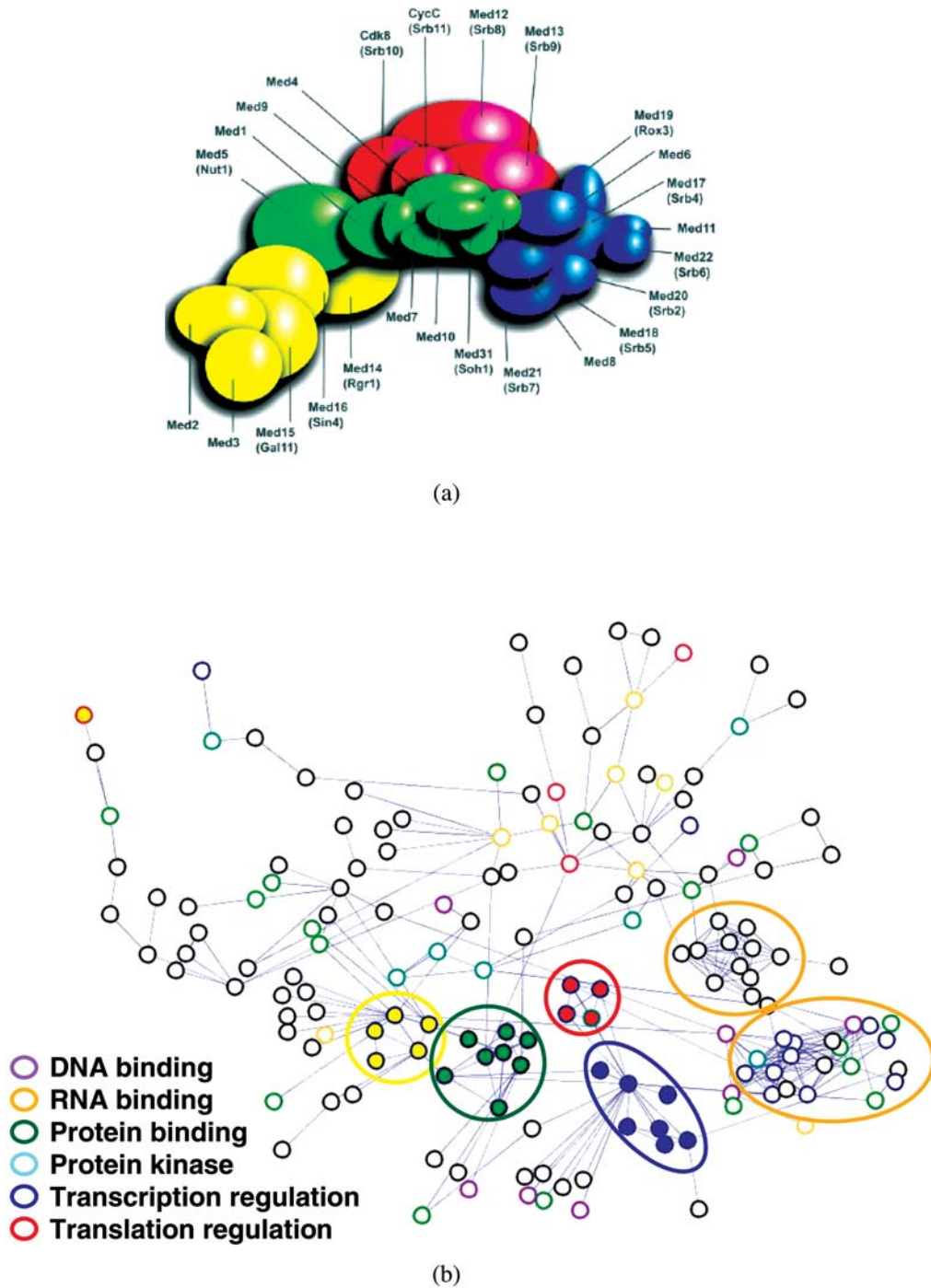


FIG. 8. (a) The mediator complex (taken from Figure 8b of Gugliemi *et al.* (2004) with permission). (b) Part of the interaction network predicted by our method (shown are interactions predicted with probability ≥ 0.5). Nodes are colored according to their GO annotation, and mediator complex subunits are painted as in (a). The lower orange circle marks the TFIID complex and the upper circle marks the proteasome complex.

Extending our framework to more elaborate models and networks that consider a larger number of proteins poses several technical challenges. Approximate inference in larger networks is both computationally demanding and less accurate. Generalizations of the basic loopy belief propagation method (e.g., Yedidia *et al.* [2002]) as well as other related alternatives (Jordan *et al.*, 1998; Wainwright *et al.*, 2002), may improve both the accuracy and the convergence of the inference algorithm. Learning presents additional computational and statistical challenges. In terms of computation, the main bottleneck lies in multiple invocations of the inference procedure. One alternative is to utilize information learned efficiently from few samples to prune the search space when learning larger models. Recent results suggest that large margin discriminative training of Markov networks can lead to a significant boost in prediction accuracy (Taskar *et al.*, 2004a). These methods, however, apply exclusively to fully observed training data. Extending these methods to handle partially observable data needed for constructing protein–protein interaction networks is an important challenge.

Finding computational solutions to the problems discussed above is a crucial step on the way to a global and accurate protein–protein interaction model. Our ultimate goal is to be able to capture the essential dependencies between interactions, interaction attributes, and protein attributes, and at the same time to be able to infer hidden entities. Such a probabilistic integrative model can elucidate the intricate details and general principles of protein–protein interaction networks.

ACKNOWLEDGMENTS

We thank Aviv Regev, Daphne Koller, Noa Shefi, Einat Sprinzak, Ilan Wapinski, Tommy Kaplan, Moran Yassour, and the anonymous reviewers for useful comments on previous drafts of this paper. Part of this research was supported by grants from the Israeli Ministry of Science, the United States–Israel Binational Science Foundation (BSF), the Isreal Science Foundation (ISF), European Union Grant QLRT-CT-2001-00015, and the National Institute of General Medical Sciences (NIGMS).

REFERENCES

- Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, United Kingdom.
- Bock, J.R., and Gough, D.A. 2001. Predicting protein–protein interactions from primary structure. *Bioinformatics* 17(5), 455–460.
- Buntine, W. 1995. Chain graphs for learning. *Proc. 11th Conf. on Uncertainty in Artificial Intelligence (UAI '95)*, 46–54.
- Cooper, G.F. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intell.* 42, 393–405.
- Costanzo, M.C., Crawford, M.E., Hirschman, J.E., Kranz, J.E., Olsen, P., Robertson, L.S., Skrzypek, M.S., Braun, B.R., Hopkins, K.L., Kondu, P., Lengieza, C., Lew-Smith, J.E., Tillberg, M., and Garrels, J.I. 2001. Ypd, pombe, and worm: Model organism volumes of the bioknowledge library, an integrated resource for protein information. *Nucl. Acids Res.* 29, 75–79.
- Della Pietra, S., Della Pietra, V., and Lafferty, J. 1997. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(4), 380–393.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B* 39, 1–39.
- Deng, M., Chen, T., and Sun, F. 2004. An integrated probabilistic model for functional prediction of proteins. *J. Comp. Biol.* 11, 463–475.
- Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95(25), 14863–14868.
- Frank, O., and Strauss, D. 1986. Markov graphs. *J. Am. Statist. Assoc.* 81.
- Freeman, W., and Pasztor, E. 2000. Learning low-level vision. *Int. J. Computer Vision* 40(1), 25–47.
- Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. 1999. Learning probabilistic relational models. *Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI '99)*, 1300–1309.
- Gavin, A.C., Bosche, M., Krause, R., *et al.* 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415(6868), 141–147.

- Getoor, L., Friedman, N., Koller, D., and Taskar, B. 2001. Learning probabilistic models of relational structure. *18th Int. Conf. on Machine Learning (ICML)*.
- Gugliemi, B., van Berkum, N.L., Klapholz, B., Bijma, T., Boube, M., Boschiero, C., Bourbon, H.M., Holstege, F.C., and Werner, M. 2004. A high resolution protein interaction map of the yeast mediator complex. *Nucl. Acid. Res.* 32, 5379–5391.
- Heckerman, D. 1998. A tutorial on learning Bayesian networks, in Jordan, M.I., ed., *Learning in Graphical Models*, Kluwer, Dordrecht, Netherlands.
- Huh, W.K., Falvo, J.V., Gerke, L.C., Carroll, A.S., Howson, R.W., Weissman, J.S., and O’Shea, E.K. 2003. Global analysis of protein localization in budding yeast. *Nature* 425, 686–691.
- Iossifov, I., Krauthammer, M., Friedman, C., Hatzivassiloglou, V., Bader, J.S., White, K.P., and Rzhetsky, A. 2004. Probabilistic inference of molecular networks from noisy data sources. *Bioinformatics* 20, 1205–1213.
- Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* 98(8), 4569–4574.
- Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., and Gerstein, M. 2003. A Bayesian networks approach for predicting protein–protein interactions from genomic data. *Science* 302(5644), 449–453.
- Jordan, M.I., Ghahramani, Z., Jaakkola, T., and Saul, L.K. 1998. An introduction to variational approximations methods for graphical models, in Jordan, M.I., ed., *Learning in Graphical Models*, Kluwer, Dordrecht, Netherlands.
- Kumar, A. 2002. Subcellular localization of the yeast proteome. *Genes Dev.* 16, 707–719.
- Leone, M., and Pagnani, A. 2005. Predicting protein functions with message passing algorithms. *Bioinformatics* 21, 239–247.
- Letovsky, S., and Kasif, S. 2003. Predicting protein function from protein protein interaction data: A probabilistic approach. *Bioinformatics* 19(Suppl. 1), i97–204.
- McEliece, R., McKay, D., and Cheng, J. 1998. Turbo decoding as an instance of pearl’s belief propagation algorithm. *IEEE J. on Selected Areas in Communication* 16, 140–152.
- Mewes, H.W., Hani, J., Pfeiffer, F., and Frishman, D. 1998. MIPS: A database for genomes and protein sequences. *Nucl. Acids Res.* 26, 33–37.
- Morris, Q.D., Frey, B.J., and Paige, C.J. 2004. Denoising and untangling graphs using degree priors. *Advances in Neural Information Processing Systems* 16.
- Murphy, K., and Weiss, Y. 1999. Loopy belief propagation for approximate inference: An empirical study. *Proc. 15th Conf. on Uncertainty in Artificial Intelligence (UAI ’99)*, 467–475.
- Neal, R.M. 1993. Probabilistic inference using Markov chain Monte Carlo methods. Technical report CRG-TR-93-1, Department of Computer Science, University of Toronto.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, New York.
- Pellegrini, M., Marcotte, E., Thompson, M., Eisenberg, D., and Yeates, T. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA* 96(8), 4285–4288.
- Rigaut, G., Shevchenko, A., Rutz, B., Wilm, M., Mann, M., and Seraphin, B. 1999. A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnol.* 17(10), 1030–1032.
- Segal, E., Wang, H., and Koller, D. 2003. Discovering molecular pathways from protein interaction and gene expression data. *Proc. 11th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*.
- Sprinzak, E., and Margalit, H. 2001. Correlated sequence-signatures as markers of protein–protein interaction. *J. Mol. Biol.* 311(4), 681–692.
- Sprinzak, E., Sattath, S., and Margalit, H. 2003. How reliable are experimental protein–protein interaction data? *J. Mol. Biol.* 327(5), 919–923.
- Taskar, B., Pieter Abbeel, A.P., and Koller, D. 2002. Discriminative probabilistic models for relational data. *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI ’02)*, 485–492.
- Taskar, B., Guestrin, C., Abbeel, P., and Koller, D. 2004a. Max-margin Markov networks. *Advances in Neural Information Processing Systems* 16.
- Taskar, B., Wong, M.F., Abbeel, P., and Koller, D. 2004b. Link prediction in relational data. *Advances in Neural Information Processing Systems* 16.
- Uetz, P., Giot, L., Cagney, G., et al. 2000. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature* 403(6770), 623–627.
- von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., and Bork, P. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417(6887), 399–403.
- Wainwright, M.J., Jaakkola, T., and Willsky, A.S. 2002. A new class of upper bounds on the log partition function. *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI ’02)*.
- Yedidia, J., Freeman, W., and Weiss, Y. 2002. Constructing free energy approximations and generalized belief propagation algorithms. Technical report TR-2002-35, Mitsubishi Electric Research Laboratories.

Zhang, L.V., Wong, S.L., King, O.D., and Roth, F.P. 2004. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics* 5(1), 38.

Address correspondence to:

Nir Friedman
Hebrew University
Jerusalem 91904, Israel

E-mail: nir@cs.huji.ac.il

3 Paper chapter: Template based inference in symmetric relational Markov random fields

Ariel Jaimovich, Ofer Meshi and Nir Friedman.

In the 23rd Conference on Uncertainty in Artificial Intelligence (*UAI 2007*).

Template Based Inference in Symmetric Relational Markov Random Fields

Ariel Jaimovich Ofer Meshi Nir Friedman

School of Computer Science and Engineering

Hebrew University of Jerusalem

Jerusalem, Israel 91904

{arielj,meshi,nir}@cs.huji.ac.il

Abstract

Relational Markov Random Fields are a general and flexible framework for reasoning about the joint distribution over attributes of a large number of interacting entities. The main computational difficulty in learning such models is inference. Even when dealing with complete data, where one can summarize a large domain by sufficient statistics, learning requires one to compute the expectation of the sufficient statistics given different parameter choices. The typical solution to this problem is to resort to approximate inference procedures, such as loopy belief propagation. Although these procedures are quite efficient, they still require computation that is on the order of the number of interactions (or features) in the model. When learning a large relational model over a complex domain, even such approximations require unrealistic running time.

In this paper we show that for a particular class of relational MRFs, which have inherent symmetry, we can perform the inference needed for learning procedures using a *template-level* belief propagation. This procedure's running time is proportional to the size of the relational model rather than the size of the domain. Moreover, we show that this computational procedure is equivalent to synchronous loopy belief propagation. This enables a dramatic speedup in inference and learning time. We use this procedure to learn relational MRFs for capturing the joint distribution of large protein-protein interaction networks.

1 Introduction

Relational probabilistic models are a rich framework for reasoning about structured joint distributions [6, 9]. Such models are used to model many types of domains like the web [22], gene expression measurements [20] and protein-protein interaction networks [11]. In these domains, they can be used for diverse tasks, such as prediction of missing

values given some observations [11], classification [22], and model selection [20]. All of these tasks require the ability to perform inference in these models. Since in many models exact inference is infeasible, most studies resort to approximate inference such as variational approximations [12] and sampling [8]. Unfortunately in many cases even these approximations are computationally expensive. This is especially problematic in settings where inference is performed many times, such as parameter estimation.

In this paper we show that we can exploit symmetry properties of relational models to perform efficient approximate inference. Our basic observation is that symmetry in the relational model implies that many of the intermediate results of approximate inference procedures, such as loopy belief propagation, are identical. Thus, instead of recalculating the same terms over and over, we can perform inference at the template level. We define formally a large class of relational models that have these symmetry properties, show how we can use them to perform efficient approximate inference and compare our results with other methods. This is, to the best of our knowledge, the first approximate inference algorithm that works on the template level of the model. However, this efficient inference procedure is limited to cases where we have no evidence on the model, since such evidence can break the symmetry properties. Nevertheless, we show that in many cases, inference with no evidence is useful, especially in learning tasks. Finally, we show a real life application by learning the properties of a model for protein-protein interactions.

2 Symmetric relational models

Relational probabilistic models [6, 9, 18, 21] provide a language for defining how to construct models from reoccurring sub-components. Depending on the specific *instantiation*, these sub-components are duplicated to create the actual probabilistic model. We are interested in models that can be applied for reasoning about the relations between entities. Our motivating example will be reasoning about the structure of interaction networks (*e.g.*, social interaction networks or protein-protein interaction networks). We now define a class of relational models that will be convenient for reasoning about these domains. We define a language

that is similar to ones previously defined [19], but also a bit different, to make our claims in the following section more clear.

As with most relational models in the literature we distinguish the *template-level* model that describe the types of objects and components of the model and how they can be applied, from the *instantiation-level* that describes a particular model which is an instantiation of the template to a specific set of entities.

To define a template-level model we first set up the different types of entities we reason about in the model. We distinguish *basic entity types* that describe atomic entities from *complex types* that describe composite entities.

Definition 2.1: Given a set $\mathcal{T}_{\text{basic}} = (T_1, \dots, T_n)$ of *basic entity types* we define two kinds of **complex types**:

- If T_1, \dots, T_k are basic types, then $T_1 \times \dots \times T_k$ denotes the type of *ordered tuples* of entities of these types. If e_1, \dots, e_k are entities of types T_1, \dots, T_k , respectively, then $\langle e_1, \dots, e_k \rangle$ is of type $T_1 \times \dots \times T_k$.
- If T is a basic type, then T^k denotes the type of *unordered tuples* of entities of type T . If e_1, \dots, e_k are entities of type T , then $[e_1, \dots, e_k]$ is of type T^k . When considering ordered tuples, permutations of the basic elements still refer to the same complex entity. Thus, if e_1, e_2 are of type T , then both $[e_1, e_2]$ and $[e_2, e_1]$ refer to the same complex entity of type T^2 .

For example, suppose we want to reason about undirected graphs. If we define a type T_v for vertices then an undirected edge is of type $T_e \equiv T_v^2$ since an edge is a composite object that consists of two vertices. Note that we use unordered tuples since the edge does not have a direction. That is, both $[v_1, v_2]$ and $[v_2, v_1]$ refer to the same relationship between the two vertices. If we want to model directed edges, we need to reason about ordered tuples $T_e \equiv T_v \times T_v$. Now $\langle v_1, v_2 \rangle$ and $\langle v_2, v_1 \rangle$ refer to two distinct edges. We can also consider social networks, where vertices correspond to people. Now we might also add a type T_l of physical locations. In order to reason about relationships between vertices (people) and locations we need to define pairs of type $T_p \equiv T_v \times T_l$. Note that tuples that relate between different types are by definition ordered.

Once we define the template-level set of types \mathcal{T} over some set of basic types $\mathcal{T}_{\text{basic}}$, we can consider particular instantiations in terms of entities.

Definition 2.2: An *entity instantiation* \mathcal{I} for $(\mathcal{T}_{\text{basic}}, \mathcal{T})$ consists of a set of *basic entities* \mathcal{E} and a mapping $\sigma : \mathcal{E} \mapsto \mathcal{T}_{\text{basic}}$ that assigns a basic type to each basic entity. ■

Based on an instantiation, we create all possible instantiations of each type in \mathcal{T} :

- if $T \in \mathcal{T}_{\text{basic}}$ then $\mathcal{I}(T) = \{e \in \mathcal{E} : \sigma(e) = T\}$

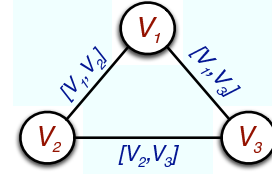


Figure 1: An instantiation of the graph scheme over a domain of three vertices.

- If $T = T_1 \times \dots \times T_k$ then $\mathcal{I}(T) = \mathcal{I}(T_1) \times \dots \times \mathcal{I}(T_k)$.
- If $T = T_1^k$ then $\mathcal{I}(T) = \{[e_1, \dots, e_k] : e_1, \dots, e_k \in \mathcal{I}(T_1), e_1 \leq \dots \leq e_k\}$ where \leq is some (arbitrary) order over $\mathcal{I}(T)$ ¹.

Once we define a set of basic entities, we assume that all possible complex entities of the given type are defined (see Figure 1 for an instantiation of the graph example).

The basic and complex entities define the structure of our domain of interest. Our goal, however, is to reason about the properties of these entities. We refer to these properties as *attributes*. Again, we start by the definition at the template level, and proceed to examine their application to a specific instantiation:

Definition 2.3: A **template attribute** $A(T)$ defines a property of entities of type T . The set of values the attribute can take is denoted $Val(A(T))$. ■

A template attribute denotes a specific property we expect each object of the given type to have. In general, we can consider attributes of basic objects or attributes of complex objects. In our example, we can reason about the color of a vertex, by having an attribute $Color(T_v)$. We can also create an attribute $Exist(T_e)$ that denotes whether the edge between two vertices exists. We can consider other attributes such as the weight of an edge and so on. All these template attribute are defined at the level of the scheme and we will denote by \mathcal{A} the set of template attributes in our model.

Given a concrete entity instance \mathcal{I} we consider all the attributes of each instantiated type. We view the attributes of objects as random variables. Thus, each template attribute in \mathcal{A} defines a set of random variables:

$$\mathcal{X}_{\mathcal{I}}(A(T)) = \{X_A(e) : e \in \mathcal{I}(T)\}$$

We define $\mathcal{X}_{\mathcal{I}} = \cup_{A(T) \in \mathcal{A}} \mathcal{X}_{\mathcal{I}}(A(T))$ to be the set of all random variables that are defined over the instantiation \mathcal{I} . For example, if we consider the attributes $Color$ over vertices and $Exist$ over unordered pairs of vertices,

¹For example, considering undirected edges again, we think of $[v_1, v_2]$ and $[v_2, v_1]$ as two different names of the same entity. Our definition ensures that only one of these two objects is in the set of entities and we view the other as an alternative reference to the same entity.

and suppose that $\mathcal{E} = \{v_1, v_2, v_3\}$ are all of type T_v , then we have three random variables in $\mathcal{X}(\text{Color}(T_v))$ which are $X_{\text{Color}}(v_1), X_{\text{Color}}(v_2), X_{\text{Color}}(v_3)$, and four random variables in $\mathcal{X}(\text{Exist}(T_e))$ which are $X_{\text{Exist}}([v_1, v_2]), X_{\text{Exist}}([v_1, v_3])$, and so on.

Given a set of types, their attributes and an instantiation, we defined a universe of discourse, which is the set $\mathcal{X}_{\mathcal{I}}$ of random variables. An *attribute instantiation* ω (or just instantiation) is an assignment of values to all random variables in $\mathcal{X}_{\mathcal{I}}$. We use both $\omega(X_A(e))$ and $x_A(e)$ to refer to the assigned value to the attribute A of the entity e .

We now turn to the final component of our relational model. To define a log-linear model over the random variables $\mathcal{X}_{\mathcal{I}}$, we need to introduce *features* that capture preferences for specific combinations of values to small groups of related random variables. In our graph example, we can introduce a univariate feature for edges that describes the prior potential for the existence of an edge in the graph. A more complex feature can describe preferences over triplets of interactions (e.g., prefer triangles over open chains).

We start by defining template level features as a recipe that will be assigned to a large number of specific sets of random variables in the instantiated model. Intuitively, a template feature defines a function that can be applied to a set of attributes of related entities. To do so, we need to provide a mechanism to capture sets of entity attributes with particular relationships. For example, to put a feature over triangle-like edges, we want a feature over the variables $X_{\text{Exist}}([v_1, v_2]), X_{\text{Exist}}([v_1, v_3])$, and $X_{\text{Exist}}([v_2, v_3])$ for every choice of three vertices v_1, v_2 , and v_3 . The actual definition, thus involves entities that we quantify over (e.g., v_1, v_2 , and v_3), the complex entities over these arguments we examine (e.g., $[v_1, v_2], [v_1, v_3]$, and $[v_2, v_3]$), the attributes of these entities, and the actual feature.

Definition 2.4: Template Feature A *template feature* \mathcal{F} is defined by four components:

- A tuple of *arguments* $\langle \xi_1, \dots, \xi_k \rangle$ with a corresponding list of *type signature* $\langle T_1^q, \dots, T_k^q \rangle$, such that ξ_i denotes an entity of basic type T_i^q .
- A list of formal entities $\varepsilon_1, \dots, \varepsilon_j$, with corresponding types T_1^f, \dots, T_j^f such that each formal entity ε is either one of the arguments, or a complex entity constructed from the arguments. (For technical reasons, we require that formal entities refer to each argument at most once.)
- A list of attributes $A_1(T_1^f), \dots, A_j(T_j^f)$.
- A function $f : \text{Val}(A_1(T_1^f)) \times \dots \times \text{Val}(A_j(T_j^f)) \mapsto \mathbb{R}$.

For example, Table 1 shows such a formalization for a graph model with two such template level features.

	Arguments	Formal entities	Attr.	Function
\mathcal{F}_e	$\langle \xi_1, \xi_2 \rangle$ $\langle T_v, T_v \rangle$	$[\xi_1, \xi_2]$ T_e	Exist	$f_\delta(z) = \mathbf{I}\{z = 1\}$
\mathcal{F}_t	$\langle \xi_1, \xi_2, \xi_3 \rangle$ $\langle T_v, T_v, T_v \rangle$	$[\xi_1, \xi_2]$ $[\xi_1, \xi_3]$ $[\xi_2, \xi_3]$ T_e, T_e, T_e	Exist Exist Exist	$f_3(z_1, z_2, z_3) = \mathbf{I}\{(z_1 = 1) \wedge (z_2 = 1) \wedge (z_3 = 1)\}$

Table 1: Example of two template-level features for a graph model. The first is a feature over single edges, and the second is one over triplets of coincident edges (triangles).

We view a template-level feature as a recipe for generating multiple instance-level features by applying different *bindings* of objects to the arguments. For example, in our three vertices instantiation, we could create instances of the feature \mathcal{F}_e such as $f_\delta(X_{\text{Exist}}([v_1, v_2]))$ and $f_\delta(X_{\text{Exist}}([v_1, v_3]))$. We now formally define this process.

Definition 2.5: Let \mathcal{F} be a template feature with components as in Definition 2.4, and let \mathcal{I} be an entity instantiation. A *binding* of \mathcal{F} is an ordered tuple of k entities $\beta = \langle e_1, \dots, e_k \rangle$ such that $e_i \in \mathcal{I}(T_i^q)$. A binding is *legal* if each entity in the binding is unique. We define

$$\text{Bindings}(\mathcal{F}) = \{ \beta \in \mathcal{I}(T_1^q) \times \dots \times \mathcal{I}(T_k^q) : \beta \text{ is legal for } \mathcal{F} \}$$

Given a binding $\beta = \langle e_1, \dots, e_k \rangle \in \text{Bindings}(\mathcal{F})$, we define the entity $\varepsilon_i | \beta$ to be the entity corresponding to ε_i when we assign e_i to the argument ξ_i . Finally, we define the *ground feature* $\mathcal{F}|_\beta$ to be the function over ω :

$$\mathcal{F}|_\beta(\omega) = f(\omega(X_{A_1}(\varepsilon_1 | \beta)), \dots, \omega(X_{A_j}(\varepsilon_j | \beta)))$$

For example, consider the binding $\langle v_1, v_2, v_3 \rangle$ for \mathcal{F}_t of Table 1. This binding is legal since all three entities are of the proper type and are different from each other. This binding defines the ground feature

$$\mathcal{F}_t|_{\langle v_1, v_2, v_3 \rangle}(\omega) = f_3(x_{\text{Exist}}([v_1, v_2]), x_{\text{Exist}}([v_1, v_3]), x_{\text{Exist}}([v_2, v_3]))$$

That is, $\mathcal{F}_t|_{\langle v_1, v_2, v_3 \rangle}(\omega) = 1$ iff there is a triangle of edges between the vertices v_1, v_2 , and v_3 . Note that each binding defines a ground feature. However, depending on the choice of feature function, some of these ground features might be equivalent. In our last example, the binding $\langle v_1, v_3, v_2 \rangle$ creates the same feature. While this creates a redundancy, it does not impact the usefulness of the language. We now have all the components in place.

Definition 2.6: A *Relational MRF scheme* \mathcal{S} is defined by a set of types \mathcal{T} , their attributes \mathcal{A} and a set of template

features $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$. A *model* is a scheme combined with a vector of *parameters* $\theta = \langle \theta_1, \dots, \theta_k \rangle \in \mathbb{R}^k$. Given an entity instantiation \mathcal{I} a scheme uniquely defines the universe of discourse $\mathcal{X}_{\mathcal{I}}$. Given all this together we can define the joint distribution of a full assignment ω as:

$$P(\omega : \mathcal{S}, \mathcal{I}, \theta) = \frac{1}{Z(\theta, \mathcal{I})} \exp \sum_{i=1}^k \theta_i \mathcal{F}_i(\omega) \quad (1)$$

where (with slight abuse of notation)

$$\mathcal{F}_i(\omega) = \sum_{\beta \in \text{Bindings}(\mathcal{F}_i)} \mathcal{F}_i|_{\beta}(\omega)$$

is the total weight of all groundings of the feature \mathcal{F}_i , and Z is the normalizing constant. ■

This definition of a joint distribution is similar to standard log-linear models, except that all groundings of a template feature share the same parameter [4].

3 Compact Approximate Inference

One broad class of approximate inference procedure are *variational methods* [12]. Roughly speaking, in such methods we approximate the joint distribution by introducing additional *variational* parameters. Depending on the particular method, these additional parameters can be thought of as capturing approximation of marginal beliefs about selected subsets of variables. Although the general idea we present here can be applied to almost all variational methods, for concreteness and simplicity we focus here on *loopy belief propagation* [16, 23] which is one of the most common approaches in the field.

To describe loopy belief propagation we consider the data structure of a *factor graph* [14]. A factor graph is a bipartite graph that consists of two layers. In the first layer, we have for each random variable in the domain a *variable node* X . In the second layer we have *factor nodes*. Each factor node ω is associated with a set \mathbf{C}_{ω} of random variables and a feature π_{ω} . If $X \in \mathbf{C}_{\omega}$, then we connect the variable node X to the factor node ω . Graphically we draw variable nodes as circles and factor nodes as squares (see Figure 2(a)).

A factor graph is *faithful* to a log-linear model if each feature is assigned to a node whose scope contains the scope of the feature. Adding these features multiplied by their parameters defines for each potential node ω a potential function $\pi_{\omega}[\mathbf{c}_{\omega}]$ that assigns a real value for each value of \mathbf{C}_{ω} . There is usually a lot of flexibility in defining the set of potential nodes. For simplicity, we focus now on factor graphs where we have a factor node for each ground feature.

For example, let us consider a model over a graph where we also depict the colors of the vertices. We create for each vertex v_i a variable node $X_{\text{Color}}(v_i)$ and for

each pair of vertices $[v_i, v_j]$ a variable node $X_{\text{Exist}}([v_i, v_j])$. We consider two template features - the triangle feature we described earlier, and a co-colorization feature that describes a preference of two vertices that are connected by an edge to have the same color. To instantiate the triangle feature, we go over all directed tuples of three vertices $\beta = \langle v_i, v_j, v_k \rangle \in \text{Bindings}(\mathcal{F}_t)$ and define ω_{β} with scope $\mathbf{C}_{\beta} = \{X_{\text{Exist}}([v_i, v_j]), X_{\text{Exist}}([v_i, v_k]), X_{\text{Exist}}([v_j, v_k])\}$. See Figure 2(a) to see such a factor graph for an instantiation of 4 vertices. This factor graph is faithful since each ground feature is assigned to a dedicated feature node.

Loopy belief propagation over a factor graph is defined as repeatedly updating messages of the following form:

$$m_{X \rightarrow \omega}(x) \leftarrow \prod_{\omega' : X \in \mathbf{C}_{\omega'}, \omega' \neq \omega} m_{\omega' \rightarrow X}(x)$$

$$m_{\omega \rightarrow X}(x) \leftarrow \sum_{\mathbf{c}_{\omega}(X)=x} \left(e^{\pi_{\omega}[\mathbf{c}_{\omega}]} \prod_{X \neq X' \in \mathbf{C}_{\omega}} m_{X' \rightarrow \omega}(x') \right)$$

where $\mathbf{c}_{\omega}(X)$ is the value of X in the assignment of values \mathbf{c}_{ω} to \mathbf{C}_{ω} . When these messages converge, we can define belief about variables as

$$b_{\omega}(\mathbf{c}_{\omega}) \propto e^{\pi_{\omega}[\mathbf{c}_{\omega}]} \prod_{X' \in \mathbf{C}_{\omega}} m_{X' \rightarrow \omega}(\mathbf{c}_{\omega}(X'))$$

where the beliefs over \mathbf{C}_{ω} are normalized to sum to 1. These beliefs are the approximation of the marginal probability over the variables in \mathbf{C}_{ω} [23].

Unfortunately, trying to reason about a network over 1000 vertices with the features we described earlier, will produce $\binom{1000}{2}$ variable nodes (one for each edge), $2 \cdot \binom{1000}{2}$ edge feature nodes and $3 \cdot \binom{1000}{3}$ triplet feature nodes². Building such a graph and performing loopy belief propagation with it is a time consuming task. However, our main insight is that we can exploit some special properties of this model for much efficient representation and inference. The basic observation is that the factor graphs for the class of models we defined satisfy basic symmetry properties.

Specifically, consider the structure of the factor graph we described earlier. An instantiation of graph vertices defines both the list of random variables and of features that will be created. Each feature node represents a ground feature that originates from a legal binding to a template feature. The groundings for an edge feature and for an edge random variable span two vertices, while the grounding of triplet feature covers three vertices. Since we are considering all legal bindings (*i.e.*, all 2-mers and 3-mers of vertices) while spanning the factor graph, each edge variable node will be included in the scope of 2 edge feature nodes and $(n-2) \cdot 3$ triplet feature nodes. More importantly,

²Since we defined the template feature using ordered tuples and our edges are defined using unordered tuples, we will have two features over each edge and three features over each triplet.

since all the edge variables have the same “local neighborhood”, they will also compute the same messages during belief propagation over and over again. We now formalize this idea and show we can use it to enable efficient representation and inference.

Definition 3.1: We say that two nodes in the factor graph have the same **type** if they were instantiated from the same template (either template attribute or template feature). ■

Given this definition, we can present our main claim formally:

Theorem 3.2: In every stage t of synchronous belief propagation that is initiated with uniform messages, if v_i, v_k are from the same type and also v_j, v_l are from the same type then $m_{v_i \rightarrow v_j}^t(x) = m_{v_k \rightarrow v_l}^t(x)$.

We start by proving the local properties of symmetry of the model:

Lemma 3.3: In a model created according to Definition 2.6, if two nodes in the factor graph have the same type, then they have the same **local neighborhood**. That is, they have the same number of neighbors of each type.

The proof of Theorem 3.2 is a direct consequence of Lemma 3.3 by induction over the stage of the belief propagation. We now turn to prove Lemma 3.3:

Proof: If v_i and v_j are feature nodes, then since they are of the same type, they are instantiations of the same template feature. From Definition 2.4 and Definition 2.5 we can see that this means that they are defined over variables from the same type. Since each feature is connected only to the variables in its scope, this proves our claim. However, if v_i and v_j are variable nodes, it suffices to show that they take part in the same kind of features, and in the same number of features of each such kind. Note that Definition 2.6 shows that we use all legal binding for each feature. For simplicity, we will assume that v_i is instantiated from the attribute of some basic type T (the proof in case it is a complex type is similar). We need to compute how many ground features contain v_i in their scope, and do not contain v_j . From Definition 2.5 we can see that all the legal bindings that include v_i and do not include v_j are legal also if we replace v_i with v_j . ■

After showing that many calculations are done over and over again, we now show how we can use a more efficient representation to enable much faster inference.

Definition 3.4: A **template factor graph** over a template log-linear model is a bi-partite graph, with one level corresponding to attributes and the other corresponding to template features. Each template attribute T that corresponds to a formal entity in some template feature \mathcal{F} is mapped to a *template attribute node* on one side of the graph. And each template feature is mapped to a *template feature node* on the other side of the graph. Each template attribute node is

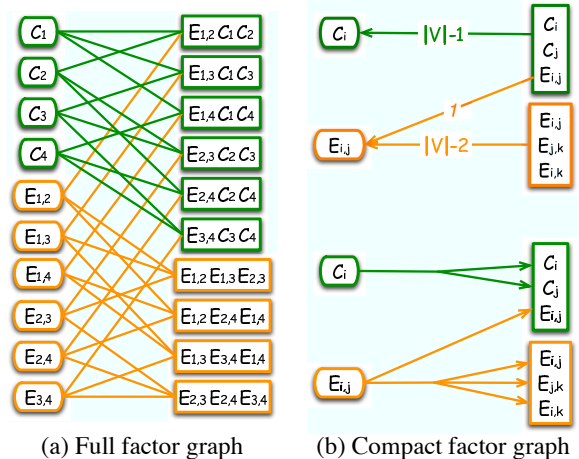


Figure 2: Shown are the full (a) and compact (b) factor graphs modeling a colored graph. We have basic types for colors and vertices, and a complex type for edges. We consider two template features - the triangle feature and a co-colorization feature. For clarity, $X_{\text{Exist}}([v_i, v_j])$ is shown as $E_{i,j}$ and $X_{\text{Color}}(v_i)$ is shown as C_i . Orange edges show the edges connected to edge variables and green edges are connected to color variables. $|V|$ shows the number of vertices in the graph.

connected with an edge to all the template feature nodes that contain this feature in their scope. A feature node needs to distinguish between its neighbors, since each message refers to a message about different variable. Hence, in the template factor graph we term an association to a variable inside a template feature node *port*. If a factor contains more than one variable of the same type, the corresponding edge splits to the corresponding ports when arriving to the factor node. In addition, each ground variable node takes part in many features that were instantiated by the same template feature with different bindings. Hence, each edge from a template feature node to a template attribute node in the template factor graph is assigned with a number indicating the number of repetitions it has in the full factor graph. ■

Figure 2(b) shows such a template factor graph for our running example.

Running loopy belief propagation on this template factor graph is straightforward. The algorithm is similar to the standard belief propagation only that when an edge in the template-graph represents many edges in the instance-level factor graph, we interpret this by multiplying the appropriate message the appropriate number of times. Since Theorem 3.2 shows that at all stages in the standard synchronous belief propagation the messages between nodes of the same type are similar, running belief propagation on the template factor graph is equivalent to running synchronous belief propagation on the full factor graph. However, we reduced the cost of representation and inference from being proportional to the size of the instantiated model, to be propor-

tional to the size of the domain. Specifically, this representation does not depend on the size of the instantiations and can deal with a huge number of variables.

4 Evaluation

4.1 Inference

We start by evaluating our method in inference tasks. We build a model representing a graph using the univariate and triangle features described in the previous section and perform inference with various parameter combinations. In the first step we consider instantiations of small graphs where we can also perform exact inference. We compared exact inference, MCMC (Gibbs sampling) [8], standard asynchronous belief propagation [23], and compact belief propagation on the template-level model. A simple way to compare inference results is by examining the marginal beliefs. Such a comparison is possible since in all methods the computed marginal probabilities for all edge variables were equal. Hence, Figure 3 shows a comparison of the marginal distributions over edge variables for different parameter settings and different inference methods. We observe that in small graphs the marginal beliefs are very similar for all inference methods. To quantify the similarity we calculate the relative deviation from the true marginal. We find that on average MCMC deviates by 0.0118 from the true marginal (stdev: 0.0159), while both belief propagation methods deviate on average by 0.0143 (stdev: 0.0817) and are virtually indistinguishable. However, in the graph over 7 vertices we notice that exact inference and MCMC are slightly different from the two belief propagation methods in the case where the univariate parameter is small and the triplet parameter is large (lower right corner).

An alternative measurement of inference quality is the estimate of the partition function. This is especially important for learning applications, as this quantity serves to compute the likelihood function. When performing loopy belief propagation, we can approximate the log-partition function using the Bethe approximation [23]. As seen in Figure 4, the estimate of the log partition function by belief propagation closely tracks the exact solution. Moreover, as in the marginal belief test, the two variants of belief propagation are almost indistinguishable. It is important to stress that running times are substantially different between the methods. For example, using exact inference with the 7 vertices graph (*i.e.*, one pixel in the matrices shown in Figure 3) takes 80 seconds on a 2.4 GHz Dual Core AMD based machine. Approximating the marginal probability using MCMC takes 0.3 seconds, standard BP takes 12 seconds, and compact BP takes 0.07 seconds.

On larger graphs, where exact inference and standard belief propagation are infeasible, we compare only the compact belief propagation and MCMC (see Figure 5). While there are some differences in marginal beliefs, we

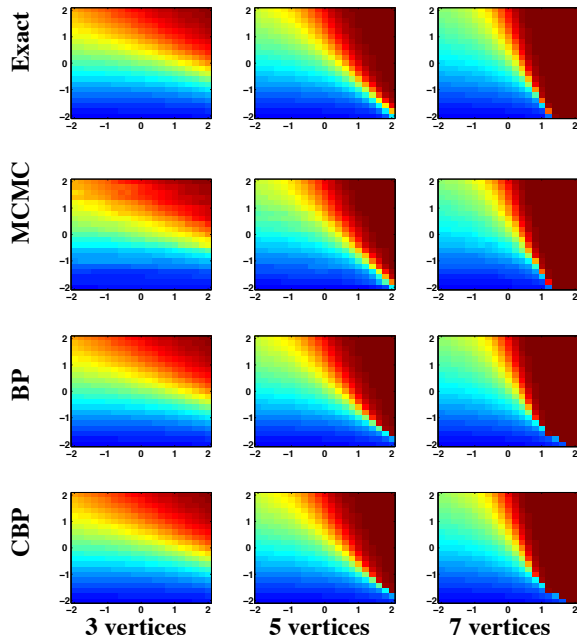


Figure 3: Comparison of inference methods via marginal beliefs. Each panel visualizes the the probability of an interaction when we vary two parameters: the univariate potential for interaction (y -axis) and the the potential over closed triplet (x -axis). The color indicates probability where blue means probability closer to 0 and red means probability closer to 1. The first row of panels shows exact computation, the second MCMC, the third standard asynchronous belief propagation, and the fourth our compact belief propagation.

see again that in general there is good agreement between the two inference procedures. As the graph becomes larger the gain in run-time increases. Since the mixing time of MCMC should depend on the size of the graph (if accuracy is to be conserved), running MCMC inference on a 100-node graph takes 5 minutes. As expected, compact BP still runs for only 0.07 seconds since it depends on the size of the scheme which remains the same. For protein-protein interaction networks over hundreds of vertices (see below), all inference methods become infeasible except for compact belief propagation.

4.2 Parameter estimation

Consider the task of learning the parameters $\Theta = \langle \theta_1 \dots \theta_k \rangle$ for each template feature. To learn such parameters from real-life data we can use the *Maximum Likelihood* (ML) estimation [4]. In this method we look for the parameters that best explain the data in the sense that they find $\text{argmax}_{\theta \in \Theta} p(\mathcal{D}|\theta)$. Since there is no closed form for finding the maximum likelihood parameters of a log-linear model, a common approach is to resort to greedy search methods such as gradient ascent. In such approaches an efficient calculation of the derivative is needed. The partial

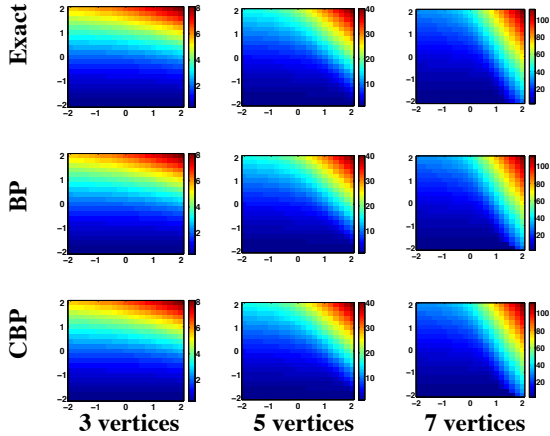


Figure 4: Comparison of inference methods for computing the log-partition function. Each panel visualizes the log-partition function (or its approximation) for different parameter setting (as in Figure 3). In the belief propagation methods, the log-partition function is approximated using the Bethe free energy approximation. On the first row is the exact computation, the second row shows standard asynchronous belief propagation and the third row shows our compact belief propagation.

derivative of the log likelihood $\ell(\mathbf{D})$ for a parameter θ_j that corresponds to a template feature \mathcal{F}_j can be described as:

$$\frac{\partial \ell(\mathbf{D})}{\partial \theta_j} = \hat{\mathbf{E}}[\mathcal{F}_j] - \mathbf{E}_\theta[\mathcal{F}_j] \quad (2)$$

Where $\hat{\mathbf{E}}[\mathcal{F}_j]$ is the number of times we actually see the feature j in \mathbf{D} , and

$$\mathbf{E}[\mathcal{F}_j] = \sum_{\beta \in \text{Bindings}(\mathcal{F}_j)} \mathbf{E}[\mathcal{F}_j | \beta]$$

is the sum of times we expect to see each grounding of the feature j according to Θ (see [4]). The first term is relatively easy to compute in cases where we learn from fully observed instances, since it is simply the count of each feature in \mathbf{D} . And the second term can be approximated efficiently by our inference algorithm.

To evaluate this learning procedure we start by generating samples from a model using a Gibbs sampler [8]. We then use these samples to estimate the original parameters using exact and approximate inference. In this synthetic context, we model a graph over seven vertices using only triplet (\mathcal{F}_t) and open chain (\mathcal{F}_c) features and try to recover the parameter of these features. As can be seen in Figure 6, using both approximate and exact inference retrieved parameter values that are close to these we used to generate the data. However, we can see that since the approximate and exact likelihoods create a different scenery, the trace of the exact search is much shorter, and retrieves better parameters.

We now proceed to learning a real-life model over interactions between proteins. We build on a model described

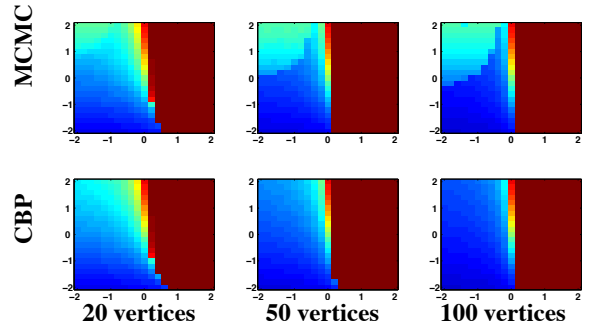


Figure 5: Comparison of approximate inference methods on larger graph instances. As before, we show the probability of an interaction as a function of parameter settings. On the first row is MCMC and the second row shows our compact belief propagation.

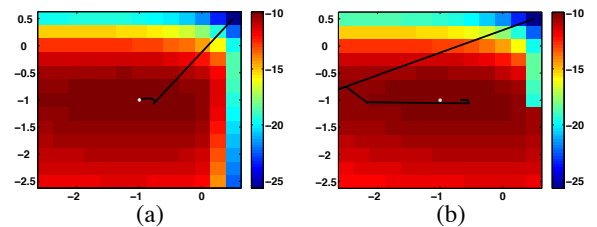


Figure 6: Learning trace of the parameters using exact (a) and approximate (b) inference on a 7 vertex graph. In both panels values of θ_{111} are shown on the x -axis while values of θ_{011} are shown on the y -axis. The dark line shows the advancement of the conjugate gradient learning procedure, and the bright asterisk in the middle shows the original parameters used for generating the samples. Color scale shows the exact and approximate log-likelihood respectively

in [11] for protein-protein interactions. This model is analogous to our running example, where the vertices of the graph are proteins and the edges are interactions. We define the basic type T_p for proteins and the complex type $T_i = [T_p, T_p]$ for interactions between proteins. As with edges, we consider the template attribute $X_e(T_i)$ that equals one if the two proteins interact and zero otherwise. We reason about an instantiation for a set of 813 proteins related to DNA transcription and repair [2]. We collected statistics over interactions between these proteins from various experiments [1, 7, 13, 15].

We adopt an incremental approach considering only the simplest template feature at the beginning and adding more complex features later on (this approach is somewhat similar to Della Pietra *et al.* [4]). We start by learning a model with only univariate features over interactions. As expected, the parameters we learn reflect the probability of an interaction in the data. We can now consider more complex features to the model by fixing the univariate parameter and adding various features. We start by adding two features, \mathcal{F}_t and \mathcal{F}_c that describe the closed triangle of

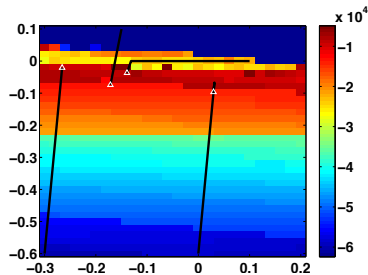


Figure 7: Exploration of the approximate log-likelihood landscape. In this example, the univariate parameter is fixed, the weights of two features over three interactions, triangle and chains, are varied. The x -axis shows the triangle parameter (θ_{111}) and the y -axis shows the chain parameter (θ_{011}). The dark lines show traces of conjugate gradient runs initiated from arbitrary starting points. The bright triangles mark the final parameter values returned by the algorithm.

interactions and open chain of interactions respectively.

Using our efficient inference approximation we can reevaluate the likelihood and its derivative for many parameter values and thereby gain an unprecedented view of the likelihood landscape of the model. For example, Figure 7 shows the log-likelihood calculated for a grid of parameter values and traces of a conjugate gradient learning procedure initialized from different starting points. We find that this view of the likelihood function is highly informative as it shows the influence of different parameter values on the model behavior. Specifically, the results show that the likelihood sensitivity to each parameter is quite different. This can be seen as a horizontal ridge in the upper part of the region, meaning that changes in θ_{111} have smaller effect on likelihood value than changes in θ_{011} . This behavior might reflect the fact that there are 3-times more occurrences of open chains than occurrences of closed triangles in the graph. Furthermore, our unique view of the likelihood landscape, and especially the horizontal ridge we see, illustrate that there is a strong relation between the parameters. As each of the gradient ascent runs converge to a different local maxima, we can use the landscape to determine whether this a consequence of rough landscape of the approximate likelihood or is due to redundancies in the parametrization that result in an equi-probable region.

We repeated the same exploration technique for other features such as colocalization of proteins [11], star-2 and star-3 [10], and quadruplets of interactions (results not shown). We find that the overall gain in terms of likelihood is smaller than in the case of triplet features. Again, we find that whenever one of the features is more abundant in the network, its influence on the approximate marginal beliefs and likelihood is much larger. In such cases the interesting region - where likelihood is high - narrows to a small range of parameter values of the abundant feature.

5 Discussion

We have shown how we exploit symmetry in relational MRFs to perform approximate inference at the template-level. This results in an extremely efficient approximate inference procedure. We have shown that this procedure is equivalent to synchronous belief propagation in the ground model. We have also empirically shown that on small graphs our inference algorithm approximates the true marginal probability very well. Furthermore, other approximation methods, such as MCMC and asynchronous BP yield inference results that are similar to ours. Note that other works show that synchronous and asynchronous belief propagation are not always equivalent [5].

Other works attempted to exploit relational structure for more efficient inference. For example, Pfeffer *et al.* [17] used the relational structure to cache repeated computations of intermediate terms that are identical in different instances of the same template. Several recent works [3, 18] derive rules as to when variable elimination can be performed at the template level rather than the instance level, which saves duplicate computations at the instance levels. These methods focus on speeding exact inference, and are relevant in models where the intermediate calculations of exact inference have tractable representations. These approaches cannot be applied to models, such as the ones we consider, where the tree-width is large, and thus intermediate results of variable elimination are exponential. In contrast, our method focuses on template level inference for approximate inference in such intractable models.

We stress that the main ideas developed here can be applied in other variational methods such as generalized belief propagation or structured mean field. Furthermore, it is clear that the class of relational models we defined is not the only one that has symmetry properties that can be exploited by our procedure. In fact, all the relational models that obey Lemma 3.3 can be run in template level. For example, it can be shown that a square wrap-around grid also obeys such symmetry.

The key limitation of our procedure is that it relies on the lack of evidence. Once we introduce evidence the symmetry is disrupted and our method does not apply. While this seems to be a serious limitation, we note that inference without evidence is the main computational step in learning such models from data. We showed how this procedure enables us to deal with learning problems in large relational models that were otherwise infeasible. Though the search space proves to be very difficult [10], our method allows us to perform many iterations of parameter estimation in different settings and thereby get a good overview of the likelihood landscape. This brings us one step closer towards successful modeling of networks using relational probabilistic models.

Acknowledgements

We thank Chen Yanover, Tal El-Hay, Gal Elidan, and the anonymous reviewers for helpful remarks on previous versions of this manuscript. Part of this research was supported by a grant from the United States-Israel Binational Science Foundation (BSF). Ariel Jaimovich is supported by the Eshkol fellowship from the Israeli Ministry of Science.

References

- [1] S. R. Collins, *et al.* Towards a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol Cell Proteomics*, 2007.
- [2] S. R. Collins, *et al.* Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, 2007.
- [3] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *IJCAI* 2005.
- [4] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.
- [5] G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *UAI* 2006.
- [6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI* 1999.
- [7] A. C. Gavin, *et al.* Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.
- [8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 721–741, 1984.
- [9] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *ICML* 2001.
- [10] M. S. Handcock. Assessing degeneracy in statistical models of social networks. Technical Report 39, University of Washington, 2003.
- [11] A. Jaimovich, G. Elidan, H. Margalit, and N. Friedman. Towards an integrated protein-protein interaction network: a relational Markov network approach. *J. Comput. Biol.*, 13:145–164, 2006.
- [12] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational approximations methods for graphical models. In *Learning in Graphical Models*, 1998.
- [13] N. J. Krogan, *et al.* Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, 440(7084):637–643, 2006.
- [14] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), 2001.
- [15] HW Mewes, J. Hani, F. Pfeiffer, and D. Frishman. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, 26:33–37, 1998.
- [16] K. Murphy and Y. Weiss. Loopy belief propagation for approximate inference: An empirical study. In *UAI* 1999.
- [17] A. Pfeffer, D. Koller, B. Milch, and K. Takusagawa. SPOOK: A system for probabilistic object-oriented knowledge representation. In *UAI* 1999.
- [18] D. Poole. First-order probabilistic inference. In *IJCAI* 2003.
- [19] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.
- [20] E. Segal, M. Shapira, A. Regev, D. Pe’er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2):166–176, 2003.
- [21] B. Taskar, A. Pieter Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI* 2002.
- [22] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *NIPS* 2004.
- [23] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2002-35, Mitsubishi Electric Research Laboratories, 2002.

4 Paper chapter: FastInf - an efficient approximate inference library

Ariel Jaimovich, Ofer Meshi, Ian McGraw and Gal Elidan.
Journal of Machine Learning Research 11:1733-1736, 2010.

FastInf: An Efficient Approximate Inference Library

Ariel Jaimovich

Ofer Meshi

*School of Computer Science and Engineering
Hebrew University of Jerusalem
Jerusalem, Israel 91904*

ARIELJ@CS.HUJI.AC.IL

MESHI@CS.HUJI.AC.IL

Ian McGraw

*Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA*

IMCGRAW@MIT.EDU

Gal Elidan

*Department of Statistics
Hebrew University of Jerusalem
Jerusalem, Israel 91905*

GALEL@HUJI.AC.IL

Editor: Cheng Soon Ong

Abstract

The FastInf C++ library is designed to perform memory and time efficient approximate inference in large-scale discrete undirected graphical models. The focus of the library is propagation based approximate inference methods, ranging from the basic loopy belief propagation algorithm to propagation based on convex free energies. Various message scheduling schemes that improve on the standard synchronous or asynchronous approaches are included. Also implemented are a clique tree based exact inference, Gibbs sampling, and the mean field algorithm. In addition to inference, FastInf provides parameter estimation capabilities as well as representation and learning of shared parameters. It offers a rich interface that facilitates extension of the basic classes to other inference and learning methods.

Keywords: graphical models, Markov random field, loopy belief propagation, approximate inference

1. Introduction

Probabilistic graphical models (Pearl, 1988) are a framework for representing a complex joint distribution over a set of n random variables $\mathcal{X} = \{X_1 \dots X_n\}$. A qualitative graph encodes probabilistic independencies between the variables and implies a decomposition of the joint distribution into a product of local terms:

$$P(\mathcal{X}) = \frac{1}{Z} \prod_i \psi_i(C_i),$$

where C_i are subsets of \mathcal{X} defined by the cliques of the graph structure and $\psi_i(C_i)$ are the quantitative parameters (potential functions) that define the distribution. Computing marginal probabilities and likelihood in graphical models are critical tasks needed both for making predictions and to facilitate learning. Obtaining exact answers to these inference queries is often infeasible even for relatively

modest problems. Thus, there is a growing need for inference methods that are both efficient and can provide reasonable approximate computations. Despite few theoretical guarantees, the Loopy Belief Propagation (LBP, Pearl, 1988) algorithm has gained significant popularity in the last two decades due to impressive empirical success, and is now being used in a wide range of applications ranging from transmission decoding to image segmentation (Murphy and Weiss, 1999; McEliece et al., 1998; Shental et al., 2003). Recently there has been an explosion in practical and theoretical interest in propagation based inference methods, and a range of improvements to the convergence behavior and approximation quality of the basic algorithms have been suggested (Wainwright et al., 2003; Wiegnerinck and Heskes, 2003; Elidan et al., 2006; Meshi et al., 2009).

We present the *FastInf* library for efficient approximate inference in large scale discrete probabilistic graphical models. While the library’s focus is propagation based inference techniques, implementations of other popular inference algorithms such as mean field (Jordan et al., 1998) and Gibbs sampling are also included. To facilitate inference for a wide range of models, *FastInf*’s representation is flexible allowing the encoding of standard Markov random fields as well as template-based probabilistic relational models (Friedman et al., 1999; Getoor et al., 2001), through the use of shared parameters. In addition, *FastInf* also supports learning capabilities by providing parameter estimation based on the Maximum-Likelihood (ML) principle, with standard regularization. Missing data is handled via the Expectation Maximization (EM) algorithm (Dempster et al., 1977).

FastInf has been used successfully in a number of challenging applications, ranging from inference in protein-protein networks with tens of thousands of variables and small cycles (Jaimovich et al., 2005), through protein design (Fromer and Yanover, 2008) to object localization in cluttered images (Elidan et al., 2006).

2. Features

The *FastInf* library was designed while focusing on generality and flexibility. Accordingly, a rich interface enables implementation of a wide range of probabilistic graphical models to which all inference and learning methods can be applied. A basic general-purpose propagation algorithm is at the base of all propagation variants and allows straightforward extensions.

A model is defined via a graph interface that requires the specification of a set of cliques $C_1 \dots C_k$, and a corresponding set of tables that quantify the parametrization $\psi_i(C_i)$ for each joint assignment of the variables in the clique C_i . This general setting can be used to perform inference both for the directed Bayesian network representation and the undirected Markov one.

2.1 Inference Methods

FastInf includes implementations of the following inference methods:

- Exact inference by the Junction-Tree algorithm (Lauritzen and Spiegelhalter, 1988)
- Loopy Belief Propagation (Pearl, 1988)
- Generalized Belief Propagation (Yedidia et al., 2005)
- Tree Re-weighted Belief Propagation (Wainwright et al., 2005)
- Propagation based on convexification of the Bethe free energy (Meshi et al., 2009).
- Mean field (Jordan et al., 1998)
- Gibbs sampling (Geman and Geman, 1984)

By default, all methods are used with standard asynchronous message scheduling. We also implemented two alternative scheduling approaches that can lead to better convergence properties (Wainwright et al., 2002; Elidan et al., 2006). All methods can be applied to both sum and max product propagation schemes, with or without damping of messages.

2.2 Relational Representation

In many domains, a specific local interaction pattern can recur many times. To represent such domains, it is useful to allow multiple cliques to share the same parametrization. In this case a set of template table parametrizations ψ_1, \dots, ψ_T are used to parametrize all cliques using

$$P(\mathcal{X}) = \frac{1}{Z} \prod_t \prod_{i \in I(t)} \psi_t(C_i),$$

where $I(t)$ is the set of cliques that are mapped to the t 'th potential. This template based representation allows the definition of large-scale models using a relatively small number of parameters.

2.3 Parameter Estimation

FastInf can also be used for learning the parameters of the model from evidence. This is done by using gradient-based methods with the Maximum-Likelihood (ML) objective. The library also handles partial evidence by applying the EM algorithm (Dempster et al., 1977). Moreover, FastInf supports L_1 and L_2 regularization that is added as a penalty term to the ML objective.

3. Documentation

For detailed instructions on how to install and use the library, examples for usage and documentation on the main classes of the library visit FastInf home page at: <http://compbio.cs.huji.ac.il/FastInf>.

Acknowledgments

We would like to acknowledge Menachem Fromer, Haidong Wang, John Duchi and Varun Ganapathi for evaluating the library and contributing implementations of various functions. This library was initiated in Nir Friedman's lab at the Hebrew University and developed in cooperation with Daphne Koller's lab at Stanford. AJ was supported by the Eshkol fellowship from the Israeli Ministry of Science. IM and GE were supported by the DARPA transfer learning program under contract FA8750-05-2-0249

References

- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–39, 1977.
- G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *UAI 2006*.

- G. Elidan, G. Heitz, and D. Koller. Learning object shape: From drawings to images. In *CVPR* 2006.
- N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI* 1999.
- M. Fromer and C. Yanover. A computational framework to empower probabilistic protein design. In *Bioinformatics*, pages 214–222, 2008.
- S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 721–741, 1984.
- L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *ICML* 2001.
- A. Jaimovich, G. Elidan, H. Margalit, and N. Friedman. Towards an integrated protein-protein interaction network. In *RECOMB*, 2005.
- M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational approximations methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, Dordrecht, Netherlands, 1998.
- S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 1988.
- R. McEliece, D. McKay, and J. Cheng. Turbo decoding as an instance of pearl’s belief propagation algorithm. *IEEE Journal on Selected Areas in Communication*, 16:140–152, 1998.
- O. Meshi, A. Jaimovich, A. Globerson, and N. Friedman. Convexifying the bethe free energy. In *UAI* 2009.
- K. Murphy and Y. Weiss. Loopy belief propagation for approximate inference: An empirical study. In *UAI* 1999.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, California, 1988.
- N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations with the GBP typical cut algorithm. In *ICCV* 2003.
- M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization for approximate estimation on loopy graphs. In *NIPS* 2002.
- M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Exact map estimates by (hyper)tree agreement. In *NIPS* 2002.
- M. J. Wainwright, T.S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.
- W. Wiegand and T. Heskes. Fractional belief propagation. In *NIPS* 2002.
- J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2005.

5 Paper chapter: Modularity and directionality in genetic interaction maps

Ariel Jaimovich*, Ruty Rinott*, Maya Schuldiner, Hanah Margalit and Nir Friedman.

In the 18th Conference for Intelligent Systems in Molecular Biology (*ISMB 2010*).

* These authors contributed equally to this work

Modularity and directionality in genetic interaction maps

Ariel Jaimovich^{1,2,†}, Ruty Rinott^{1,†}, Maya Schuldiner^{3,*}, Hanah Margalit^{2,*}
and Nir Friedman^{1,4,*}

¹School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, ²Department of Microbiology and Molecular Genetics, IMRIC, Faculty of Medicine, The Hebrew University of Jerusalem, Jerusalem 91120, ³Department of Molecular Genetics, Weizmann Institute of Science, Rehovot 76100 and ⁴The Alexander Silberman Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

ABSTRACT

Motivation: Genetic interactions between genes reflect functional relationships caused by a wide range of molecular mechanisms. Large-scale genetic interaction assays lead to a wealth of information about the functional relations between genes. However, the vast number of observed interactions, along with experimental noise, makes the interpretation of such assays a major challenge.

Results: Here, we introduce a computational approach to organize genetic interactions and show that the bulk of observed interactions can be organized in a hierarchy of modules. Revealing this organization enables insights into the function of cellular machineries and highlights global properties of interaction maps. To gain further insight into the nature of these interactions, we integrated data from genetic screens under a wide range of conditions to reveal that more than a third of observed aggravating (i.e. synthetic sick/lethal) interactions are unidirectional, where one gene can buffer the effects of perturbing another gene but not vice versa. Furthermore, most modules of genes that have multiple aggravating interactions were found to be involved in such unidirectional interactions. We demonstrate that the identification of external stimuli that mimic the effect of specific gene knockouts provides insights into the role of individual modules in maintaining cellular integrity.

Availability: We designed a freely accessible web tool that includes all our findings, and is specifically intended to allow effective browsing of our results (<http://compbio.cs.huji.ac.il/GIAnalysis>).

Contact: maya.schuldiner@weizmann.ac.il;
hanahm@ekmd.huji.ac.il; nir@cs.huji.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

A major goal in biology is to understand how thousands of genes act together to create a functional cellular environment. An emerging powerful strategy for investigating functional relations between genes involves high-throughput genetic interaction maps (Butland *et al.*, 2008; Byrne *et al.*, 2007; Collins *et al.*, 2007a; Fiedler *et al.*, 2009; Makhnevych *et al.*, 2009; Pan *et al.*, 2006; Roguev *et al.*, 2008; Schuldiner *et al.*, 2005; Segrè *et al.*, 2005; Tong *et al.*, 2001; Wilmes *et al.*, 2008), which measure the extent by which a mutation in one gene modifies the phenotype of a mutation in another. The interactions in these maps can be divided to *alleviating interactions*, where the defect of the double mutant is less than expected from

two independent effects, and *aggravating interactions*, where the defect of the double mutant is greater than expected from the single-gene perturbations. Such systematic mapping typically uncovers a large number of observed genetic interactions, which confounds straightforward interpretation. Despite the large number of published maps, a systematic methodology for extracting biological insights remains a major challenge.

Previous analyses of genetic interaction data have primarily focused on hierarchical clustering, resulting in many new discoveries in key cellular processes (Collins *et al.*, 2007a; Pan *et al.*, 2006; Schuldiner *et al.*, 2005). Nonetheless, hierarchical clustering has two major drawbacks: first, the similarity score between genes is based on their entire interaction profile (with all other genes) allowing large fraction of background interactions to dominate the similarity. Second, it does not directly extract meaningful groups of genes or interactions between such groups, preventing a system-level view of the interaction map. Both challenges were addressed by several methods. For example, the PRISM algorithm (Segrè *et al.*, 2005) uses monochromatic interactions (i.e. solely aggravating or solely alleviating) within and between groups of genes to define pathways (Fig. 1A). However, this algorithm, which was evaluated on simulated interaction maps, fails on actual data from large-scale maps due to the added complexity in real cellular systems and assay noise (data not shown). Biclustering is another approach that was suggested as an alternative to hierarchical clustering, aiming to identify local signatures of functional modules in the genetic interaction maps (Pu *et al.*, 2008). While this approach identifies many modules of genes, it does not eliminate their overlap, hampering the generation of one coherent network structure describing both the intra- and inter-modular interactions. One possible way to overcome these drawbacks is by adding different types of data or additional constraints. For example, methods that combine physical protein–protein interactions in the analysis of genetic interaction data identify functional modules with high precision (Bandyopadhyay *et al.*, 2008; Kelley and Ideker, 2005; Ulitsky *et al.*, 2008). However, the requirement for physical interaction data limits such approaches to protein sets and organisms where such data exist, and may miss many functional pathways that are not mediated by protein complexes (e.g. metabolic pathways).

Here, we introduce an automated approach that builds a concise representation of large-scale genetic interaction maps. Toward this goal, we relied on previous observations that complexes and pathways induce signatures in the form of monochromatic cliques and bi-cliques (Fig. 1A; Beyer *et al.*, 2007; Boone *et al.*, 2007; Segrè *et al.*, 2005). Our method seeks to find an organization that is globally coherent, in the sense that genes are organized into a hierarchy of modules. Moreover, our method requires that the

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First authors.

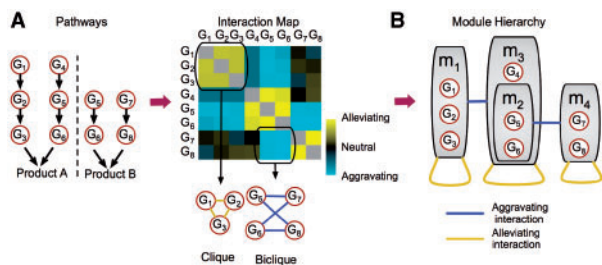


Fig. 1. Modularity of genetic interactions. (A) Pathway architecture (left) leads to expected patterns of genetic interactions between genes (right). Each row/column represents the genetic interactions of a specific gene with all other genes. Among these there are subsets of interactions that can be represented as monochromatic cliques and bicliques. (B) Monochromatic interactions can be captured by edges within and between modules (grey boxes) organized in a hierarchical structure.

interactions between these modules will account for a large portion of the data. We show how the resulting representation facilitates better understanding of the underlying cellular phenomena. In turn, we use these insights to shed light on the function of concrete cellular pathways and also to provide information on the overall organization of the network. We demonstrate how integration of data from genetic screens for reduced fitness under various conditions results in automatic creation of biological insights into the functional role of gene modules.

2 HIERARCHY OF INTERACTING MODULES

Our basic premise is that a good hierarchical organization is defined by a trade-off between succinct description of the network on one hand, and capturing as much of the interactions in the map on the other hand. To capture this quality, we devised a score based on the minimum description length (MDL) principle (Rissanen, 1983) and devised an iterative procedure that optimizes this score.

2.1 Hierarchical representation

The hierarchical representation consists of two parts. The first is a hierarchy of modules. Briefly, a hierarchy is a set \mathcal{M} of *modules*, such that each module m is associated with a subset of genes $\text{Genes}(m)$ and a parent module $\text{Parent}(m) \in \mathcal{M} \cup \{\epsilon\}$, where ϵ represents a null module (i.e. the module is a root). We say that a module m' is an *ancestor* of m if $m' = \text{Parent}^k(m)$ for some $k \geq 1$. The hierarchy is legal if for every $m, m' \in \mathcal{M}$ such that $m' = \text{Parent}(m)$, we have that $\text{Genes}(m) \subset \text{Genes}(m')$, and moreover $\text{Genes}(m) \cap \text{Genes}(m') \neq \emptyset$ if and only if m is an ancestor of m' or vice versa. In the hierarchy of Figure 1B, we have four modules, so that $\text{Genes}(m_1) = \{G_1, G_2, G_3\}$, $\text{Genes}(m_2) = \{G_5, G_6\}$, $\text{Genes}(m_3) = \{G_4, G_5, G_6\}$, and $\text{Genes}(m_4) = \{G_7, G_8\}$. In this example, $\text{Parent}(m_1) = \text{Parent}(m_3) = \text{Parent}(m_4) = \epsilon$, and $\text{Parent}(m_2) = m_3$.

The second component of the hierarchy describes a set \mathcal{E} of *edges* between modules. An edge can be of two types, alleviating (denoted in yellow in our figures) or aggravating (denoted in blue). Each edge represents a type of genetic interactions that is common for the members of the modules linked by the edge. Formally, an edge $m_1 \leftrightarrow m_2$ represent the set $\text{Int}(m_1 \leftrightarrow m_2) = \text{Genes}(m_1) \times \text{Genes}(m_2)$ of genetic interactions. Edges in the hierarchy can be self-edges,

in which case they induce a clique of interactions, or between two different modules in which case they induce a bi-clique of interactions. In the example of Figure 1B, we have the alleviating edges $m_1 \leftrightarrow m_1$, $m_3 \leftrightarrow m_3$, $m_4 \leftrightarrow m_4$, and the aggravating edges $m_1 \leftrightarrow m_3$ and $m_2 \leftrightarrow m_4$. These edges represent the interactions described in the interaction matrix of Figure 1A.

2.2 Minimal description length score

We use the MDL principle (Rissanen, 1983) to score the quality of module hierarchy as a guide for lossless encoding of the genetic interaction map. Conceptually, imagine that we need to transmit the genetic interaction map over a channel and search for the encoding that would require the fewest bits. Under this principle, the length of the transmission is a proxy for the quality of the representation, with a shorter encoding denoting a better representation.

The application of this principle involves deciding how we encode the interactions in the map. When we do not have any organization of the map, we use the same codebook for each interaction. Since weak interactions are much more abundant than strong ones, their code words will be shorter (Cover and Thomas, 2001). Thus, we will incur a penalty for strong interactions. When we have a module hierarchy, we can use a different codebook for each edge in the hierarchy and an additional codebook for background interactions. This allows us to exploit a group of monochromatic interactions for efficient encoding by a codebook that assigns strong interactions of the appropriate short codewords. The benefit from covering a large portion of the map with coherent edges is offset by the cost of transmitting the codebooks themselves, which involves coding the hierarchical organization and the edges with their signs. Thus, when evaluating a possible organization of the genetic interaction map there is a trade-off between the coverage of interactions and the number of modules and edges.

Formally, if we denote the genetic interaction map by D and the hierarchical organization by $(\mathcal{M}, \mathcal{E})$ then the MDL score consists of two main terms:

$$S(D; \mathcal{M}, \mathcal{E}) = \text{DL}(\mathcal{M}, \mathcal{E}) + \text{DL}(D | \mathcal{M}, \mathcal{E})$$

where $\text{DL}(\mathcal{M}, \mathcal{E})$ is the description length of the hierarchical organization and $\text{DL}(D | \mathcal{M}, \mathcal{E})$ is the description length of the interactions, given that we already encoded the hierarchy. We start with the first term, $\text{DL}(\mathcal{M}, \mathcal{E})$. Here, we need to encode the module hierarchy (which module is the parent of each module), the assignment of genes to modules and the list of edges. This is a relatively straightforward encoding using standard MDL practices.

The second term represents how to describe the genetic interaction map once we know the modular organization. Standard results in information theory (Cover and Thomas, 2001) show that if the frequency of each word is $p(w)$, then the optimal codebook is one where encoding a word w is of length $-\log_2 p(w)$. Thus, in each codebook we use the distribution of the strengths of interactions covered by an edge to build an efficient codebook. We assume that the different values are distributed according to a Gaussian distribution. Thus, the encoding length is the minus log-probability (or likelihood) of the data given the parameters of each Gaussian codebook (i.e. the closer the distribution is to its parametric description, the score is higher). To this length, we add the number of bits needed to encode the parameters of each distribution. To calculate the encoding length, for each edge $e \in \mathcal{E}$ we estimate the

maximum likelihood parameters, (μ_e, σ_e) . In addition, we estimate the background distribution (μ_b, σ_b) . We then define

$$\begin{aligned} \text{DL}(D|\mathcal{M}, \mathcal{E}) = & - \sum_{e \in \mathcal{E}} \sum_{(i,j) \in \text{Int}(e)} \log_2 p(I_{i,j}|\mu_e, \sigma_e) \\ & - \sum_{(i,j) \in \text{Bg}} \log_2 p(I_{i,j}|\mu_b, \sigma_b) \\ & + \sum_{e \in \mathcal{E}} \log_2 |\text{Int}(e)| + \log_2 |\text{Bg}| \end{aligned}$$

where $p(I_{i,j}|\mu, \sigma)$ is the likelihood of the genetic interaction score $I_{i,j}$ according to the Gaussian $N(\mu, \sigma^2)$, Bg is the set of interactions that do not belong to any edge in \mathcal{E} , and $\log_2(|\text{Int}(e)|)$ is the encoding length of the parameters for the edge. Thus, we score interactions in their specific context (either inside an edge or in the background).

For practical concerns, we restrict the network to include only coherent edges. Thus, we require that an edge satisfies $|\mu_e| - \sigma_e > \alpha$, where α is a strictness parameter (which we set to 1 in the results below). If this is not the case, the network receives a large penalty which effectively excludes it from consideration.

2.3 Constructing module hierarchy

Given a genetic interaction map D , we want to find the module hierarchy that minimizes the MDL score. This problem is non-trivial as the search space is huge. To address this we combine two ideas. First, we use hierarchical clustering to get a good initial guess for our hierarchical organization. Second, once we have a reasonable initial candidate, a heuristic search procedure can perform ‘local’ improvements to find a much better one. Our procedure implements these ideas by performing the following steps.

Clustering: we cluster the genetic interaction map using hierarchical clustering with uncentered Pearson correlation (Eisen *et al.*, 1998). This results in a dendrogram, which in our terminology is a detailed hierarchy, where each internal node defines a group of genes that correspond to the leaves in its sub-tree and each pair of such internal nodes defines a rectangle in the clustered matrix (Fig. 2a).

Identifying edges: treating the dendrogram as an initial hierarchy of modules, the procedure traverses overall pairs of internal nodes in the dendrogram and in a greedy fashion adds modules and edges as long as they increase the MDL score. At this stage, we have a very large number of modules and some number of edges. We then prune modules that do not participate in edges (while maintaining the ancestral relationships between the remaining modules). This results in a hierarchy that summarizes the initial clustering (Fig. 2b).

Greedy improvements: to re-evaluate and refine the modular structure, the procedure performs a heuristic search by evaluating local changes to the modular organization. These local changes include: addition/removal of a gene to/from an existing module, merging a module with its parent, transferring an edge from a module to its parent (or vice-versa) and addition/removal of an edge. Each of these local changes is evaluated and based on their score the procedure decide which one to apply. We use a best-first-search heuristic combined with a TABU list (Glover *et al.*, 1993) to avoid revisiting explored networks and thus escape local maxima. This search leads to a refined model (Fig. 2c).

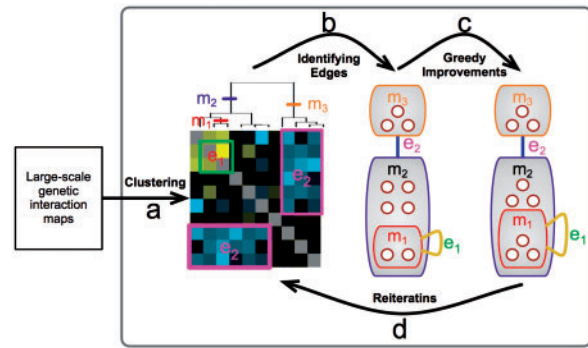


Fig. 2. Outline of our iterative algorithm. After clustering the interactions (left) our procedure identifies modules of genes in the clustering hierarchy that define monochromatic on-diagonal squares (e_1) and off-diagonal rectangles (e_2), resulting in a hierarchical organization of genes into modules (middle). Next, the module graph is refined by a series of local changes (e.g. moving one gene from m_2 to m_1 ; right). At the end of each iteration (bottom arrow), we re-cluster the genetic interaction matrix while maintaining the identified modules. These steps are iterated until convergence.

Reiterations: to find structures that might elude local search steps, the procedure iterates by returning to the first step. In each re-iteration, we re-cluster the genetic interaction map while conserving the module hierarchy from the previous step. That is, we allow only agglomerative steps that do not break existing modules into separate subunits. This constraint forces the resulting clustering to maintain the found structure, but it can identify new sub-modules as well as new modules of genes that are not assigned to a module. These iterations are repeated until convergence (in score) (Fig. 2d).

2.4 Application to genetic interaction maps in *Saccharomyces cerevisiae*

We applied our methodology to two large-scale genetic interaction maps in the budding yeast *S. cerevisiae*. The first contains genes localized to the Early Secretory Pathway (ESP; Schuldiner *et al.*, 2005) and the other comprises genes involved in Chromosome Biology (CB; Collins *et al.*, 2007b). This procedure automatically constructed a hierarchical organization of modules in both: in the ESP map it identified 113 modules covering 264 genes (out of 424) and in the CB map it identified 242 modules covering 487 genes (out of 743). Most of these modules represent functionally coherent groups of genes (ESP: 76/113, CB: 193/242; Appendix A in the Supplementary website), such as physical complexes (e.g. Mediator subunits, HIR complex, SAS complex) and functional pathways (e.g. *N*-linked glycosylation, chromatid cohesion). Inter- and intra-module interactions correspond to a large fraction of the interactions in the original maps, particularly the high confidence ones (Fig. 3A and B). In addition, the edges we capture are also coherent in the sense that most interactions covered by alleviating edges have positive interaction scores and most interactions covered by aggravating edges have negative scores (Fig. 3C and D). Thus, the modular organization of the genetic interactions faithfully captures a large portion of these maps.

The hierarchical nature of the network allows the definition of large modules with more general functions that contain sub-modules with more specific functions, which are distinguished by sets of unique interactions. For example, module ESP-98 comprises eight

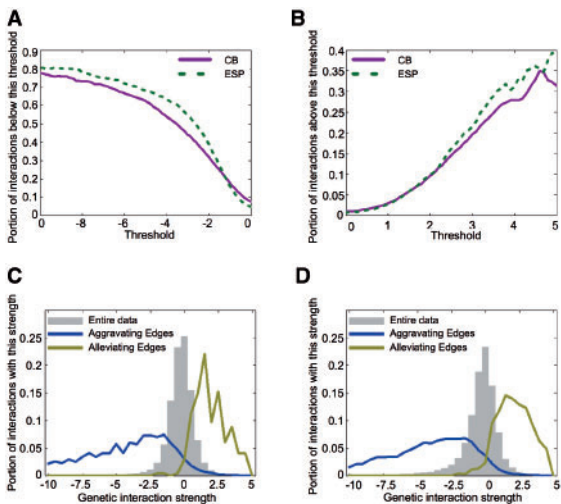


Fig. 3. Edges capture most interactions. (A) Coverage of aggravating interactions by our network (y-axis) as a function of threshold for EMAP score (x-axis). Magenta solid lines and green dashed lines show results for CB and ESP networks, respectively. (B) Coverage of alleviating interactions. (C) Coherence of aggravating and alleviating edges in the CB network. Shown is a histogram (y-axis) of EMAP scores (x-axis) for interactions covered by aggravating and alleviating edges in blue and yellow, respectively. Histogram for the entire data is shown in grey. (D) Coherence of edges in our ESP network.

genes that take part in the maturation of glycoproteins within the ER lumen (Fig. 4). Specifically, these genes encode the sequential enzymes adding on sugar moieties to a synthesized polysaccharide chain. Our analysis identified two sub-modules that correspond to two distinct stages in this process: one module (ESP-97) involves genes encoding proteins that transfer mannose residues to the nascent chain, and the second module (ESP-96) involves genes that subsequently transfer glucose residues to the nascent chain (Helenius and Aebi, 2004). This division was obtained automatically, based on interactions that are specific to each of these sub-modules (Fig. 4). Notably, the protein products of genes in these two modules do not form physical complexes, and thus could not be identified by methods that use protein–protein interactions to define the modules. In addition, this subdivision was not obtained by solely applying hierarchical clustering methods (Schuldiner *et al.*, 2005).

2.5 Comparison to other methods

Comparing our method to previous methods for analysing genetic interaction maps is difficult due to the different focus of the various methods. A common theme to most methods is the determination of gene modules. Although this is only one aspect of our analysis, we compared our module list to modules found by other studies of the CB map (Bandyopadhyay *et al.*, 2008; Pu *et al.*, 2008; Ulitsky *et al.*, 2008). Comparing to the methods of Bandyopadhyay *et al.* (2008) and Ulitsky *et al.* (2008, Fig. 5A and B), we find many more modules (242 modules compared with 91 and 62, respectively), covering more genes (487 genes compared with 374 and 313, respectively).¹ In addition, many of these modules are

¹When comparing to Bandyopadhyay *et al.* (2008) we considered only modules with more than one gene.

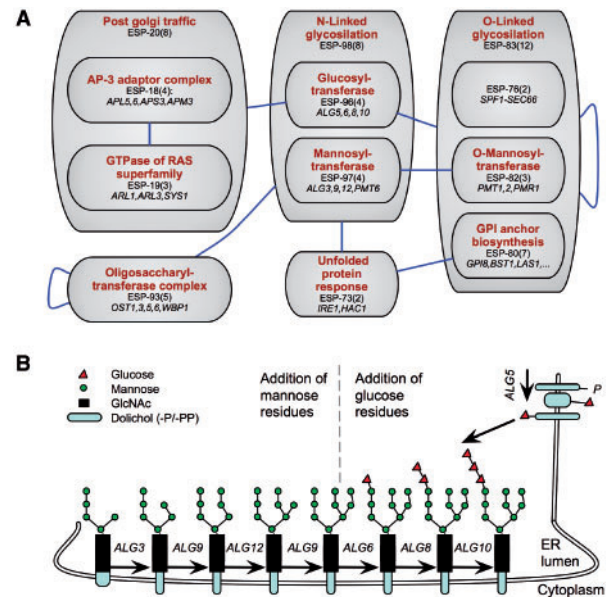


Fig. 4. Hierarchical organization of modules represents functional hierarchy. (A) Modules are denoted by grey boxes (red labels denote functional assignment based on annotations; black labels denote the name of each module and in parentheses the number of genes included in it). Blue edges between modules indicate that these modules create aggravating bicliques. Module ESP-98 contains eight genes related to N-linked glycosylation. It is further divided into two sub-modules (ESP-96 and ESP-97), each identified by different interactions, which have more specific functions. (B) Schematic view of the N-linked glycosylation pathway (adapted from Helenius and Aebi, 2004). Inside the ER lumen, four mannose residues (green circles) are added to Man5GlcNAc2 by Alg3, Alg9 and Alg12 (comprising module ESP-97). In turn, three glucose residues (red triangles) are added by Alg5, Alg6, Alg8 and Alg10 (comprising module ESP-96).

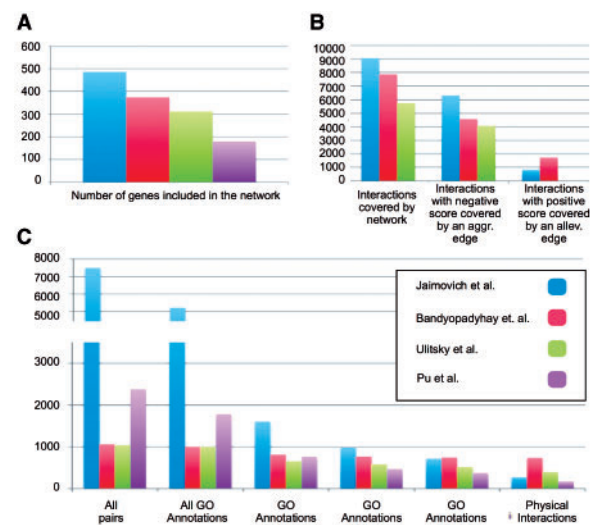


Fig. 5. Comparison to other methods: bar charts showing how many genes (A) and interactions (B) are covered by each method. (C) Bar chart showing how many of the protein pairs that are in the same module share a GO function annotation, or physically interact with each other. GO annotations are divided into categories according to the number of genes in the annotation.

not enriched with physical protein–protein interactions, yet have a coherent function. Furthermore, our approach is also applicable to other systems, in which the protein–protein interaction data is very sparse (such as in the ESP dataset) or in organisms in which it does not exist. When comparing our results to those of Pu *et al.* (2008) who finds 298 overlapping modules covering 181 genes, we see that we find similar numbers of modules organized in a global hierarchy and covering more genes. However, these advantages come at the price of lower precision (Fig. 5C). Yet, as the larger modules at the top of the hierarchy might correspond to more global functions, their enrichment in more general GO terms is reasonable. We conclude that each of the methods strikes a different trade off between precision, sensitivity and global coherence.

3 UNCOVERING UNIDIRECTIONAL COMPENSATION

Strikingly, a relatively large number of the gene pairs exhibit genetic interactions, especially aggravating ones. We find that aggravating interactions play a major role in the definition of many modules (e.g. 150 of the 242 modules in the CB network are defined solely based on aggravating interactions). Aggravating interactions are commonly interpreted as an indication of bidirectional compensation, where each gene can compensate for the absence of the other by performing a similar function. However, in many cases this explanation cannot account for the observed patterns of aggravating interactions and the large number of such interactions between genes with distantly related functions.

An alternative explanation (Boone *et al.*, 2007; Pan *et al.*, 2006) is that one gene is crucial for functions that compensate for the abnormal cellular state resulting from the loss of the other gene. In this scenario, termed *unidirectional compensation*, the relationship between the genes is asymmetric in the sense that one gene can compensate for the loss of the other but not vice versa. We refer to the gene whose knockout causes the perturbation as the *upstream gene* and to the compensating gene as the *downstream gene*. While examples for this type of interpretation have been shown on existing data (Pan *et al.*, 2006), no systematic test was carried out to identify the aggravating interactions that can be explained by such unidirectional interpretation and to assess their fraction within the observed aggravating interactions.

3.1 Identifying unidirectional compensation

Our premise is that we can identify unidirectional compensation by comparing the perturbation of a putative upstream gene with perturbations caused by external stimuli. We say that an external stimulus (e.g. a drug or an environmental insult) *phenocopies* a gene deletion if the genes required for coping with the stimulus are the same ones required to compensate for the perturbation of the upstream gene. Stated in terms of available data, this definition implies a significant overlap between the genes whose knockout lead to sensitivity to the stimulus and these that have aggravating interactions with the upstream gene. Moreover, genes in this overlap are downstream to the specific upstream gene. By establishing such phenocopy relations, we implicate unidirectional interactions from the upstream genes and their matching downstream genes.

For example, deletion of the *CHL1* gene leads to abnormal chromosome segregation similar to the damage caused by external

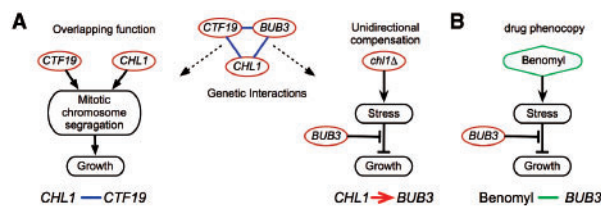


Fig. 6. Identifying unidirectional interactions. **(A)** An example of aggravating interactions (middle) that might be due to different mechanisms. Both *CHL1* and *CTF19* genes (red ellipses) have functions related to sister chromatid pairing during the S-phase. Thus, their aggravating interaction (denoted by a blue line) might be a result of their overlapping functions (left). However, the aggravating interactions of *CHL1* and *BUB3*, which is part of the spindle assembly checkpoint, is more likely the result of a different mechanistic reason (denoted by a directed red arrow; right), where the lack of a gene (i.e. *chl1*Δ) induces abnormal chromosome segregation, that requires the activation of the spindle assembly checkpoint including *BUB3*. **(B)** Yeast cells exposed to benomyl (denoted by a green diamond) show the same sensitivity to *BUB3* perturbation as the *chl1*Δ strain, suggesting that *chl1*Δ background causes a stress similar to exposure to benomyl.

microtubule depolymerizing agents (e.g. benomyl). In turn, the deletion strain of *bub3*Δ shows growth retardation under benomyl. Thus, we interpret the aggravating interaction between *CHL1* and *BUB3* as resulting from unidirectional compensation, where *CHL1* is the upstream gene and *BUB3* is the downstream gene (Fig. 6). Indeed, this interpretation is conceivable, as *Chl1* is involved in sister chromatid pairing during the S phase, and *Bub3* is part of the spindle assembly checkpoint, in charge of delaying anaphase in cases of abnormal spindle assembly.

When elaborating this reasoning we have to be careful not to confuse unidirectional compensation with *dosage effect*: if a gene phenocopies a stimulus, we might expect to see that its deletion amplifies the effect of this stimulus, showing higher sensitivity to its application (loosely stated, higher dosage of the stimulus). In such cases, we might mistakenly implicate an upstream gene to be downstream to another gene that also phenocopies the same stimulus. However, in such situations we will, by definition, identify bidirectional interactions where one gene is both upstream and downstream to another gene. Thus, we can detect these situations, and distinguish them from a proper unidirectional compensations.²

The reasoning we outline here (and apply below) detects, up to usual concerns about experimental or statistical noise, asymmetries of aggravating interactions with respect to phenotypes of external stimuli. This is a well-defined and clear criterion. A more ambitious step is to deduce from this asymmetry directionality in the underlying biological mechanisms. In our example of *CHL1* and *BUB3*, we have strong intuitions about the causal direction (as sister chromatid pairing precedes spindle assembly). In other cases, the underlying causality is much murkier. Moreover, we can imagine external perturbations that will lead to opposite asymmetry. For example, if a certain drug targets in a specific manner the spindle assembly checkpoint, we would detect asymmetric behavior of *CHL1* and *BUB3* to it, but in the opposite direction. This thought exercise implies that we need to be careful about deducing

²We estimate that up to five percent of unidirectional interactions are actually caused by dosage effect but were not identified as such since not all the genes were tested in all the screens (data not shown).

directionality in the underlying biology. However, we believe it is reasonable to assume that in most cases external perturbations are ones that causes cellular imbalances or stress conditions rather than disable mechanisms that cope with such situations.

3.2 Application to genetic interaction maps in *S.cerevisiae*

To systematically detect unidirectional compensation, we collected data from genetic screens that measured growth of yeast deletion strains under various external conditions and insults compared to YPD conditions (Bennett *et al.*, 2001; Dudley *et al.*, 2005; Giaever *et al.*, 2002; Hillenmeyer *et al.*, 2008; Parsons *et al.*, 2004, 2006). We considered deletion strains from both homozygote diploid and haploid deletions. We converted all measurements into a binary score, by defining genes with growth defects as those that passed the threshold defined by the authors of each study (for a detailed description of how we handled each dataset see Appendix B in the Supplementary website).

This process resulted in listing for each external stimulus the repertoire of deletion strains that display a growth defect in its presence. In a similar manner, each gene deletion defines a list of genes that are sensitive to its deletion, i.e. display aggravating interactions with it (using the same threshold, -2.5 , as Collins *et al.*, 2007a; Schuldiner *et al.*, 2005). We then define a unidirectional compensation between genes X and Y (associated with external perturbation P) if (i) there exists an aggravating interaction between X and Y ; (ii) the perturbation of Y leads to sensitivity to the external perturbation P ; (iii) X has aggravating interactions with a significant number of genes whose perturbations cause sensitivity to the perturbation P (using hyper-geometric test with FDR of 0.1); and (iv) at least one of the conditions 2 or 3 do not hold on the opposite direction (when switching the roles of X and Y).³

We applied this procedure to the CB and ESP genetic interaction maps and found 348 gene deletions that are phenocopied by at least one external stimulus. These stimuli include a wide range of external perturbations that match the nature of the specific data set analyzed. For example, many external stimuli corresponding to gene deletions in the CB map include agents causing DNA damage and microtubule depolymerization, while the stimuli related to the ESP map mostly include agents causing protein synthesis and glycosylation inhibition (see Supplementary website). To our surprise, more than one-third of the aggravating genetic interactions (CB: 4659/11539; ESP: 1036/2718) could be explained by unidirectional compensation.

4 ELUCIDATING THE FUNCTION OF CELLULAR PATHWAYS

We next asked whether unidirectional compensation can also be assigned within the modular hierarchy in terms of upstream and downstream modules. Toward this end, we incorporated these unidirectional interactions into our hierarchical organization of interacting modules. We annotated an aggravating edge between two modules as caused by unidirectional compensation if the majority

³To measure the statistical significance of the interactions we found, we created a random permutation of the names of the genes in the genetic interaction screen, and repeated the procedure described above. In 10 repeats, no significant overlaps between genes and external stimuli were found, thus no unidirectional interactions were identified.

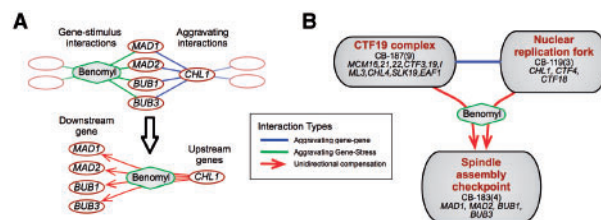


Fig. 7. Inter-module unidirectional interactions. **(A)** Systematic identification of unidirectional interactions: a systematic search discovers cases of statistically significant overlap between patterns of gene sensitivities under specific external stimuli (green lines) and the aggravating partners of specific genes (blue lines). We annotate these aggravating interactions as unidirectional, and denote them by red arrows directed from the upstream gene (whose deletion causes the cell perturbation) to the downstream genes (which deal with the particular cell perturbation). **(B)** All inter-module aggravating edges were scanned and searched for potential unidirectional edges. If the majority of the interactions involved in an inter-module aggravating edge are marked as consistent unidirectional interactions (corresponding to the same external stimulus and in the same direction), this edge was annotated as a unidirectional edge with respect to the specific external stimulus (green diamond).

of interactions between these modules are unidirectional and share the same context (i.e. have the same directionality and are related to the same external stimulus; Fig. 7A; Supplementary website). By requiring consistent unidirectional interactions between modules, this incorporation also removes potential errors in the annotation of unidirectional interactions (Supplementary website). We find that this designation elucidates the cellular role of modules and their interactions. Coming back to our previous example, we find that perturbations of modules CB-119 and CB-187 lead to stress conditions similar to those caused by microtubule de-polymerizing agent benomyl (Fig. 7B). Our analysis identified module CB-183 as downstream to benomyl-like stress caused by mutations of genes in CB-119 and CB-187. Indeed, the protein products of the genes in CB-119 and CB-187 are components of the machinery responsible for the correct distribution of chromosomes during cell division (Hanna *et al.*, 2001; Measday *et al.*, 2002). By de-polymerizing microtubules that create the spindle fibres, benomyl attacks a crucial component of this process. Finally, the genes in module CB-183 participate in the spindle assembly checkpoint that delays the onset of anaphase in cells with defects in mitotic spindle assembly (Nasmyth, 2005). This example demonstrates the power of our approach in automatically providing biological insights into the function of the genes in various modules.

The concise representation of the observed genetic interactions as edges within and between modules, in combination with the specific interpretation of many aggravating edges as caused by unidirectional compensations, pinpoints novel functions of modules that are not readily apparent from clustering of genetic interactions alone. The results of our automatic search provide an elaborate network of such inter- and intra-module edges, thus, we constructed a web-tool providing a user-friendly interface to browse our results in an effective manner (Supplementary website).

For example, examining unidirectional edges related to DNA damage agents, such as hydroxyurea and camptothecin, we find multiple upstream and downstream modules (Fig. 8A). A notable downstream module (CB-137) comprises three sub-modules; of

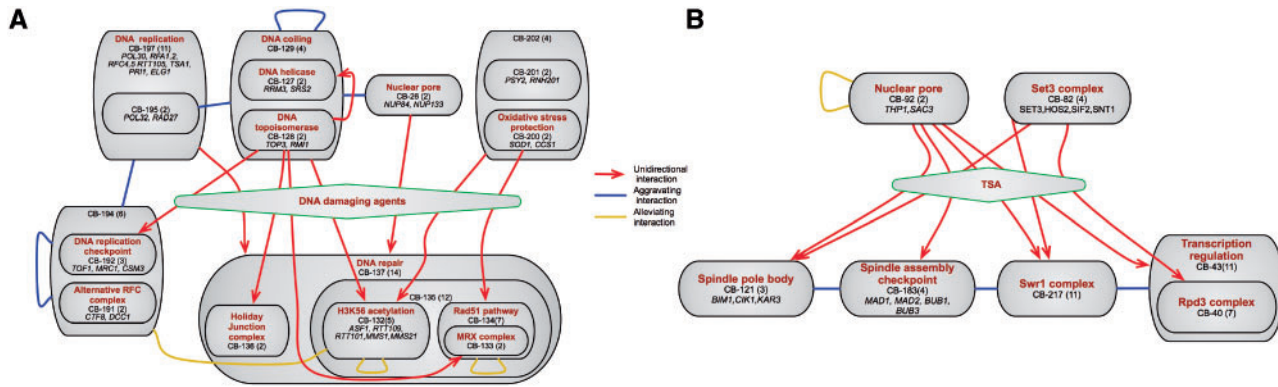


Fig. 8. Unidirectional interactions enable inference of functional hypotheses. Unidirectional edges between modules (grey boxes) are annotated by red arrows. Aggravating and alleviating interactions between modules are annotated by blue and yellow lines, respectively. (A) Unidirectional edges involving stimuli similar to hydroxyurea and camptothecin, two DNA damage-inducing drugs. Some edges were omitted from the graphical view for clarity. (B) Unidirectional edges involving the deacetylation inhibitor Trichostatin-A (TSA). Some edges were omitted from the graphical view for clarity.

these, both CB-136, that contains the Holiday junction complex, and CB-134 that comprises genes of the Rad51 pathway and MRX complex are established mechanisms of DNA damage repair. The third sub-module (CB-132) comprises five genes whose protein products were recently characterized as involved in the acetylation of histone H3 lysine 56 (H3K56Ac) pathway (Collins *et al.*, 2007a). In addition, we find an alleviating interaction between the H3K56Ac module and S-phase-related module (CB-194), suggesting that the function of H3K56Ac pathway is S-phase-related. This example illustrates the power of the combination between the hierarchical structure of modules and the annotation of unidirectional edges. Our method identifies one parent module with a general DNA repair annotation that contains three sub-modules with different interactions that imply different specific functions. For example, the alleviating interaction of CB-132 with CB-194 suggests that the H3K56Ac pathway is involved in relieving DNA damage in the S - phase. Indeed, loss of H3K56 acetylation results in higher sensitivity to exposure to DNA damaging agents during S -phase (Masumoto *et al.*, 2005) and this pathway was proposed as a DNA integrity check point following replication (Collins *et al.*, 2007a).

Another example regards the unidirectional edges related to TSA, a histone deacetylation inhibitor that affects class I and II histone deacetylases (Furumai *et al.*, 2001; Fig. 8B). We find two modules whose perturbation is phenocopied by TSA: Set3 complex (CB-82) and Thp1–Sac3 complex (CB-92). Set3 complex is a histone deacetylation complex, and thus it is plausible that TSA phenocopies its perturbation. However, the relation of the Thp1–Sac3 complex, comprising mRNA export factors associated with the nuclear pore, to deacetylation is less obvious. Clues to this puzzle can be found when examining the downstream modules with respect to this external stimulus. Most of these downstream modules are related to chromosome segregation (CB-121 and CB-183) and the Swr1 complex (CB-218), a chromatin modifier with genome integrity phenotype (van Attikum *et al.*, 2007). This suggests that TSA damages chromosome integrity, and that perturbations of Thp1–Sac3 complex and Set3 complex lead to similar damage. Indeed, previous studies showed that Thp1–Sac3 complex has a role in transcription elongation, and that its perturbation affects genome stability (González-Aguilera *et al.*, 2008). Previous works suggested

that histone deacetylation by Set3 is also associated with active transcription (Kim and Buratowski, 2009; Wang *et al.*, 2002), leading us to hypothesize that perturbations of these complexes interfere with transcription elongation, resulting in chromosome instability. Interestingly, we observe a directed interaction from Set3 to the Rpd3 complex (CB-40), also a histone deacetylase. This asymmetry is consistent with the wider range of functions of Rpd3 (Suka *et al.*, 2001) in contrast to the specificity of Set3 targets (Wang *et al.*, 2002), explaining why Rpd3 can (partially) compensate for defects in Set3 and not vice versa.

5 DISCUSSION

From maps to networks: our methodology takes a step forward towards automating the extraction of biological knowledge from large-scale genetic interaction maps. A crucial step in dealing with the large quantities of interaction data is summarizing the observations in a representation that identifies patterns in the data. Previous works mainly used local signatures to capture interactions between pairs of modules (Bandyopadhyay *et al.*, 2008; Pu *et al.*, 2008) or learn a network of disjoint modules that are coherent in terms of physical and genetic interactions (Ulitsky *et al.*, 2008). Here, we focus on finding a global representation that captures the bulk of the genetic interactions, without requiring additional information, and employ a module hierarchy to capture functional specialization of different sub-modules. Our representation facilitates inspection of the large-scale results, by presenting each module along with all its interacting partners as well as its hierarchical context. This representation defines the modules within their biological context, minimizing the requirements for expert knowledge for inference of testable biological hypotheses from genetic interaction data.

Our empirical results on two very different genetic interaction maps show that this representation captures much of the patterns of interactions in the data. Although our representation captures many interactions, it does not include all the interactions. Some of the missed interactions may be false positives, and thus at this front our analysis would serve to purge such data from the genetic interaction maps. There are, however, various reasons for missing

true interactions. For example, some interactions are excluded since we restrict the module size to at least two genes, so that noisy measurements for a specific deletion will not dominate the results. This implies that our procedure may miss a consistent set of interactions between a single gene and a module. Also, the constraint of a strict hierarchy may lead to situations where a gene with multiple functions has to choose which module to belong to and thus to miss some of its interactions (Pu *et al.*, 2008). A natural extension of our method, which can partially resolve this issue, is to allow an extended hierarchy, where a module can be the child of more than one parent. As demonstrated by the success of GO ontology in capturing functional annotations (Ashburner *et al.*, 2000), such hierarchical graphs are natural in the context of functional gene organization.

Striving for mechanisms: one goal of the analysis of genetic interaction maps is to decipher the causal explanation underlying the observed interactions. Automating this aspect of the analysis provides a significant advance toward interpretation of genetic interaction maps. Earlier studies mostly focused on interpretations that involve complexes and pathways (alleviating interactions among members of the complex/pathway, and a similar spectrum of interactions with other genes) and redundant functions of such complexes/pathways (parallel pathways may have aggravating interactions between genes involved in these pathways). Although other explanations were acknowledged (Boone *et al.*, 2007; Pan *et al.*, 2006) and implicitly used in interpreting the results, these were not reflected in automated analyses. Here, we introduce a novel automated analysis to systematically detect unidirectional interactions where a downstream gene buffers or compensates for the effect of the perturbation of an upstream gene.

Using our automated analysis, we find that a large portion of the observed aggravating genetic interactions (at least a third) can be attributed to such unidirectional interactions. This finding partially accounts for the large number of aggravating interactions between genes of distantly related functions. Moreover, the analysis annotates interactions by the type of damage caused by the perturbation of the upstream genes, providing informative clues for interpreting the results. Finally, we combine this analysis with the modular hierarchy representation to understand the relations between modules. When looking at the types of external stimuli phenocopied by gene deletions in our analysis, we find that many of them can cause major stress conditions in the cell such as DNA damage (e.g. by UV, hydroxyurea, camptothecin and MMS) and translation inhibition (e.g. cycloheximide and hygromycin B). In this case, we can interpret unidirectional compensations as connecting between a module whose perturbation causes stress and a module that has a part in relieving this stress. Indeed, many of the downstream modules associated with such stress conditions are known to be central players in the cellular response to various stress conditions, for example the DNA damage repair module (CB-137) and spindle assembly checkpoint (CB-183).

Global examination of the resulting network shows that many highly connected modules have a high percentage of their aggravating partners related through unidirectional edges related with major stress conditions (Fig. 9). Moreover, highly connected modules tend to be either upstream (i.e. their removal causes stress conditions) or downstream (i.e. stress relieving), but not both (Supplementary website). These observations suggest that

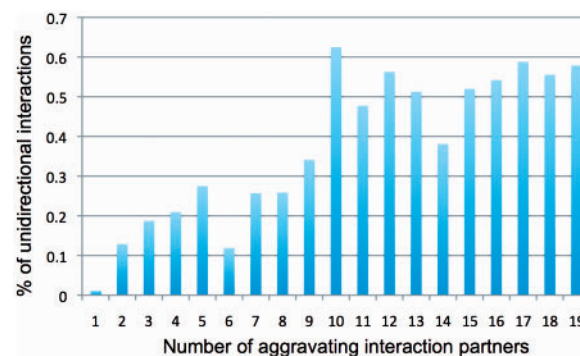


Fig. 9. Many hubs of genetic interactions are related to unidirectional compensation. A histogram of the fraction of unidirectional edges (y-axis) for modules with different degree of aggravating edges (x-axis). Each bar shows the portion of unidirectional edges out of all edges that are connected to modules with this degree.

unidirectional compensation plays a pivotal role in forming interaction hubs in genetic interaction maps. Furthermore, they suggest that responses of cellular integrity mechanisms to genetic perturbations are a major factor in shaping genetic interaction maps.

Toward organizational principles of genetic interaction maps: the methodology we present here puts forward two major contributions toward understanding the organization of genetic interaction maps. First, the hierarchy of modules is automatically built independent of additional data sources, allowing its application to various existing genetic interaction maps and also to less studied organisms. Moreover, the creation of a visual platform to study these results should boost the usability of these datasets, many of which are currently only used to find single interactions between genes of interest. Second, we elucidate some of the mechanisms underlying the interactions between modules. By integrating an additional data source we enabled the distinction between uni- and bi-directional aggravating interactions, and provided more functionally coherent interpretations to the genetic interaction maps. Our results demonstrate that searching for a causal explanation for the genetic interactions highlights specific insights into the cellular roles of genes and pathways as well as elucidates global features of the genetic interaction map. With the increasing availability of genetic interaction maps in yeast and as they become available for a large number of organisms, many of them with sparser annotation (Butland *et al.*, 2008; Byrne *et al.*, 2007; Roguev *et al.*, 2008), we believe that these methods can be generalized and will prove valuable in the automated highlighting of both the functional structure of the network as well as specific biological phenomena. This should allow us to make the first steps necessary to turn high-throughput maps into a true understanding of cellular complexity by interpreting how such maps relate to the underlying landscape of interacting cellular pathways.

ACKNOWLEDGEMENTS

We thank N. Barkai, S. Gasser, Z. Itzhaki, T. Kaplan, P.D. Kaufman, O.J. Rando, A. Regev, M. Yassour, E. Yeger-Lotem, I. Wapinski, and J.S. Weissman for discussions and useful comments on the article.

We also thank S. Collins and N. Krogan for making data available prior to publication.

Funding: Eshkol fellowship from the Israeli Ministry of Science (to A.J.); Rudin Foundation (to R.R.); Human Frontiers Science Program Career Development Award (to M.S.); European Union grant 3D-Repertoire, contract number LSHG-CT-2005-512028 (to H.M.); National Institutes of Health grant 1R01CA119176-01 (to N.F.).

Conflict of Interest: none declared.

REFERENCES

- Ashburner,M. et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Gen.*, **25**, 25–29.
- Bandyopadhyay,S. et al. (2008) Functional maps of protein complexes from quantitative genetic interaction data. *PLoS Comput. Biol.*, **4**, e1000065.
- Bennett,C.B. et al. (2001) Genes required for ionizing radiation resistance in yeast. *Nat. Gen.*, **29**, 426–434.
- Beyer,A. et al. (2007) Integrating physical and genetic maps: from genomes to interaction networks. *Nat. Rev. Genet.*, **8**, 699–710.
- Boone,C. et al. (2007) Exploring genetic interactions and networks with yeast. *Nat. Rev. Genet.*, **8**, 437–449.
- Butland,G. et al. (2008) eSGA: *E. coli* synthetic genetic array analysis. *Nat. Methods.*, **5**, 789–795.
- Byrne,A.B. et al. (2007) A global analysis of genetic interactions in *Caenorhabditis elegans*. *J. Biol.*, **6**, 8.
- Collins,S. et al. (2007a) Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, **446**, 806–810.
- Collins,S. et al. (2007b) Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol. Cell Proteomics*, **6**, 439–450.
- Cover,T. and Thomas,J. (2001) *Elements of Information Theory*. City College of New York, John Wiley, New York.
- Dudley,A. et al. (2005) A global view of pleiotropy and phenotypically derived gene function in yeast. *Mol. Syst. Biol.*, **1**, 2005.0001.
- Eisen,M. et al. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci.*, **95**, 14863–14868.
- Fiedler,D. et al. (2009) Functional organization of the *S. cerevisiae* phosphorylation network. *Cell*, **136**, 952–963.
- Furumai,R. et al. (2001) Potent histone deacetylase inhibitors built from trichostatin A and cyclic tetrapeptide antibiotics including trapoxin. *Proc. Natl Acad. Sci. USA.*, **98**, 87–92.
- Giaever,G. et al. (2002) Functional profiling of the *Saccharomyces cerevisiae* genome. *Nature*, **418**, 387–391.
- Glover,F. et al. (1993) A user's guide to TABU search. *Ann. Oper. Res.*, **41**, 1–28.
- González-Aguilera,C. et al. (2008) The THP1-SAC3-SUS1-CDC31 complex works in transcription elongation-mRNA export preventing RNA-mediated genome instability. *Mol. Biol. Cell.*, **19**, 4310–4318.
- Hanna,J. et al. (2001) *Saccharomyces cerevisiae* CTF18 and CTF4 are required for sister chromatid cohesion. *Mol. Cell Biol.*, **21**, 3144–3158.
- Helenius,A. and Aebi,M. (2004) Roles of N-linked glycans in the endoplasmic reticulum. *Annu. Rev. Biochem.*, **73**, 1019–1049.
- Hillenmeyer,M. et al. (2008) The chemical genomic portrait of yeast: uncovering a phenotype for all genes. *Science*, **320**, 362–365.
- Kelley,R. and Ideker,T. (2005). Systematic interpretations of genetic interactions using protein networks. *Nat. Biotech.*, **23**, 561–566.
- Kim,T. and Buratowski,S. (2009) Dimethylation of H3K4 by Set1 recruits the Set3 histone deacetylase complex to 5' transcribed regions. *Cell*, **137**, 259–272.
- Makhnevych,T. et al. (2009) Global map of SUMO function revealed by protein-protein interaction and genetic networks. *Mol. Cell*, **33**, 124–135.
- Masumoto,H. et al. (2005) A role for cell-cycle-regulated histone H3 lysine 56 acetylation in the DNA damage response. *Nature*, **436**, 294–298.
- Measday,V. et al. (2002) Ctf3p, the Mis6 budding yeast homolog, interacts with Mcm22p and Mm16p at the yeast outer kinetochore. *Genes Dev.*, **16**, 101–113.
- Nasmyth,K. (2005) How do so few control so many? *Cell*, **120**, 739–746.
- Pan,X. et al. (2006) A DNA integrity network in the yeast *Saccharomyces cerevisiae*. *Cell*, **124**, 1069–1081.
- Parsons,A. et al. (2004) Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways. *Nat. Biotechnol.*, **22**, 62–69.
- Parsons,A. et al. (2006) Exploring the mode-of-action of bioactive compounds by chemical-genetic profiling in yeast. *Cell*, **126**, 611–625.
- Pu,S. et al. (2008) Local coherence in genetic interaction patterns reveals prevalent functional versatility. *Bioinformatics*, **24**, 2376–2383.
- Rissanen,J. (1983) A universal prior for integers and estimation by minimum description length. *Ann. Stat.*, **11**, 416–431.
- Roguev,A. et al. (2008) Conservation and rewiring of functional modules revealed by an epistasis map in fission yeast. *Science*, **332**, 405–410.
- Schuldiner,M. et al. (2005) Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile. *Cell*, **123**, 507–519.
- Segrè,D. et al. (2005) Modular epistasis in yeast metabolism. *Nat. Genet.*, **37**, 77–83.
- Suka,N. et al. (2001) Highly specific antibodies determine histone acetylation site usage in yeast heterochromatin and euchromatin. *Mol. Cell.*, **8**, 473–479.
- Tong,A. et al. (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, **294**, 2364–2368.
- Ulitsky,I. et al. (2008) From E-MAPs to module maps: dissecting quantitative genetic interactions using physical interactions. *Mol. Syst. Biol.*, **4**, 209.
- van Attikum,H. et al. (2007) Distinct roles for SWR1 and INO80 chromatin remodeling complexes at chromosomal double-strand breaks. *EMBO J.*, **26**, 4113–4125.
- Wang,A. et al. (2002) Requirement of hos2 histone deacetylase for gene activity in yeast. *Science*, **298**, 1412–1414.
- Wilmes,G.M. et al. (2008) A genetic interaction map of RNA-processing factors reveals links between sem1/dss1-containing complexes and mRNA export and splicing. *Mol. Cell*, **32**, 735–746.

6 Discussion and conclusions

In this dissertation I presented a methodology to learn the properties of a protein-protein interaction network, while taking into account uncertainty originating from noise in large-scale experimental data. The main premise is that this should be viewed as a relational learning problem, where the large-scale experiments serve as noisy observations over hidden interaction random variables. I started by presenting the general formalization of this model, along with a proof-of-concept implementation of this model for simultaneous prediction of interactions given large-scale interaction data (Jaimovich et al., 2006). Next, to allow realistic application of this model to interaction networks covering millions of interactions, I developed tools that perform efficient approximate inference in such models (Jaimovich et al., 2007). In another work (not included in this thesis), together with Ofer Meshi, we worked on finding better approximate inference algorithms that will enable successful reconstructions of such models from noisy observations (Meshi et al., 2009). Our methodology was implemented in a general code library that facilitates implementation of similar ideas to other problems (Jaimovich et al., 2010b). Finally, I demonstrated how network analysis of genetic interaction data can elucidate biological insights from large-scale genetic interaction maps (Jaimovich et al., 2010a).

6.1 Learning relational graphical models of interaction networks

The main advantage of the application of graphical models to represent interaction networks is that it allows an elegant methodology to cope with uncertainty while dealing with various tasks, such as predicting missing links in the network and learning its properties from noisy observations. Furthermore, similar methods were successfully applied to other types of networks, such as the World Wide Web (Taskar et al., 2004) and social networks (Robins et al., 2007; Toivonen et al., 2009). However, the application of graphical models to actual biological interac-

tion networks poses many challenges. The biggest challenge is estimating the 'true' model, both in terms of feature selection and in terms of parameterization. In this setting one has to learn the set of features that define the qualitative nature of the model (that is, the set of independencies it defines) as well as the correct parameterization for these features. Della Pietra et al. (1997) proposed, what has now become a classical approach for this challenge, starting with a basic set of features and then performing a greedy search over all possible feature-sets that improve some criterion (*e.g.*, the likelihood of the data). This kind of search usually requires the calculation of the likelihood function and the expected sufficient statistics for each of the features in order to enable a gradient descent optimization of the estimated parameters for each feature-set.

During my PhD work, I have tried to develop efficient approximations that will facilitate such learning procedures. One of the reasons that makes learning such models a tough task, is that it is hard to estimate the quality of a learned model. Thus, it is hard to tell the effect the approximations have on the results of the learning procedure. One way to estimate this effect is by sampling networks from a synthetic model using Markov Chain Monte Carlo methods (Geman and Geman, 1984; Gilks et al., 1996). In turn, we can use our approximate inference procedures to learn the parameters and features from these 'synthetic' networks and compare them to those of the model we sampled from. Furthermore, by considering small networks where exact inference is feasible one can compare the same learning strategy with exact computations to estimate the effect of approximate inference on the learning results. Such comparison is not a trivial task, especially for large models, as even seemingly different feature-sets and parameterizations can describe the same distributions.

Initial results show that applying these ideas, the learning methodology described by Della Pietra et al. (1997), using exact inference to compute the likelihood and sufficient statistics, and standard statistical criteria as stopping conditions, it was possible to recover models that were relatively close to the models we sampled from. Unfortunately, when using our implementations for approximate

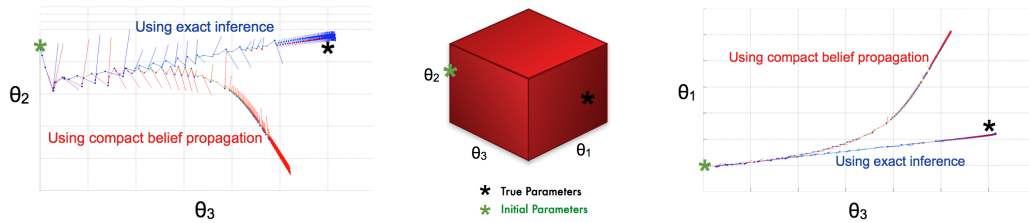


Figure 6: **Visualization of parameter estimation:** This image shows tracking of the parameter estimation for a model with three features (univariate, chain and closed triplets, denoted by $\theta_1, \theta_2, \theta_3$, respectively) from synthetic data. The black asterisk shows the parameters that were used to sample the data and the green asterisk shows the parameters used to initiate the optimization. Shown are two tracks of gradient descent iterations where the likelihood and expected sufficient statistics were estimated with exact and approximate inference (shown in blue and red, respectively). The parameter space is illustrated by a red three dimensional cube and the left and right panels show its projections on two dimensions.

inference by compact loopy belief propagation (Jaimovich et al., 2007), we observed that the error introduced by the approximation in loopy belief propagation (LBP) leads to large errors in parameter estimation (Figure 6).

One alternative that could improve the approximation quality is to use generalized belief propagation (GBP) (Yedidia et al., 2001), which was shown to provide excellent approximations for other models (*e.g.*, 2-dimensional grids (Yedidia et al., 2001)). However, the challenge here is choosing the appropriate regions and counting numbers for our model (see Section 1.2.2). In collaboration with Ofer Meshi, in a paper not included in this thesis, we developed and implemented a general version of GBP to allow for a free choice of regions and counting numbers (Meshi et al., 2009). In addition, although many works suggest specific choices of counting numbers with proven guarantees (Wainwright et al., 2005b), it is not clear how to define regions and counting numbers for a general model. Moreover, in many cases the standard LBP often performs empirically better than its generalizations. We made a systematic comparison of various choices for counting numbers on models over interaction networks, and suggested a novel method to choose counting numbers that produce better results. Unfortunately, also this

improvement did not result in successful learning procedures.

Another possible solution to this problem is to use an alternative score to evaluate the models (instead of the likelihood function), which might introduce some bias, but is easier to compute. One such alternative is *pseudo-likelihood* (Besag, 1975, 1977). Although in a study of social networks this score was suggested to be inferior to other approximate inference methods (van Duijn et al., 2009), it was successfully used to learn biological networks in a model very close to the one suggested here (Saul and Filkov, 2007). Our initial experiments on synthetic data also indicate that using pseudo-likelihood as a score function results in accurate reconstruction of the model parameters. However, inference of biological hypotheses by its application to actual interaction networks still poses additional computational challenges that we are trying to address these days.

6.2 Lifted inference in models of interaction networks

The second paper in this dissertation presented a methodology for performing inference in the template level of the model. This method allows efficient approximate calculations of both the likelihood function and the marginal distribution in a symmetric relational model over huge networks. However, one major drawback of this method is that it requires perfect symmetry in the model. In the case of the interaction network I am aiming to model this has two implications. First, we can perform inference only with either full evidence or no evidence and not with partial evidence. Second, the symmetry preservation implies that we can consider models that span all possible interactions between the relevant proteins. Furthermore, these models must assign the same parameter and counting number to each instantiation of a template feature, regardless of the actual variables it encompasses. Importantly, although these are serious limitations, this methodology still enables efficient computation of marginal probabilities and of the likelihood function for fully observed datasets over symmetric models. These calculations enable implementation of efficient procedures for parameter estimation from fully observed data. However, when dealing with partially observed

datasets, standard procedures of parameter estimation often use *expectation maximization* (EM) techniques that fill in the missing observations by computation of marginal distributions conditioned on the partially observed evidence (Dempster et al., 1977; Lauritzen, 1995). Unfortunately, our current implementation does not allow efficient approximation for this kind of tasks.

Constraining the model to only consider the entire repertoire of possible interactions between the proteins is a serious limitation. As we want to consider networks over a large number of proteins, degeneracy of the model becomes a serious concern. Specifically, for some models many parametric settings will yield either a full or an empty network (Handcock, 2003). One approach to deal with this issue, is to introduce features that capture the degree of each protein in the network (Saul and Filkov, 2007). A different, computationally oriented solution, is to allow models that break the symmetry but still use the relational nature of the model to efficiently compute approximate inference. Recently, two works built upon the work described in the second chapter of this dissertation to suggest such algorithms (Singla and Domingos, 2008; Kersting et al., 2009). Their basic approach is very similar to ours, but they allow its implementation in more general cases. Instead of requiring the entire model to be symmetric, they identify local symmetries by grouping together similar nodes in the graphical model. Thus, if the model is fully symmetric the two methods are equivalent as they will identify all local symmetry properties we assume. However, in cases the model is not fully symmetric, their methodology captures the local symmetries that allow to save computation time. Using these algorithms, we can consider a smaller set of interactions by creating datasets that will be designed to have high coverage (regardless of their false positive rate). In turn, we can use our methodology to highlight reliable interactions in this dataset and characterize them. These methods can also be used for parameter estimation based on partially-observed data. However, their efficiency remains questionable, as it is not clear whether the application of these methods to our models will result in a dramatic decrease of computation time.

6.3 *In-vivo* measurements of protein-protein interactions

One of the major concerns regarding the classical approaches for large-scale assays of protein-protein interactions is that they do not query the interactions in native conditions. For example, the affinity purification assay (Rigaut et al., 1999) usually over-expresses the TAP-tagged protein, and the yeast two-hybrid screens (Uetz et al., 2000; Ito et al., 2001) require the introduction of both prey and bait proteins into the yeast nucleus. Recently, Tarassov et al. (2008) introduced a novel method that enables *in-vivo* characterization of protein-protein interactions. The idea behind their method is similar to that of the yeast two-hybrid methodology. The difference is that instead of attaching the prey and bait proteins to two parts of a transcription factor, they are attached to two parts of a reporter protein. If the proteins interact the two parts of the reporter protein will be fused, and it will become active. By using homologous recombination, both bait and prey proteins are expressed in their endogenous genomic location under their original promoters. Furthermore, using high-throughput microscopy, this method can detect the interactions *in-vivo* in single cell resolution. This exciting methodology offers endless possibilities: testing interactions under different cellular conditions and with various perturbations, measuring cell-to-cell variability of protein-protein interactions and so on. Analysis of such experiments will highlight the need for models that take into account the dynamic nature of the interactions. The models I introduced in this dissertation offer elegant extensions for both discrete (Murphy, 2002) and continuous (Nodelman et al., 2002; El-Hay et al., 2006) dynamic models, and could be naturally extended to model the results of such assays.

6.4 Analysis of genetic interaction maps

Comprehensive genetic interaction screens for a variety of organisms, ranging from simple bacteria to multicellular eukaryotes, are becoming more and more popular (Byrne et al., 2007; Typas et al., 2008; Wilmes et al., 2008; Roguev et al., 2008; Butland et al., 2008; Breslow et al., 2008; Fiedler et al., 2009; Costanzo

et al., 2010). Analysis of these data, mainly using manual intervention and expert knowledge, has already led to many biological insights. Yet, fully automated analysis of large-scale genetic interaction screens poses a real challenge to the computational biology community. In the fourth chapter of this dissertation I presented a novel method that makes a step forward in this direction. Several works tried to identify functional modules of genes from such screens (Ulitsky et al., 2008; Pu et al., 2008; Bandyopadhyay et al., 2008). In this work we created a hierarchical organization of these modules in one coherent structure and showed how such organization makes it easier to deduce biological insights from the results of genetic interaction screens. However, this hierarchical organization comes at the price of lower coverage of the interactions since we can only cover interactions that are consistent with our hierarchical organization. One possible extension of this work would be to enable larger coverage while preserving some organization of the results in a coherent structure. This can be obtained using a more sophisticated hierarchical structure that allows a child module to have more than one parent module. By limiting the number of parents one can ensure a reasonably coherent structure. However, searching the space of such structures might be much more complicated. Another interesting addition to this model is to consider single-gene interactions. One of the reasons we join genes into modules is to overcome noise in a single observed genetic interaction between two genes. However, once the modules are created, we can look for patterns of interactions between a single gene with all the genes in one module, and hope that these will be coherent, since the genes in the module should have the same function. Thus, we can systematically look for single genes that have a well defined pattern of interactions with some module, leading to more focused insights into the function of specific genes.

Another relatively delicate point in our work is that we are building our modules based directly on the actual values of the genetic interactions between the genes. Although this results in coherent relations between modules, it is relatively prone to noise. We hope to overcome this noise by looking for groups of inter-

actions with the same sign and strength. We also indirectly take into account the correlation between genetic interaction profiles of two genes by using the hierarchical clustering as the starting point for our greedy optimization procedure. One possible extension of this work is to consider the correlation between two genes also in the greedy improvement steps. Previous works have shown that this correlation is a more robust measure for functional relation (Schuldiner et al., 2005; Ulitsky et al., 2008) and can help in coping with noise in the data. In addition, it can help in discriminating between proteins that share the same roles in a complex and proteins that take part in the same complex but play different roles. This also shows that considering various observations on the phenotype of each single perturbation (in this case, its profile of genetic interactions) can strengthen the computational analysis. In fact, a critical part of our work is based on using additional data on the phenotype of each single deletion, its sensitivity to chemical perturbations, to shed additional light on the observed genetic interactions.

Notably, the most obvious phenotype of the single perturbations, its growth rate, was not measured directly in the assays we used in our analysis (Schuldiner et al., 2005; Collins et al., 2007b). Instead, by assuming that most gene-pairs do not have a significant functional relation they estimated the effect of the single perturbations based on the double perturbations of each gene with all other genes. The strength of this method is that it is relatively not sensitive to noise that may occur when directly measuring the single perturbation phenotype. However, more recent methods did measure directly the phenotype of single perturbations (Breslow et al., 2008; Jonikas et al., 2009; Costanzo et al., 2010) and showed how such measurements can help in the analysis of genetic interactions. Furthermore, a recent work used the results of these assays to systematically identify functional pathways as well as the directionality of interactions within them (Battle et al., 2010). Their main idea is that if a protein x is 'upstream' to a protein y in some pathway then the phenotype of perturbing both x and y will be similar to that of a single perturbation of x (and not to that of perturbing y). Furthermore, in case of a longer pathways this same logic produces more complex predictions that can

also add confidence to observed interactions with low scores. Namely, if we are confident that x is upstream to y and that y is upstream to z then we can increase the confidence of x being upstream of z .

Examining the results of Schuldiner et al. (2005) and Collins et al. (2007b), it seems that using the growth rate phenotype to identify genetic interactions creates a bias towards processes that are involved in cell division and growth. Although other processes are also identified in the results, the most clear and coherent signals belong to pathways and complexes related to cell division, DNA duplication, and other related cellular functions. This illustrates that the measured phenotype will have a strong effect on the type of functional relations captured by this type of large-scale assays. Advances in technology lead to exciting developments in this field by allowing genetic screens with much more precise phenotypes. For example, to analyze the activity of the unfolded protein response (UPR) Jonikas et al. (2009) used a green fluorescent protein (GFP) attached to synthetic promoter that contains multiple binding sites of a major UPR regulator (Hac1). The activity of the fluorescent protein can be measured in single cell resolution using flow cytometry. By creating yeast strains that carry this promoter, along with various single and double perturbations of yeast genes, they provided a large-scale assay that characterizes genes required for protein folding in the endoplasmic reticulum. Vizeacoumar et al. (2010) introduced a GFP fused tubulin protein to the arrayed collection of deletion mutants, and later to a set of double deletion strains. By using high throughput microscopy, they were able to explore the yeast spindle morphogenesis in great detail. In these works, the measured phenotype is aimed to enable analysis of a specific cellular mechanism. Obviously, this leads to some information loss on genes whose functions are related to other mechanisms, but on the other hand the characterized relations between genes readily provide actual biological insights. Furthermore, these 'next generation' genetic screens produce single-cell resolution data on the system they query. Although current analyses use only means (or medians) of these single-cell data (Jonikas et al., 2009; Vizeacoumar et al., 2010; Battle et al., 2010), it is interesting to examine more detailed

information in other features of the distribution of the reporter activity within the population of cells. In fact, in an ongoing research [Rinott, Jaimovich and Friedman ; submitted] we show that by looking at the cell-to-cell variability in the data of Jonikas et al. (2009) one can learn about the global properties of transcription regulation as well as on the regulation of the unfolded protein response.

6.5 Implications of our methodology for analysis of networks

The relevance of network analysis has always been a matter of controversy. The seminal work of Milo et al. (2002) has shown that by learning some local properties of the network (in their case, the over-represented network connected patterns, termed motifs), one can actually learn about biological principles that are implemented in many cellular processes. An interesting question that remains open is to what extent the global properties of the graph are derived by this series of local features. In this sense my methodology suggests a substantial advance in our understanding of the network structure. First, when learning the set of features of such a model and considering the addition of a new feature, the increase in the model's fit is tested given the current set of features that describe it. This means that a feature will be added to the model only if it improves the model's fit to the observed network. Importantly, this strategy requires some statistical tools that make sure we do not over-fit the data. Using advanced tools for learning such models, we can also allow removal of existing features from the model once they become unnecessary. For example, if the existence of a large feature in the network gives rise to an overabundance of smaller motifs, we expect the model to identify the fact that once this large feature is added, the smaller features are not needed any more. Unlike the work of Milo et al. (2002) who considered each feature independently, this process should result in a non-redundant set of features. One example that illustrates the significance of this advantage can be seen in our analysis of genetic interactions. Originally, our research has started by looking for enriched sets of network motifs in the network using the algorithms introduced by Milo et al. (2002) and Yeager-Lotem et al. (2004). This search re-

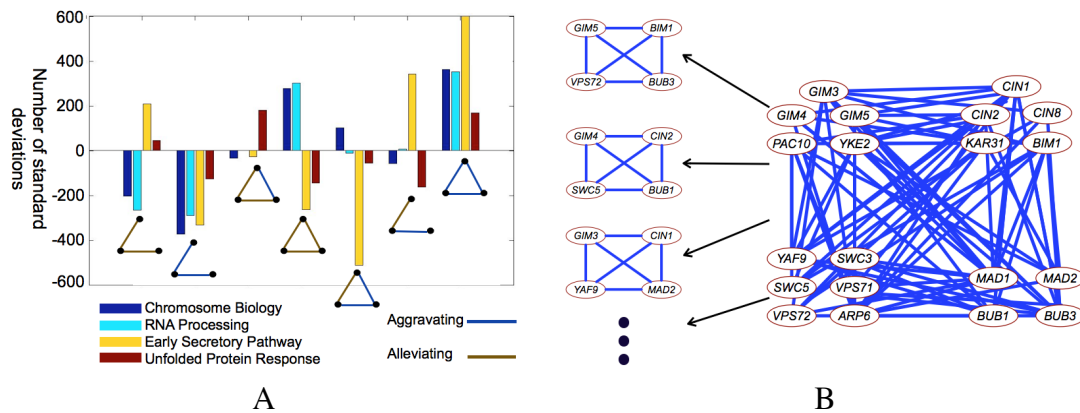


Figure 7: **Small and large features in the genetic interaction networks** Left panel shows network motifs analysis (Milo et al., 2002) using the algorithms of Yeager-Lotem et al. (2004) applied to four large-scale genetic interaction assays. The bars are colored according to the corresponding assays: Collins et al. (2007b) in blue bars, Wilmes et al. (2008) in light blue bars, Schuldiner et al. (2005) in yellow bars and Jonikas et al. (2009) in red bars. The x-axis shows the motifs (alleviating interactions in yellow and aggravating in blue) and the y-axis shows the enrichment of this motif in terms of number of standard deviations compared with random networks. Right panel shows how interactions between a small number of functional modules can create numerous small motifs.

sulted in many small motifs that seemed significantly over represented (Figure 7a). However, manual examination of these results by grouping known pathways discovered that interactions between functional units are in fact responsible for this over-representation (Figure 7b). Thus, in this case the interesting phenomenon is defined by the larger motifs. Following this conclusion, we turned to search for more complex structures in the network.

The more interesting question, however, is to what extent the global properties of the network can be explained as a direct result of its local features. The generative nature of our models allows one direction to approach this question by sampling networks from the learnt model and comparing their global features to the interaction network. This could take us one step closer to understanding the nature of biological interaction networks.

6.6 Integration of interaction networks with other data sources

The results of Jaimovich et al. (2010a) show that integrating additional layers of information with models of interaction networks leads to deciphering actual biological mechanisms. In our work, we consider the additional information only after detection of the modules, to shed light on the mechanisms that cause their interactions. Integrating information from additional data sources on single protein attributes as well as on relations between proteins within the process of identification of functional modules can take us another step towards deduction of biological insights from such observations. One possible algorithm that might yield such an automatic division can be based on the work of Roy et al. (2007). Their basic idea is to use data on single node attributes and on relations between nodes to learn an annotated hierarchy that gives the best division of the nodes into categories. This algorithm presents several advantages. First, it finds a hierarchical organization that will optimize the categorization in all relations and attributes simultaneously. Second, each such organization is scored by integrating over all possible divisions into categories that are consistent with this hierarchical structure (in contrast to considering only the best division). Finally, Roy et al. (2007) also devised a dynamic programming algorithm that enables exact calculation of the score of each organization efficiently. A drawback of this method is that its implementation for a search in the space of possible organizations is infeasible for networks with hundreds of nodes, such as the genetic interaction networks. As this space is very complex, devising an algorithm that will yield the best possible search strategy remains a challenge.

6.7 Open source software

One of the main problems of the computational scientific community is the ability to reproduce and build upon existing algorithms towards extending and improving current methodologies. Thus, when presenting novel computational tools that suggest either algorithmic improvements or a novel methodology for analysis of

data, it is of crucial importance to present open source software that implements these ideas. Although this statement seems obvious, it is not trivial to produce a software package that will truly offer both an efficient implementation along with a documented class hierarchy that will allow easy extension of the implemented algorithms. Most of the methods I developed during my PhD work were implemented in an open source library, which also contains implementation of many other existing algorithms. Together with Ofer Meshi and Gal Eldian we have made a special effort to document its base classes and to offer user-guides that will ease algorithmic and implementation improvements of our algorithms. Since it was made officially publicly available, our software package was downloaded from the Machine Learning Open Source Software (MLOSS) website by more than 300 users. Even before its publication, we shared it upon request and it was used in a number of applications both in our labs and by our collaborators. Starting with improvements of approximate inference techniques (Elidan et al., 06; Meshi et al., 2009), followed by many applications in protein design (Fromer and Yanover, 2008, 2009; Fromer and Shifman, 2009; Fromer et al., 2010), localization of objects in cluttered images (Elidan et al., 2006; Heitz et al., 2009) and Cryo Electron Tomography image alignment (Amat et al., 2008). I hope that many other works will be able to use this infrastructure to create new tools that both improve current methodologies and implements existing ideas in other scientific problems.

6.8 Concluding remarks

In my PhD work I strived to advance the field of biological network analysis on two related fronts. The first front regards methodology that will provide a rich modeling language of interaction networks. At the second front I aimed to show how such modeling can actually lead to insights into biological mechanisms. In recent years technological advances facilitate production of large-scale assays measuring both physical and genetic interactions. I believe that the models and algorithms presented in this dissertation will be a useful in the efforts to analyze

this kind of results.

References

- U Alon. Biological networks: the tinkerer as an engineer. *Science*, 301: 1866–7, 2003.
- F Amat, F Moussavi, L R Comolli, G Elidan, K H Downing, and M Horowitz. Markov random field based automatic image alignment for electron tomography. *J Struct Biol*, 161: 260–75, 2008.
- Y Artzy-Randrup, S J Fleishman, N Ben-Tal, and L Stone. Comment on ”network motifs: simple building blocks of complex networks” and ”superfamilies of evolved and designed networks”. *Science*, 305: 1107, 2004.
- S Bandyopadhyay, R Kelley, N Krogan, and T Ideker. Functional maps of protein complexes from quantitative genetic interaction data. *PLoS Comput Biol*, 4: e1000065, 2008.
- A L. Barabasi and R Albert. Emergence of scaling in random networks. *Science*, 286: 509–512, 1999.
- A L. Barabasi and Z N. Oltvai. Network biology: understanding the cell’s functional organization. *Nat. Rev. Genet.*, 5: 101–113, 2004.
- N N Batada, T Reguly, A Breitkreutz, L Boucher, B Breitkreutz, L D Hurst, and M Tyers. Stratus not altocumulus: a new view of the yeast protein interaction network. *PLoS Biol*, 4: e317, 2006.
- N N Batada, T Reguly, A Breitkreutz, L Boucher, B Breitkreutz, L D Hurst, and M Tyers. Still stratus not altocumulus: further evidence against the date/party hub distinction. *PLoS Biol*, 5: e154, 2007.
- A Battle, M C Jonikas, P Walter, J S Weissman, and D Koller. Automated identification of pathways from quantitative genetic interaction data. *Mol Syst Biol*, 6: 379, 2010.

- N Bertin, N Simonis, D Dupuy, M E Cusick, J J Han, H B Fraser, F P Roth, and M Vidal. Confirmation of organized modularity in the yeast interactome. *PLoS Biol*, 5: e153, 2007.
- J Besag. Statistical analysis of non-lattice data. *The statistician*, 1975.
- J Besag. Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, 1977.
- J Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36: 192–236, 1974.
- J R. Bock and D A. Gough. Whole-proteome interaction mining. *Bioinformatics*, 19: 125–134, 2003.
- C Boone, H Bussey, and B Andrews. Exploring genetic interactions and networks with yeast. *Nat Rev Genet*, 8: 437–49, 2007.
- D Breslow, D Cameron, S Collins, M Schuldiner, J Stewart-Ornstein, H Newman, S Braun, H Madhani, N Krogan, and J Weissman. A comprehensive strategy enabling high-resolution functional analysis of the yeast genome. *Nat Methods*, 2008.
- W Buntine. Chain graphs for learning. In *UAI '95*, 46–54, 1995.
- G Butland, M Babu, J Díaz-Mejía, F Bohdana, S Phanse, B Gold, W Yang, J Li, A Gagarinova, O Pogoutse, H Mori, B Wanner, H Lo, J Wasniewski, C Christopoulos, M Ali, P Venn, A Safavi-Naini, N Sourour, S Caron, J Choi, L Laigle, A Nazarians-Armavil, A Deshpande, S Joe, K Datsenko, N Yamamoto, B Andrews, C Boone, H Ding, B Sheikh, G Moreno-Hagelseib, J Greenblatt, and A Emili. esga: E. coli synthetic genetic array analysis. *Nat Methods*, 5: 789–795, 2008.
- A Byrne, M Weirauch, V Wong, M Koeva, S Dixon, J Stuart, and P Roy. A global analysis of genetic interactions in caenorhabditis elegans. *J Biol*, 6: 8, 2007.

- M Chavira, A Darwiche, and M Jaeger. Compiling relational Bayesian networks for exact inference. *International Journal of Approximate Reasoning*, 42: 4–20, 2006.
- R Cohen and S Havlin. Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90: 058701, 2003.
- S Collins, P Kemmeren, X Zhao, J Greenblatt, F Spencer, F Holstege, J Weissman, and N Krogan. Toward a comprehensive atlas of the physical interactome of *saccharomyces cerevisiae*. *Mol Cell Proteomics*, 6: 439–50, 2007a.
- S Collins, K Miller, N Maas, A Roguev, J Fillingham, C Chu, M Schuldiner, M Gebbia, J Recht, M Shales, H Ding, H Xu, J Han, K Ingvarsdottir, B Cheng, B Andrews, C Boone, S Berger, P Hieter, Z Zhang, G Brown, C Ingles, A Emili, C Allis, D Toczyski, J Weissman, J Greenblatt, and N Krogan. Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, 446: 806–10, 2007b.
- S R. Collins, M Schuldiner, N J. Krogan, and J S. Weissman. A strategy for extracting and analyzing large-scale quantitative epistatic interaction data. *Genome Biol*, 7: R63, 2006.
- M Costanzo, A Baryshnikova, J Bellay, Y Kim, E D. Spear, C S. Sevier, H Ding, J L. Koh, K Toufighi, S Mostafavi, J Prinz, R P. St Onge, B VanderSluis, T Makhnevych, F J. Vizeacoumar, S Alizadeh, S Bahr, R L. Brost, Y Chen, M Cokol, R Deshpande, Z Li, Z Y. Lin, W Liang, M Marback, J Paw, B J. San Luis, E Shuteriqi, A H. Tong, N van Dyk, I M. Wallace, J A. Whitney, M T. Weirauch, G Zhong, H Zhu, W A. Houry, M Brudno, S Ragibizadeh, B Papp, C Pal, F P. Roth, G Giaever, C Nislow, O G. Troyanskaya, H Bussey, G D. Bader, A C. Gingras, Q D. Morris, P M. Kim, C A. Kaiser, C L. Myers, B J. Andrews, and C Boone. The genetic landscape of a cell. *Science*, 327: 425–431, 2010.

- S Della Pietra, V Della Pietra, and J Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19: 380–393, 1997.
- A P. Dempster, N M. Laird, and D B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39: 1–39, 1977.
- M Deng, Z Tu, F Sun, and T Chen. Mapping gene ontology to proteins based on protein-protein interaction data. *Bioinformatics*, 20: 895–902, 2004.
- T El-Hay, N Friedman, and D Koller. Continuous time markov networks. *Proceedings of the Twentyfifth Confernece on Uncertainty in Artificial Intellignce (UAI 06)*, 2006.
- G Elidan, G Heitz, and D Koller. Learning object shape: From drawings to images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- G Elidan, I McGraw, and D Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proc Twenty Second Conference on Uncertainty in Artificial Intelligence (UAI '06)*, 2006.
- P Erdos and A Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.
- D Fiedler, H Braberg, M Mehta, G Chechik, G Cagney, P Mukherjee, A C Silva, M Shales, S R Collins, S van Wageningen, P Kemmeren, F P Holstege, J S Weissman, M-C Keogh, D Koller, K M Shokat, and N J Krogan. Functional organization of the s. cerevisiae phosphorylation network. *Cell*, 136: 952–63, 2009.
- O Frank and D Strauss. Markov graphs. *Journal of American Statistics Association*, 81, 1986.

- N Friedman, L Getoor, D Koller, and A Pfeffer. Learning probabilistic relational models. In *IJCAI '99*, 1300–1309. 1999.
- N Friedman, M Linial, I Nachaman and Danan Pe'er. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7:601-20. 2000.
- M Fromer and J M. Shifman. Tradeoff between stability and multispecificity in the design of promiscuous proteins. *PLoS Computational Biology*, 5:e1000627, 2009.
- M Fromer and C Yanover. A computational framework to empower probabilistic protein design. *Bioinformatics*, 24: i214–222, 2008.
- M Fromer and C Yanover. Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. *Proteins: Structure, Function, and Bioinformatics*, 75:682–705, 2009.
- M Fromer, C Yanover, and M Linial. Design of multispecific protein sequences using probabilistic graphical modeling. *Proteins: Structure, Function, and Bioinformatics*, 78:530–547, 2010.
- M Y. Galperin and E V. Koonin. Who's your neighbor? New computational approaches for functional genomics. *Nat. Biotechnol.*, 18:609–613, 2000.
- A C Gavin, P Aloy, P Grandi, R Krause, M Boesche, M Marzioch, C Rau, L J Jensen, S Bastuck, B Dumpelfeld, A Edelmann, M A Heurtier, V Hoffman, C Hoefert, K Klein, M Hudak, A M Michon, M Schelder, M Schirle, M Remor, T Rudi, S Hooper, A Bauer, T Bouwmeester, G Casari, G Drewes, G Neubauer, J M Rick, B Kuster, P Bork, R B Russell, and G Superti-Furga. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440: 631–636, 2006.
- A C Gavin, M Boschbe, R Krause, P Grandi, M Marzioch, A Bauer, J Schultz, J M Rick, A M Michon, C M Cruciat, M Remor, C Hofert, M Schelder, M Brajenovic, H Ruffner, A Merino, K Klein, M Hudak, D Dickson, T Rudi, V Gnau,

- A Bauch, S Bastuck, B Huhse, C Leutwein, M A Heurtier, R R Copley, A Edelmann, E Querfurth, V Rybin, G Drewes, M Raida, T Bouwmeester, P Bork, B Seraphin, B Kuster, G Neubauer, and G Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415: 141–147, 2002.
- S Geman and D Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 721–741, 1984.
- L Getoor, N Friedman, D Koller, and B Taskar. Learning probabilistic models of relational structure. In *Eighteenth International Conference on Machine Learning (ICML)*. 2001.
- S Ghaemmaghami, WK Huh, K Bower, R W Howson, A Belle, N Dephoure, E K O’Shea, and J S Weissman. Global analysis of protein expression in yeast. *Nature*, 425:737 – 741, 2003.
- W R Gilks, S Richardson, and D J Spiegelhalter. *Markov Chain Monte Carlo Methods in Practice*. CRC Press, 1996.
- J D Han, N Bertin, T Hao, D S Goldberg, G F Berriz, L V Zhang, D Dupuy, A J Walhout, M E Cusick, F P Roth, and M Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430: 88–93, 2004.
- M Handcock. Assessing degeneracy in statistical models of social networks. *Cite-seer*, 2003.
- M Handcock and K Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 2010.
- D E Heckerman and B N Nathwani. Toward normative expert systems. II. Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, 31:106–16, 1992.

- G Heitz, G Elidan, B Packer, and D Koller. Shape-based object localization for descriptive classification. *International journal of computer vision*, 2009.
- Y Ho, A Gruhler, A Heilbut, G D. Bader, L Moore, S L. Adams, A Millar, P Taylor, K Bennett, K Boutilier, L Yang, C Wolting, I Donaldson, S Schandorff, J Shewnarane, M Vo, J Taggart, M Goudreault, B Muskat, C Alfarano, D Dewar, Z Lin, K Michalickova, A R. Willems, H Sassi, P A. Nielsen, K J. Rasmussen, J R. Andersen, L E. Johansen, L H. Hansen, H Jespersen, A Podtelejnikov, E Nielsen, J Crawford, V Poulsen, B D. Sørensen, J Matthiesen, R C. Hendrickson, F Gleeson, T Pawson, M F. Moran, D Durocher, M Mann, C W. Hogue, D Figeys, and M Tyers. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415:180–183, 2002.
- 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: 4569–4574, 2001.
- A Jaimovich, O Meshi, and N Friedman. Template based inference in symmetric relational markov random fields. *Proc. of the Twentysixth Conference on Artificial Intelligence (UAI 07)*, 2007.
- A Jaimovich, O Meshi, I McGraw, and G Elidan. Fastinf: An efficient approximate inference library. *Journal of Machine Learning*, 2010a.
- A Jaimovich, G Elidan, H Margalit, and N Friedman. Towards an integrated protein-protein interaction network: a relational markov network approach. *J Comput Biol*, 13: 145–64, 2006.
- A Jaimovich, R Rinott, M Schuldiner, H Margalit, and N Friedman. Modularity and directionality in genetic interaction maps. *Bioinformatics*, 26: i228–36, 2010b.
- R Jansen, H Yu, D Greenbaum, Y Kluger, N J. Krogan, S Chung, A Emili, M Snyder, J F. Greenblatt, and M Gerstein. A Bayesian networks approach for pre-

- dicting protein-protein interactions from genomic data. *Science*, 302: 449–453, 2003.
- L J Jensen, M Kuhn, M Stark, S Chaffron, C Creevey, J Muller, T Doerks, P Julien, A Roth, M Simonovic, P Bork, and C von Mering. STRING 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.*, 37:D412–416, 2009.
- T Jensen, M Neville, and J Rain. Identification of novel *saccharomyces cerevisiae* proteins with nuclear export activity: cell cycle-regulated transcription factor *ace2p* shows cell cycle-independent nucleocytoplasmic shuttling. *Molecular and Cellular Biology*, 20: 8047–58, 2000.
- H Jeong, S P Mason, A L Barabási, and Z N Oltvai. Lethality and centrality in protein networks. *Nature*, 411: 41–2, 2001.
- Martin C Jonikas, Sean R Collins, Vladimir Denic, Eugene Oh, Erin M Quan, V Schmid, J Weibezahn, B Schwappach, P Walter, J S Weissman, and M Schuldiner. Comprehensive characterization of genes required for protein folding in the endoplasmic reticulum. *Science*, 323: 1693–7, 2009.
- S Kaplan, A Bren, E Dekel, and U Alon. The incoherent feed-forward loop can generate non-monotonic input functions for genes. *Mol. Syst. Biol.*, 4:203, 2008.
- K Kersting, B Ahmadi, and S Natarajan. Counting belief propagation. *Proceedings of the 25th conference in Uncertainty in Artificial Intelligence (UAI)*, 2009.
- N J Krogan, G Cagney, H Yu, G Zhong, X Guo, A Ignatchenko, J Li, S Pu, N Datta, A P Tikuisis, T Punna, J M Peregrin-Alvarez, M Shales, X Zhang, M Davey, M D Robinson, A Paccanaro, J E Bray, A Sheung, B Beattie, D P Richards, V Canadien, A Lalev, F Mena, P Wong, A Starostine, M M Canete, J Vlasblom, S Wu, C Orsi, S R Collins, S Chandran, R Haw, J J Rilstone, K Gandi, N J Thompson, G Musso, P St Onge, S Ghanny, M H Lam, G Butland,

- A M Altaf-Ul, S Kanaya, A Shilatifard, E O'Shea, J S Weissman, C J Ingles, T R Hughes, J Parkinson, M Gerstein, S J Wodak, A Emili, and J F Greenblatt. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, 440: 637–643, 2006.
- F R Kschischang, B J Frey, and H A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), 2001.
- J Lafferty, A Mccallum, and F Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning (ICML 01)*, 2001.
- S L Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.
- G Lima-Mendez and J van Helden. The powerful law of the power law and other myths in network biology. *Mol Biosyst*, 5: 1482–93, 2009.
- S Mangan, S Itzkovitz, A Zaslaver, and U Alon. The incoherent feed-forward loop accelerates the response-time of the gal system of *Escherichia coli*. *J Mol Biol*, 356: 1073–81, 2006.
- S Mangan, A Zaslaver, and U Alon. The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *J Mol Biol*, 334: 197–204, 2003.
- E M Marcotte, M Pellegrini, H L Ng, D W Rice, T O Yeates, and D Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285: 751–753, 1999a.
- E M Marcotte, M Pellegrini, M J Thompson, T O Yeates, and D Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402: 83–86, 1999b.

- R McEliece, D McKay, and J Cheng. Turbo decoding as an instance of pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communication*, 16:140–152, 1998.
- O Meshi, A Jaimovich, and A Globerson. Convexifying the bethe free energy. *Proc Twentyfifth Conference on Uncertainty in Artificial Intelligence (UAI 09)*, 2009.
- HW Mewes, J Hani, F Pfeiffer, and D Frishman. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, 26:33–37, 1998.
- R Milo, S Itzkovitz, N Kashtan, R Levitt, S Shen-Orr, I Ayzenshtat, M Sheffer, and U Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.
- R Milo, S Shen-Orr, S Itzkovitz, N Kashtan, D Chklovskii, and U Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.
- K Murphy and Y Weiss. Loopy belief propagation for approximate inference: An empirical study. In *Proc Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, 1999.
- K P Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, Berkeley, California USA, 2002.
- U Nodelman, C Shelton, and D Koller. Continuous time bayesian networks. *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI 02)*, 2002.
- J Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, California, 1988.

- M Pellegrini, E Marcotte, M Thompson, D Eisenberg, and T Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A*, 96: 4285–4288, 1999.
- S Pu, K Ronen, J Vlasblom, J Greenblatt, and S J Wodak. Local coherence in genetic interaction patterns reveals prevalent functional versatility. *Bioinformatics*, 24: 2376–83, 2008.
- G Rigaut, A Shevchenko, B Rutz, M Wilm, M Mann, and B Seraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nat Biotechnol*, 17: 1030–1032, 1999.
- G Robins, P Pattison, Y Kalish, and D Lusher. An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 2007.
- A Rodal, J Tetreault, and P Lappalainen. Aip1p interacts with cofilin to disassemble actin filaments. *The Journal of cell biology*, 1999.
- A Roguev, S Bandyopadhyay, M Zofall, K Zhang, T Fischer, S Collins, H Qu, M Shales, H Park, J Hayles, K Hoe, D Kim, T Ideker, S Grewal, J Weissman, and N Krogan. Conservation and rewiring of functional modules revealed by an epistasis map in fission yeast. *Science*, 17: 205–410, 2008.
- D Roy, C Kemp, and V Mansinghka. Learning annotated hierarchies from relational data. *Advances in Neural Information Processing Systems*, 2007.
- Zachary M Saul and Vladimir Filkov. Exploring biological network structure using exponential random graph models. *Bioinformatics*, 23: 2604–11, 2007.
- M Schuldiner, S R. Collins, N J. Thompson, V Denic, A Bhamidipati, T Punna, J Ihmels, B Andrews, C Boone, J F. Greenblatt, J S. Weissman, and N J. Krogan. Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile. *Cell*, 123: 507–519, 2005.

- E Segal, N Friedman, D Koller, and A Regev. A module map showing conditional activity of expression modules in cancer. *Nat Genet*, 36: 1090–1098, 2004.
- S S Shen-Orr, R Milo, S Mangan, and U Alon. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, 31:64–68, 2002.
- N Shental, A Zommet, T Hertz, and Y Weiss. Learning and inferring image segmentations with the gbp typical cut algorithm. In *ICCV*, 2003.
- J Shotton, J Winn, C Rother, and A Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision ECCV 2006*, 2006.
- P Singla and P Domingos. Lifted first-order belief propagation. *Association for the Advancement of Artificial Intelligence*, 2008.
- E Sprinzak and H Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *J Mol Biol*, 311: 681–692, 2001.
- E Sprinzak, S Sattath, and H Margalit. How reliable are experimental protein-protein interaction data? *J Mol Biol*, 327: 919–923, 2003.
- K Tarassov, V Messier, C R Landry, S Radinovic, M S Molina, I Shames, Y Malitskaya, J Vogel, H Bussey, and S W Michnick. An in vivo map of the yeast protein interactome. *Science*, 320: 1465–70, 2008.
- B Taskar, A Pieter Abbeel, and D Koller. Discriminative probabilistic models for relational data. In *Proc Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI '02)*, 2002.
- B Taskar, M F. Wong, P Abbeel, and D Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems 16*, 2004.

- R Toivonen, L Kovanen, M Kivelä, and J Onnela. A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, 2009.
- A H Tong, G Lesage, G D Bader, H Ding, H Xu, X Xin, J Young, G F Berriz, R L Brost, M Chang, Y Chen, X Cheng, G Chua, H Friesen, D S Goldberg, J Haynes, C Humphries, G He, S Hussein, L Ke, N Krogan, Z Li, J N Levinson, H Lu, P Menard, C Munyana, A B Parsons, O Ryan, R Tonikian, T Roberts, A M Sdicu, J Shapiro, B Sheikh, B Suter, S L Wong, L V Zhang, H Zhu, C G Burd, S Munro, C Sander, J Rine, J Greenblatt, M Peter, A Bretscher, G Bell, F P Roth, G W Brown, B Andrews, H Bussey, and C Boone. Global mapping of the yeast genetic interaction network. *Science*, 303: 808–813, 2004.
- A Typas, R Nichols, D Siegele, M Shales, S Collins, B Lim, H Braberg, N Yamamoto, R Takeuchi, B Wanner, H Mori, J Weissman, N Krogan, and C Gross. High-throughput, quantitative analyses of genetic interactions in *e. coli*. *Nat Methods*, 5: 781–787, 2008.
- 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: 623–627, 2000.
- I Ulitsky, T Shlomi, M Kupiec, and R Shamir. From e-maps to module maps: dissecting quantitative genetic interactions using physical interactions. *Mol Syst Biol*, 4:209, 2008.
- M van Duijn, K Gile, and M Handcock. A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, 2009.
- F J Vizeacoumar, N van Dyk, F S Vizeacoumar, V Cheung, J Li, Y Sydorskyy, N Case, Z Li, A Datti, C Nislow, B Raught, Z Zhang, B Frey, K Bloom, C Boone,

- and B J Andrews. Integrating high-throughput genetic interaction mapping and high-content screening to explore yeast spindle morphogenesis. *J Cell Biol*, 188: 69–81, 2010.
- C von Mering, R Krause, B Snel, M Cornell, S G. Oliver, S Fields, and P Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417: 399–403, 2002.
- M J. Wainwright, T Jaakkola, and A S. Willsky. Exact map estimates by (hyper)tree agreement. In *Advances in Neural Information Processing Systems 15* 2002.
- M J Wainwright, T S Jaakkola, and A S Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51: 2313–2335, 2005.
- M J Wainwright and M I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1: 1–305, 2008.
- W Wiegerinck and T Heskes. Fractional belief propagation. In *Advances in Neural Information Processing Systems 15* 2003.
- G M Wilmes, M Bergkessel, S Bandyopadhyay, M Shales, H Braberg, G Cagney, S R Collins, G B Whitworth, T L Kress, J S Weissman, T Ideker, C Guthrie, and N J Krogan. A genetic interaction map of rna-processing factors reveals links between sem1/dss1-containing complexes and mrna export and splicing. *Mol Cell*, 32: 735–46, 2008.
- I Xenarios, D W Rice, L Salwinski, M K Baron, E M Marcotte, and D Eisenberg. DIP: the database of interacting proteins. *Nucleic Acids Res*, 28: 289–291, 2000.

- T Yamada and P Bork. Evolution of biomolecular networks: lessons from metabolic and protein interactions. *Nature Reviews Molecular Cell Biology*, 10: 791–803, 2009.
- J S Yedidia, W T Freeman, and Y Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, 2001.
- J S Yedidia, W T Freeman, and Y Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51: 2282–2312, 2005.
- E Yeger-Lotem, S Sattath, N Kashtan, S Itzkovitz, R Milo, R Y. Pinter, U Alon, and H Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc Natl Acad Sci U S A*, 101: 5934–5939, 2004.
- H Yu, P Braun, M A. Yildirim, I Lemmens, K Venkatesan, J Sahalie, T Hirozane-Kishikawa, F Gebreab, N Li, N Simonis, T Hao, J F. Rual, A Dricot, A Vazquez, R R. Murray, C Simon, L Tardivo, S Tam, N Svzrikapa, C Fan, A S. de Smet, A Motyl, M E. Hudson, J Park, X Xin, M E. Cusick, T Moore, C Boone, M Snyder, F P. Roth, A L. Barabasi, J Tavernier, D E. Hill, and M Vidal. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322: 104–110, 2008.
- A Yuille and A Rangarajan. The convex-concave computational procedure (cccp). In *Advances in Neural Information Processing Systems 14*, , 2002.
- L V Zhang, S L Wong, O D King, and F P Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, 5: 38, 2004.
- L V Zhang, O D King, S L Wong, D S Goldberg, A Y Tong, G Lesage, B Andrews, H Bussey, C Boone, and F P Roth. Motifs, themes and thematic maps of an integrated *saccharomyces cerevisiae* interaction network. *J Biol*, 4: 6, 2005.

XM Zhao, L Chen, and K Aihara. Protein function prediction with the shortest path in functional linkage graph and boosting. *International journal of bioinformatics research and applications*, 4: 375–84, 2008.

Servisiae והדגמתי איך ניבוי בו זמני כזה מאפשר זרימה של אינפורמציה בין האינטראקציות וניבוי מוצלח יותר של אינטראקציות שלא נצפו. למרות זאת, יישומה של השיטה על רשת האינטראקציות המלאה דורש בנייה של מודל גדול מאד שיכלול מיליוני אינטראקציות וכן חישוב יעיל של שאילתות סטטיסטיות על המודל הזה. לכן, בחלקה השני של עבודת הדוקטורט התמקדתי בפיתוח כלים שיאפשרו ייצוג ריאלי של מודלים גדולים מאד וכן יקרבו באופן יעיל חישובים של שאילתות סטטיסטיות רלוונטיות. כמו כן, הדגמתי איך הכלים החישוביים שפיתחתי מאפשרים למידה של תכונות המודלים בהתבסס על תוצאות של ניסויים רחבי הקף, תוך התחשבות באי הודאות שנובעת מהרעש הניסויי. במהלך העבודה יישמתי את כל השיטות שפיתחתי, וכן שיטות אחרות להסקה סטטיסטית, באמצעות יצירת ספריית קוד פתוח. ספרייה זו מאפשרת יישום של השיטות האלה גם לשאלות מדעיות אחרות וכן מקלה מאוד על הוספת שיפורים והרחבות לשיטות אלה. עד היום ספריית הקוד הורדה על ידי יותר מ-300 משתמשים ונעשה בה שימוש בתחומים שונים החל מאלגוריתמים לתכנון חלבונים סינטטים וכלה בזיהוי אובייקטים בתמונות רועשות.

העבודה שביצעתי בחלק האחרון של עבודת הדוקטורט התמקדה בנייתו של אינטראקציות גנטיות. בעבודה זו, בשיתוף עם רותי רינות, הדגמנו כיצד ניתוח התכונות הגלובליות של רשת האינטראקציות הגנטיות בין חלבונים מוביל להסקת מסקנות ביולוגיות קונקרטיות. פיתחנו אלגוריתם שמשמש בתוצאות של ניסויים רחבי הקף המודדים אינטראקציות גנטיות, על מנת לאפיין ארגון היררכי של הגנים למודולים פונקציונאליים. בהמשך, השתמשנו בתוצאות של סריקות גנטיות שבוצעו תחת מגוון של גירויים כימיים ופיסיקאליים על מנת לשפוך אור על התפקיד הביולוגי של מודולים ספציפיים. מכיוון שסריקות גנטיות כמו אלה שניתחנו מתבררות עתה ככלי ניסויי יעיל ומבוצעות כחלק מהרבה מחקרים, השיטות שפיתחנו הן בעלות ערך רב וניתן יהיה ליישם אותן לניתוח תוצאות ניסוייות מיצורים גבוהים יותר וכן להיעזר בהן על מנת להפיק ידע ביולוגי מתוצאות אלה.

תקציר

כל היצורים החיים מורכבים מתאים וחולקים את אותם מנגנונים תאיים בסיסיים. תאים שונים, בין אם באותו יצור או ביצורים שונים, יכולים להיות שונים מאוד בתפקוד, בצורה ובמורכבות שלהם, אבל הם מורכבים בדיוק מאותן אבני בניין: RNA, DNA וחלבונים. החלבונים לוקחים חלק חשוב בכל התהליכים התאיים: הם משתתפים במסלולי העברת אותות, מבקרים תהליכים רבים בתא, מאיצים תגובות כימיות ועוד. ברוב התהליכים האלה, חלבונים שונים משתפים פעולה, על ידי יצירת קומפלקסים בגדלים משתנים, שינוי חלבונים אחרים, שינוע של חלבונים וחומרים אחרים למיקומים תאיים שונים ועוד. זיהוי ואפיון כל האינטראקציות בין החלבונים הכרחיות להבנת אופן תפקודו של התא. בעשור האחרון, פיתוחן של טכנולוגיות חדשות אפשר מדידה ניסויית רחבת הקף של אינטראקציות אלה. מחקרים רבים השתמשו בתוצאות של ניסויים אלה על מנת לאפיין את תפקידם של חלבונים ותהליכים שונים בתא וגם כדי ללמוד על המאפיינים הגלובליים של רשתות האינטראקציה. אולם, הרעש האופייני לניסויים רחבי הקף מקשה מאוד על הניתוח של תוצאות ניסויים אלה, ומציב אתגר בפני פיתוחם של כלים חישוביים שיוכלו להתמודד עם התוצאות הללו ולהוביל לתובנות ביולוגיות חדשות.

בחלק הראשון של עבודת הדוקטורט שלי השתמשתי במודלים גרפיים יחסיים על מנת להציע שיטה סטטיסטית חדשה שנועדה לייצג רשתות אינטראקציה. השיטה שפיתחתי מאפשרת לחקור את התכונות הגלובליות של הרשת תוך כדי התחשבות באי הוודאות הנובעת מהרעש בניסויים רחבי ההקף. השיטה שפיתחתי מאפשרת ניבוי בו זמני של כל האינטראקציות בהנתן תוצאות של ניסויים המודדים אינטראקציות וכן תכונות אחרות של חלבונים (מיקום תאי, תפקיד וכדומה). יישמתי את השיטה על תוצאות רועשות של מדידת אינטראקציות בין חלבונים ומיקומי חלבונים בקנה מידה גדול בשמר האופים *Sacharomices*

עבודה זו נעשתה בהדרכתם של

פרופ' חנה מרגלית ופרופ' ניר פרידמן

הבנת רשתות אינטראקציה בין חלבונים

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת

אריאל חיימוביץ'

הוגש לסינאט האוניברסיטה העברית בירושלים.

ספטמבר 2010