

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, while Pre7 is not observed in either of those compartments; see Fig. 1a. Based on this information alone, we would probably conclude that an interaction between the two proteins is improbable. However, additional information on related proteins may be relevant. For example, interactions of Pre5 and Pup3 with both Pre9 and Pre7 were reported by large scale assays (Mewes *et al.*, 1998; Sprinzak and Margalit, 2001);

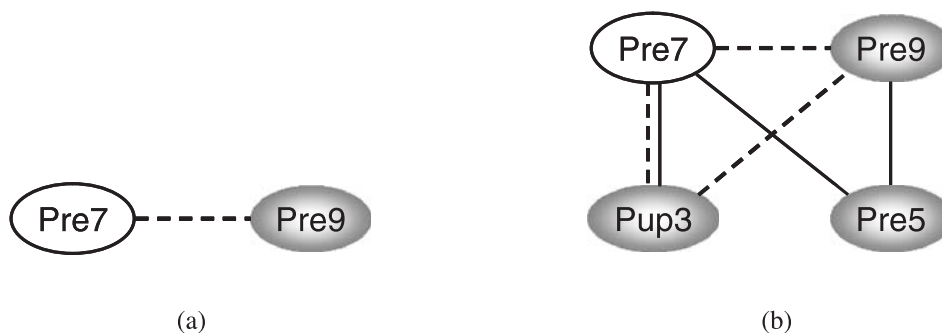


FIG. 1. Dependencies between interactions can be used to improve predictions. (a) A possible interaction of two proteins (Pre7 and Pre9). Pre9 is localized in the cytoplasm and in the nucleus (light gray) and Pre7 is not annotated to be in either one of those. This interaction was predicted by a computational 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

$$P(\mathcal{X} = \mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} e^{\psi_c(\mathbf{x}_c)} \quad (1)$$

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

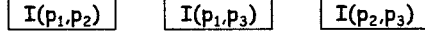
$$P(\mathcal{X}) = \frac{1}{Z} \prod_{p_i, p_j \in \mathcal{P}} e^{\psi_{i,j}(I_{p_i, p_j})} \prod_{l \in \mathcal{L}, p_i \in \mathcal{P}} e^{\psi_{l,i}(L_{l, p_i})} \prod_{l \in \mathcal{L}, p_i, p_j \in \mathcal{P}} e^{\psi_{l,i,j}(I_{p_i, p_j}, L_{l, p_i}, L_{l, p_j})} \quad (3)$$

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.

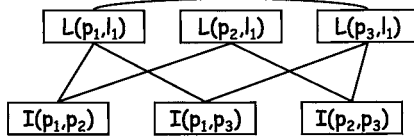
2.2. Noisy sensor models as directed potentials

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



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

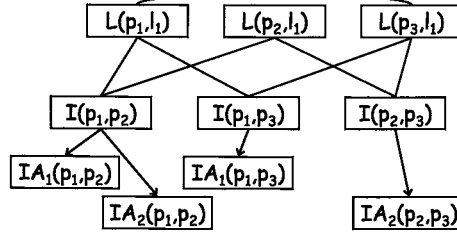
(a)



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

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

(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:

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

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

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

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 $LA_{l,p}$, 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

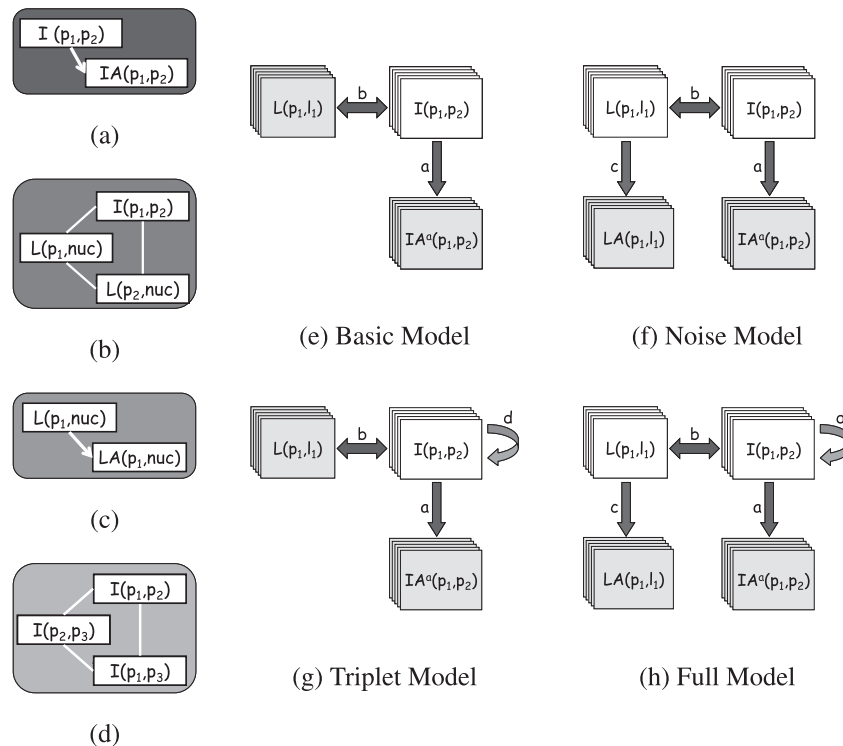


FIG. 3. Protein-protein interaction models. In all models, a plain box stands for a hidden variable, and a shadowed box represents an observed variable. The model consists of four classes of variables and four template potentials that relate them. **(a)** Conditional probability of an interaction assay 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

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

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).

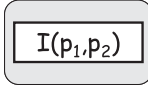
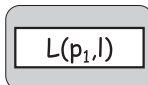
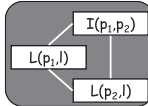
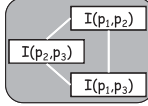
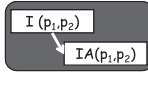
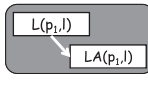
Potentials	Free parameters
Univariate interaction 	$\psi(I_{p_1, p_2} = 1)$
Univariate localization 	For each cellular compartment l : $\psi_l(L_{l, p_1} = 1)$
Colocalization 	For each cellular compartment l : $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 1, L_{l, p_2} = 1)$ $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 1, L_{l, p_2} = 0)$ $\psi_l(I_{p_1, p_2} = 1, L_{l, p_1} = 0, L_{l, p_2} = 0)$ * 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 $b(\mathbf{x}_c) \equiv P(\mathbf{x}_c)$. 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

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

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.

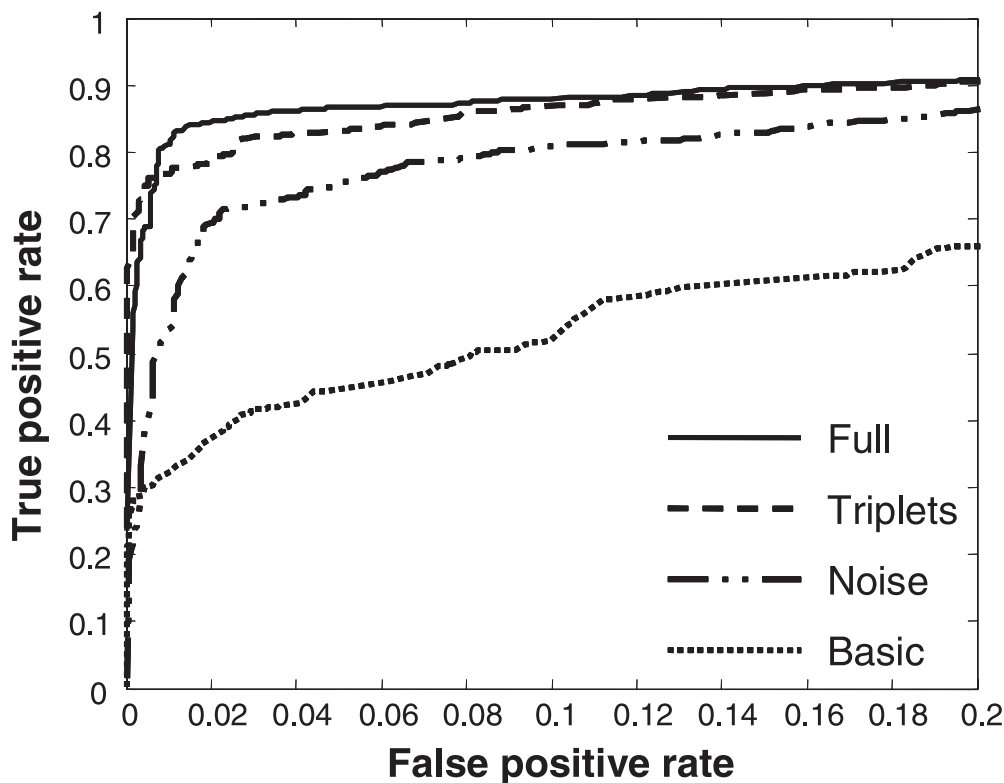


FIG. 5. Test performance (based on 4-fold cross validation) of the different models we evaluate. Shown is the true positive rate vs. the false positive rate for four models: **Basic** with just interaction, interaction assays, and localization variables; **Noise** that adds the localization assay variables;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

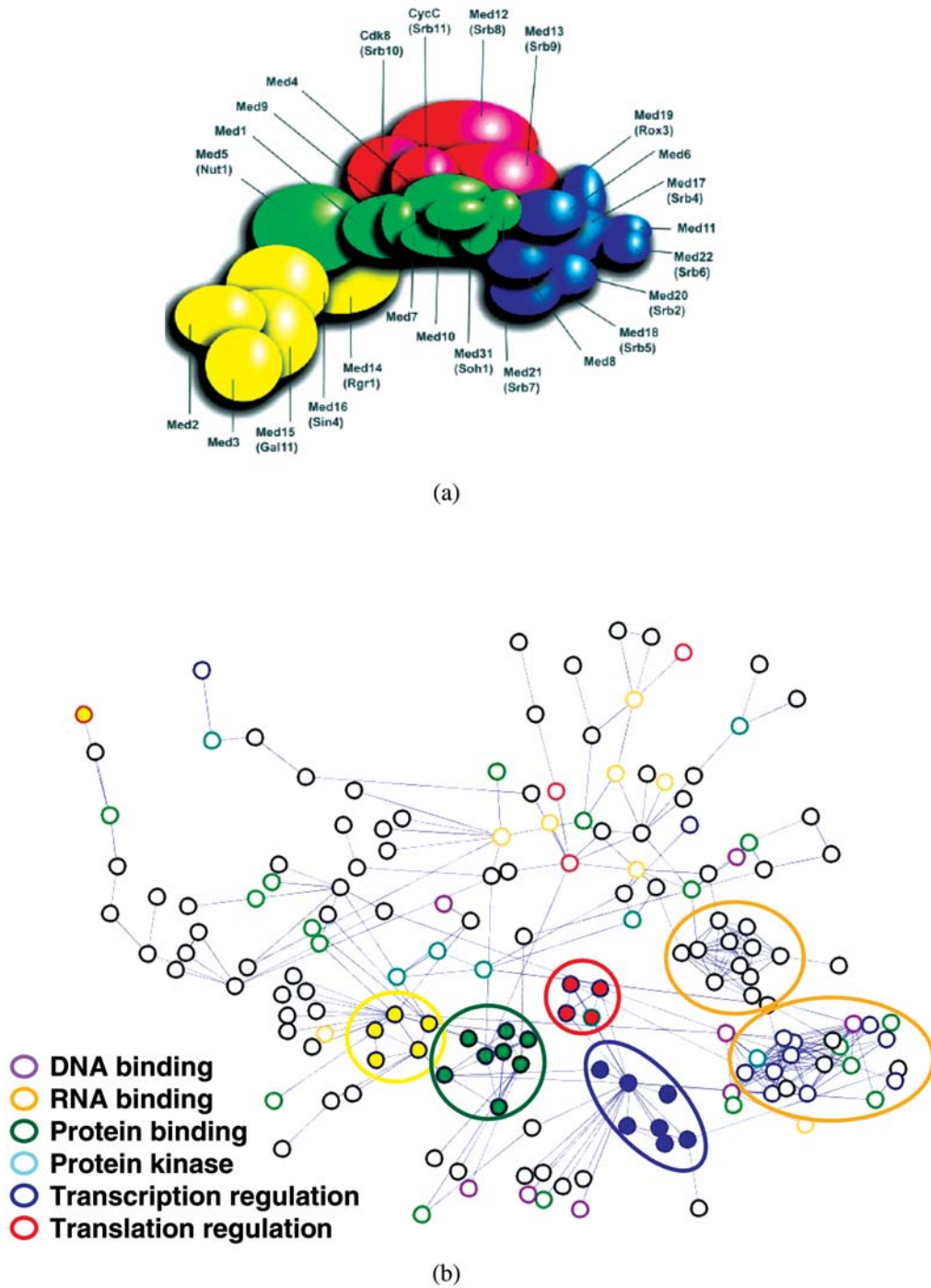


FIG. 8. (a) The mediator complex (taken from Figure 8b of Gugliemi *et al.* (2004) with permission). (b) Part of the interaction network predicted by our method (shown are interactions predicted with probability ≥ 0.5). Nodes are colored according to their GO annotation, and mediator complex subunits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