# Towards an Integrated Protein-protein Interaction Network

Ariel Jaimovich[1,2], Gal Elidan[3], Hanah Margalit[2,*], and Nir Friedman[1,*]

[1] School of Computer Science & Engineering, The Hebrew University, Jerusalem, Israel
[2] Hadassah Medical School, The Hebrew University, Jerusalem, Israel
[3] Computer Science Department, Stanford University, Stanford, California, USA

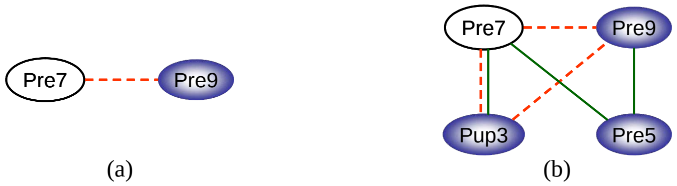
**Abstract.** Protein-protein interactions play a major role in most cellular processes. Thus, the challenge of identifying the full repertoire of interacting proteins in the cell is of great importance, and has been addressed both experimentally and computationally. Today, large scale experimental studies of interacting proteins, 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 Random Fields, we build a unified probabilistic model that includes all of these elements. We show how we can learn our model properties efficiently 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.

## 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 in the cell [32, 27]. 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* [31, 13]

[*] Contact authors: hanah@md.huji.ac.il, nir@cs.huji.ac.il.



(a) (b)

**Fig. 1.** Dependencies between interactions can be used to improve predictions. (a) Shows a possible interaction of two proteins (Pre7 and Pre9). Pre9 is localized in the cytoplasm and in the nucleus (dark blue) and Pre7 is not annotated to be in either one of those. This interaction was predicted by a computational assay ([26], shown by a dashed red line). When looking only at this evidence, we might assign to this interaction a low probability. (b) Introduces two new proteins which were also found to interact with Pre9 and Pre7 either by a computational assay [26] (shown again by a dashed red line ) or experimental assays [20] (shown by a solid green line). By looking at this expanded picture, we can both hypothesize about the localization of Pre7 and increase the reliability of the interaction between Pre9 and Pre7.

and *tandem affinity purification* [24] 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 [23] or mRNA expression [6]. 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 [26].

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* [32] and Sprinzak *et al* [27] 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. These observations 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 [1], naive Bayesian classifiers [14] and decision trees [36].

These methods, while offering a combined hypothesis, still ignore possible dependencies between different protein-protein interactions. In this paper, we argue that by explicitly modeling such dependencies in the model, 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 [26]. However, according to a large-scale localization assay [11], 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. Thus, a naive examination of this interaction alone (as in Figure 1a), might assign it a low probability. However, we can gain more confidence by looking at related interactions. For example, interactions of Pre5 and Pup3 with both Pre9 and Pre7 were reported by large scale assays [20, 26]; see Figure 1b. These observations suggest that these proteins might form a complex. Moreover, both Pre5 and Pup3 were found both in the nucleus and in the cytoplasm, implying that Pre7 can possibly be localized in these two compartments, thus increasing our belief in the existence of an interaction between Pre9 and Pre7. Indeed, this intuition is confirmed by other interaction [9] and localization [17] assays. This example illustrates two types of inferences that our model can offer. First, certain patterns of interactions (*e.g.*, within complexes) might be more probable than others. Second, an observation on one interaction can provide information about the attributes of a protein (cellular localization in this example), which in turn can influence the likelihood of other interactions.

We present a unified probabilistic model for learning an integrated protein-protein interaction network. We build on the language of relational probabilistic models [8, 28] 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 the protein-protein interaction network that are demonstrated in the example of Figure 1. We map protein-protein interactions, interaction assays, and other protein attributes into random variables, and use the language of *Markov Random Fields* to represent the joint distribution over them. We now review Markov Random Field models, and the specific models we construct for modeling protein-protein interaction networks.

**Markov Random Fields**

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a finite set of random variables. A Markov Random Field over $\mathcal{X}$ describes a joint distribution by a set of potentials $\Psi$. Each potential $\psi_c \in \Psi$ defines a measure over a set of variables $\boldsymbol{X}_c \subseteq \mathcal{X}$. We call $\boldsymbol{X}_c$ the *scope* of $\psi_c$. The potential $\psi_c$ quantifies local preferences about $\boldsymbol{X}_c$ by assigning a numerical value to each joint assignment of $\boldsymbol{X}_c$. Intuitively, the larger the value the more likely the assignment. The joint distribution is defined by combining the preferences of all potentials

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

where $\boldsymbol{x}_c$ refers to the projection of $\boldsymbol{x}$ onto the subset $\boldsymbol{X}_c$, 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, and in some cases efficient probabilistic computations.

Using this language to describe protein-protein interaction networks requires defining relevant random variables. 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 $\mathcal{P} = \{p_i, \ldots, p_k\}$, 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 Random Field 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 preference for the associated interaction. This model by itself is overly simplistic as it views interactions as independent from one another.

We can extend the model by introducing variables that denote protein attributes that can influence the probability of interactions. Here we consider cellular localization as an example of such an attribute. The intuition is clear: 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 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 there is no importance to the order of $p_i$ and $p_j$, 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 Random Field of the form:

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

### 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 discuss the noisy relations between an interaction and its assay readout within the model. For each interaction assay $a \in \mathcal{A}$ aimed towards evaluating the existence of an interaction between the proteins $p_i$ and $p_j$, we define a binary random variable $IA^a_{p_i,p_j}$ (defined with the same logic as $I_{p_i,p_j}$). [4] It is natural to view the assay variable $IA^a_{p_i,p_j}$ as a noisy sensor of the real interaction $I_{p_i,p_j}$. In this case, we can use a *conditional distribution* $P(IA^a_{p_i,p_j} \mid I_{p_i,p_j})$ that captures the probability of the observation given the underlying state of the system. Conditional probabilities have several benefits. First, due to local normalization constraints, the number of free parameters of a conditional distribution is smaller (2 instead of 3 in this example). Second, since $P(IA^a_{p_i,p_j} = 0 \mid I_{p_i,p_j}) + P(IA^a_{p_i,p_j} = 1 \mid I_{p_i,p_j}) = 1$, such potentials

[4] Note that this random variable is not necessarily symmetric, since for some assays (*e.g.*, yeast two hybrid) $IA^a_{p_i,p_j}$ and $IA^a_{p_j,p_i}$ represent the results of two different experiments.

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 $IA^a_{p_i,p_j}$, 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* [2]. Here we examine a simplified version of Chain Graphs where a dependent variable associated with a conditional distribution (i.e., $IA^a_{p_i,p_j}$) 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:

$$P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{X}) \prod_{p_i,p_j \in \mathcal{P}, a \in \mathcal{A}} P(IA^a_{p_i,p_j} | I_{p_i,p_j}) \tag{3}$$

where $P(\mathcal{X})$ is the Markov Random Field of Equation 2.

### Template Markov Random Fields

Our aim is to construct a Markov Random Field over large scale protein-protein interaction networks. 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 $6000^2$ 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 apply the same "rules" (potentials) 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 [8, 28] in that they specify a recipe with which a concrete Markov Random Field 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 then using a different potential $\psi_{l,i,j}$ for each protein pair $p_i$ and $p_j$, we use a single potential $\psi_l$. 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 what set of potentials 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 value of $l$ 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.

### Protein-protein Interaction Models

The discussion so far defined the basis for a simple template Markov Random Field for the protein-protein interaction network. The form given in Equation 3 relates protein interactions with multiple interaction assays and protein localizations. 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 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 interaction assay variables and the localization variables, then 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., [14]). We call this the **basic** model (see Figure 2e).

We now consider two extensions to this 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 (Figure 2c). 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. 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 belief in the localization of $p_i$ and thereby influence the belief about the interaction between $p_i$ and another protein, $p_k$. We use the name **Noise** model to refer to the basic model extended with localization assay variables (see Figure 2f).

The second extension to the basic model is to directly capture dependencies between interaction variables. We do so by introducing 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 $\psi_3(I_{p_i,p_j}, I_{p_i,p_k}, I_{p_j,p_k})$ (Figure 2d). This model is known in the social network literature as the *triad model* [7]. Such a triplet potential can capture properties such as preferences for (or against) adjacent edges, as well as transitive closure of adjacent edges. Given $\mathcal{P}$, the induced Markov Random Field 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 edges in any order). Thus, the actual number of parameters for $\psi_3$ is four – one when all three edges 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 Figure 2g). Finally, we use the name **Full** model to refer to the basic model with both the extensions of **Noise** and **Triplet** (see Figure 2h).

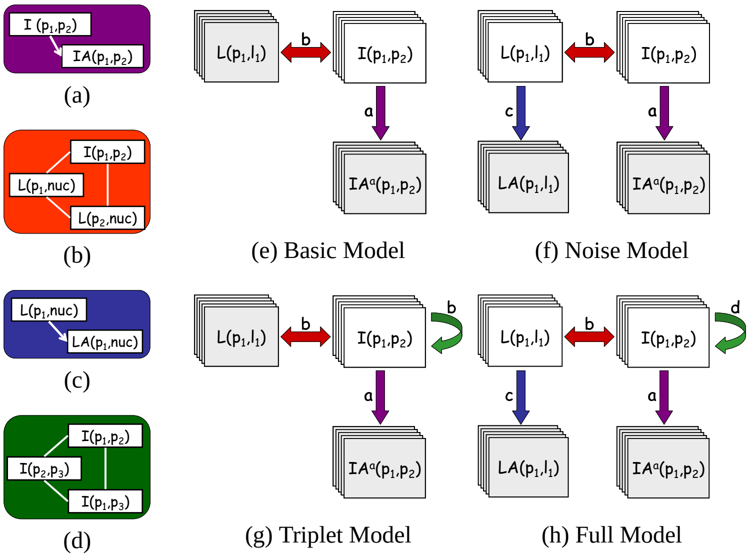


**Fig. 2.** Protein-protein interaction models. In all models, a plain box stands for a hidden variable, and a shadowed box represents an observed variable. The model consists of four classes of variables and four template potentials that relate them. (a) conditional probability of an interaction assay 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.

## 3 Inference and Learning

### Inference

The task of inference is to compute queries about the joint distribution represented by a model. Two queries that will be relevant for us are computing the *likelihood* $P(\boldsymbol{e})$ of an assignment $\boldsymbol{e}$ to a set of variables $\boldsymbol{E}$, and computing the *posterior probability* $P(\boldsymbol{X} \mid \boldsymbol{e})$ over a set of variables $\boldsymbol{X}$, given the evidence $\boldsymbol{e}$. Specifically, in our model we are interested in the likelihood of the observations (interaction assays and localization assays) and the posterior probability of the hidden variables (interactions, localizations), given these observations. When reasoning about the interaction map over thousands of proteins, our method constructs models that involve tens or hundreds of thousands of variables. Performing inference in these models is a serious challenge.

In general, although exact inference algorithms are known, these are tractable only in networks with particular structures. The networks we construct here are not amenable to exact inference. Thus, we must resort to approximate inference (*e.g.*, [15]). In this work we rely on the simple and efficient *belief propagation* (BP) algorithm [21, 35] that iteratively *calibrates* beliefs by propagation of local messages between potentials with overlapping variables. Although this algorithm is not guaranteed to converge, empirical evidence shows that it often converges in general networks and provides reasonable estimates of marginal probabilities [21].

### Learning

Our approach is data driven, meaning that we use real data to calibrate our model. Our aim is to estimate the parameters of the model from observations. Namely, we are given an observation $\boldsymbol{e}$ over some of the variables in the model and search for the parameters that give the best "explanation" for this observation. To do so we use the *maximum likelihood* principle and find a parameterization so that $\log P(\boldsymbol{e})$ is maximized.

We start with the case of *complete data* where the evidence assigns a value to all the variables in the domain. Recall that our model contains both undirected potentials and conditional probabilities. When learning from complete data, we can separately learn each type of parameters.

For the conditional probabilities, finding the maximum likelihood parameters is a relatively easy task. For example, to estimate the template parameter for $P(IA^a_{p_i,p_j} = 1 \mid I_{p_i,p_j} = 1)$ we simply count how many times we observed in $\boldsymbol{e}$ that both $IA^a_{p_i,p_j}$ and $I_{p_i,p_j}$ equals one for some $i$ and $j$, and normalize by the number of times we observe $I_{p_i,p_j}$ equals one.

Finding the maximum likelihood parameters for undirected potentials is a much harder task. Although the likelihood function is concave, there is no closed form formula that returns the optimal parameters. A common heuristic is a gradient ascent search in the parameter space. To perform such a search we need to repeatedly evaluate both the likelihood and the partial derivatives of the likelihood with respect to each of the parameters. For an entry in a specific potential $\psi_c(\boldsymbol{x}_c)$, the gradient of the log-likelihood can be written as:

$$\frac{\partial \log P(\boldsymbol{e})}{\partial \psi_c(\boldsymbol{x}_c)} = \hat{P}(\boldsymbol{x}_c) - P(\boldsymbol{x}_c) \tag{4}$$

That is, it is equal to the difference between the the empirical count $\hat{P}(\boldsymbol{x}_c)$ of the event $\boldsymbol{x}_c$ and its probability $P(\boldsymbol{x}_c)$ according to the model (before we make observations) [4]. The first quantity is directly observed, and the later one requires inference.

Recall that in template models many potentials share the same parameters. Using the chain rule of partial derivatives, it is easy to see that if $\psi_c(\boldsymbol{x}_c) = \theta$ for all $c \in \mathcal{C}$, then the derivative of the shared parameter $\theta$ is $\frac{\partial \log P(\boldsymbol{e})}{\partial \theta} = \sum_{c \in \mathcal{C}} \frac{\partial \log P(\boldsymbol{e})}{\partial \psi_c(\boldsymbol{x}_c)}$. Thus, the derivatives with respect to the template parameters are aggregates of the derivatives of the corresponding entries in the potentials of the model. We can compute these derivatives by performing a single invocation of belief propagation to evaluate (an approximation to) all the terms that appear in the gradient of the likelihood. This invocation also computes an approximation to the likelihood itself. The computational bottleneck for learning is thus the multiple calls to the approximate inference procedure for estimating the likelihood and its gradients for different values of the parameters.

In the context of our models, we introduce additional constraints on learned template potentials. These constraints reduce the number of free parameters we need to learn, and ensure that the model captures the semantics we attribute to it. First, as discussed in Section 2, we require potentials to be symmetric about protein arguments. This implies that some entries in the template potential share a single parameter. We learn the shared parameter using the methods discussed above. Second, in some cases we force the potential to be indifferent to specific assignments. For example, when looking at the relation between $I_{p_i,p_j}, L_{l,p_i}$ and $L_{l,p_j}$, we require that if $L_{l,p_i} = L_{l,p_j} = 0$

(i.e., both proteins are *not* in the cellular localization $l$), then the potential has no effect on $I_{p_i,p_j}$. This is done by fixing the value of the relevant potential entries to $0$, and not changing it during parameter optimization.

In practice, learning is further complicated by the fact that our observations are incomplete — we do not observe all the variables in the model (*i.e.*, we do not observe the real localizations but only the assays). To deal with partial observations, we use the standard method of *Expectation-Maximization* (EM) [22]. The basic idea is that given the current parameters of the model, we can “guess” the unobserved values. We can then apply complete data techniques to the completed dataset to estimate new parameters. In our case, this procedure proceeds by iterating two steps until convergence.

- **E-step** Infer (using Loopy Belief Propagation) the marginal probabilities of the random variables that are missing in the evidence given the evidence, and the current set of parameters. Use the resulting probabilities to estimate *expected* empirical counts for the relevant events.
- **M-step** Maximize parameters using method for complete data, using the estimated counts instead of actual counts. In our case, this implies direct estimation of the conditional probabilities, and performing conjugate gradient search to find new parameters.

The theory of EM guarantees that in each iteration the likelihood increases until convergence to a local maximum [22].

## 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 [31, 13]. The second is a large scale effort to curate direct experimental results from the literature about protein complexes [20]. The third is a collection of computational predictions based on correlated domain signatures learned from experimentally determined interacting pairs [26]. The fourth is a large scale experimental assay examining protein localization in the cell using GFP-tagged protein constructs [11]. 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 interaction. In addition, we have a random variable for each of the four possible localizations of each protein, and yet again another variable corresponding to each localization assay. A model for all $6000$ 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* [32] who ranked $80,000$ 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 $2000$ highest ranked protein pairs as “true” interactions. These 2000 interactions involve $867$

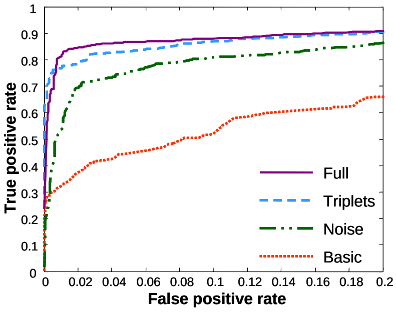


**Fig. 3.** Test performance of four models in a 4-fold cross validation experiment. Shown is the true positive vs. the false positive rates trade-off for four models: **Basic** with just interaction, interaction assays, and localization variables; **Noise** that adds the localization assay variables; **Triplets** that adds a potential over three interactions to **Basic**; **Full** that combines both extensions.

proteins. The selection of negative (non-interacting) 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 non-interacting. Since the number of such non-interacting protein pairs is very large, we randomly selected pairs from the $867$ proteins, and collected $2000$ pairs that do not appear in von Mering's ranking as "true" non-interacting pairs. Thus, we have $4000$ interactions, of these, half interacting and half non-interacting. 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 want to test our predictions on interactions that were not used for learning the model parameters. We use a standard $4$-fold cross validation technique, where in each iteration we learn the parameters using $1500$ positive and $1500$ 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 $1000$ 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 Figure 2). Figure 3 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: above 85% true positive rate with just a 10% false positive rate. This is in contrast to the baseline model that achieves less than half of the true positive rate with the same amount of false positives.

To evaluate the robustness of the results to the specific setup we used, we applied the learned parameters in additional settings. First, we chose a set of $4000$ positive examples and $10000$ negative examples and tested the quality of our predictions. Second, to deal with the concern that in real life we might have fewer observed interactions, we

| | **Basic** model | **Noise** model |
|---|---|---|
| Interaction | $0$ | $-0.02$ |
| Nucleus | $-1.13$ | $-0.91$ |
| Cytoplasm | $-1.34$ | $-1.13$ |
| Mitochondria | $-1.96$ | $-2.04$ |
| ER | $-2.52$ | $-2.52$ |

(a) Univariate potentials

| | **Basic** model | | **Noise** model | |
|---|---|---|---|---|
| localization | $L_{l,p_i} = 1$<br>$L_{l,p_j} = 0$ | $L_{l,p_i} = 1$<br>$L_{l,p_j} = 1$ | $L_{l,p_i} = 1$<br>$L_{l,p_j} = 0$ | $L_{l,p_i} = 1$<br>$L_{l,p_j} = 1$ |
| Nucleus | -0.47 | 0.66 | -0.91 | 1.15 |
| Cytoplasm | -0.66 | -0.02 | -0.94 | 1.27 |
| Mitochondria | -0.71 | 1.26 | -0.99 | 1.38 |
| ER | -0.82 | 1.18 | -0.73 | 1.16 |

(b) Localization to interaction

**Fig. 4.** Examples of learned parameters in two of our models. (a) Univariate potential for interactions $I_{p_i,p_j}$ and localization $L_{l,p_i}$. Shown is the difference between the potential values when the variable is set to $1$ and when it is set to $0$. (b) The potential between $I_{p_i,p_j}$ and $L_{l,p_i}, L_{l,p_j}$ for different localizations. Shown is the difference between the potential values when the interaction variable is set to $1$ and when it is set to $0$. As we can see, 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.

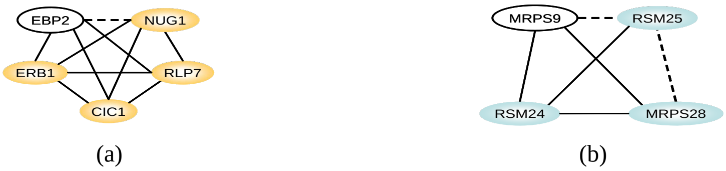
repeated our evaluations, but without use the evidence of the training interactions when making predictions on test interactions. In both cases the ROC curves are quite similar to Figure 3 with a slight decrease in sensitivity (especially in the second setting).

As an additional test, we repeated the original cross validation experiments with reshuffled localization data. As expected, the performance of the basic model decreased significantly. The performance of the full model, however, did not alter much. A possible explanation is that the training “adapted” the hidden localization variables to capture other 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. It also reinforces the caution needed in interpreting what hidden variables represent.

We can gain better insight into the effect of adding a noisy sensor model for localization by examining the estimated parameters (Figure 4). As a concrete example, consider the potentials relating an interaction variable with the localization of the two relevant proteins in Figure 4b. In both models, when only one of the proteins is localized in the compartment, non-interaction 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 more confident in the localization attributes.

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 [11]. Our model predicts that $482$ proteins are nuclear. Of these, $389$ proteins were observed as nuclear, $36$ are nuclear according to YPD [3], $45$ 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.

To get a better sense of the performance of the model, we consider specific examples where the predictions of the full model differ from those of the basic model.



(a)

(b)

**Fig. 5.** 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. Orange colored nodes represent proteins that are localized in the nucleus and blue colored ones represent proteins that are localized in the mitochondria. Uncolored 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, that also interacts with MRPS28.

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 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 Figure 5a. 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 [3, 32].

Another, more complex 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 Figure 5b, our model predicts that MRPS9 is mitochondrial, and that both interactions occur. Importantly, these predictions are again supported by independent results [3, 32]. 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 here.

## 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 a significant improvement in 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 in prediction. Finally, we have shown how evidence can propagate in the model in complex ways leading to novel hypothesis the 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 [8, 28]. In particular, our work is related to applications of these models to *link prediction* [10, 30]. 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* [7]. Recently, other studies tried to incorporate global views of the interaction network when predicting interactions. For example, Iossifov *et al* [12] propose 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. Other recent studies employ variants of Markov random fields 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 [5, 18, 19] and clustering interacting co-expressed proteins [25]. 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* [4]. Such methods can also allow us to discover high-order dependencies between interactions and protein properties.

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.*, [34]) as well as other related alternatives [16, 33], 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 random fields can lead to a significant boost in prediction accuracy [29]. These methods, however, apply exclusively to fully observed training data. Extending these method 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 necessary challenge 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 we want to be able to infer hidden entities. Such a probabilistic integrative model can elucidate the intricate details and general principals of protein-protein interaction networks.

**Acknowledgments** We thank Aviv Regev, Noa Shefi, Einat Sprinzak, Ilan Wapinski, 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 Isreal Science Foundation (ISF), European Union Grant QLRT-CT-2001-00015, and the National Institute of General Medical Sciences (NIGMS).

## References

1. J. R. Bock and D. A. Gough. Predicting protein–protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.
2. W. Buntine. Chain graphs for learning. In *Proc. Uncertainty in Art. Intel.*, p. 46–54. 1995.
3. M.C. Costanzo, *et al*. YPD, POMBEPD, and WORMPD: model organism volumes of the bioknowledge library, an integrated resource for protein information. *Nuc. Acids Res.*, 29:75–9, 2001.
4. S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pat. Anal. Mach. Intel.*, 19:380–393, 1997.
5. M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. *J. Comput. Bio.*, 11:463–75, 2004.
6. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–8, 1998.
7. O. Frank and D. Strauss. Markov graphs. *J. Am. Stat. Assoc.*, 81, 1986.
8. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. Inte. Joint Conf. Art. Intel.*. 1999.
9. A. C. Gavin, *et al* . Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, Jan 2002.
10. L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *Int. Conf. Mach. Learning*. 2001.
11. W. Huh, *et al*. Global analysis of protein localization in budding yeast. *Nature*, 425:686 – 691, 2003.
12. I. Iossifov, M. Krauthammer, C. Friedman, V. Hatzivassiloglou, J.S. Bader, K.P. White, and A. Rzhetsky. Probabilistic inference of molecular networks from noisy data sources. *Bioinformatics*, 20:1205–13, 2004.

13. 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. USA*, 98:4569–4574, 2001.
14. R. Jansen, *et al*. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302:449–453, 2003.
15. M. I. Jordan, ed. *Learning in Graphical Models*. Kluwer, 1998.
16. M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational approximations methods for graphical models. In [15].
17. A. Kumar, *et al*. Subcellular localization of the yeast proteome. *Genes. Dev.*, 16:707–719, 2002.
18. M. Leone and A. Pagnani. Predicting protein functions with message passing algorithms. *Bioinformatics*, 21:239–247, 2005.
19. S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19(Suppl 1):i97–204, 2003.
20. HW Mewes, J Hani, F Pfeiffer, and D Frishman. MIPS: a database for genomes and protein sequences. *Nuc. Acids Res.*, 26:33–37, 1998.
21. K. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. Uncertainty in Art. Intel.*, 1999.
22. R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In [15].
23. M. Pellegrini, E. M. Marcotte, and T. O. Yeates. A fast algorithm for genome-wide analysis of proteins with repeated sequences. *Proteins*, 35:440–446, 1999.
24. 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. Biotech.*, 17:1030–1032, 1999.
25. E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19(Suppl 1):i264–71, 2003.
26. E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *J. Mol. Biol.*, 311:681–692, 2001.
27. E. Sprinzak, S. Sattath, and H. Margalit. How reliable are experimental protein-protein interaction data? *J. Mol. Biol.*, 327:919–923, 2003.
28. B. Taskar, A. Pieter Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. Uncertainty in Art. Intel.*, p. 485–492, 2002.
29. B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Adv. Neu. Inf. Proc. Sys.*, 2003.
30. B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Adv. Neu. Inf. Proc. Sys.*, 2003.
31. P. Uetz, *et al*. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.
32. 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.
33. M. J. Wainwright, T. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. In *Proc. Uncertainty in Art. Intel.*, 2002.
34. J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. TR-2002-35, Mitsubishi Electric Research Labs, 2002.
35. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Adv. Neu. Inf. Proc. Sys.*, p. 689–695, 2000.
36. 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.