Using Topology Information for Protein-Protein Interaction Prediction

Adriana Birlutiu$^{(1)}$ and Tom Heskes$^{(2)}$

$^{(1)}$ Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands and Faculty of Science, "1 Decembrie 1918" University, Alba-Iulia, Romania, e-mail: adrianab@cs.ru.nl
$^{(2)}$ Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands, e-mail: tomh@cs.ru.nl

Abstract. The reconstruction of protein-protein interaction networks is nowadays an important challenge in systems biology. Computational approaches can address this problem by complementing high-throughput technologies and by helping and guiding biologists in designing new laboratory experiments. The proteins and the interactions between them form a network, which has been shown to possess several topological properties. In addition to information about proteins and interactions between them, knowledge about the topological properties of these networks can be used to learn accurate models for predicting unknown protein-protein interactions. This paper presents a principled way, based on Bayesian inference, for combining network topology information jointly with information about proteins and interactions between them. The goal of this combination is to build accurate models for predicting protein-protein interactions. We define a random graph model for generating networks with topology similar to the ones observed in protein-protein interaction networks. We define a probability model for protein features given the absence/presence of an interaction and combine this with the random graph model by using Bayes' rule, to finally arrive at a model incorporating both topological and feature information.

Keywords: protein-protein interaction, Bayesian methods, network analysis

1 Introduction

Knowledge about protein-protein interactions (PPIs) is essential to the understanding of the cellular functions and biological processes inside a living cell. Deciphering the entire network of PPIs of an organism is a very complex task since these interactions can only be established by costly and tedious laboratory experiments. Computational techniques for predicting PPIs have become standard tools to address this problem, complementing their experimental counterparts. Accurately predicting which proteins might interact can help in designing and guiding future laboratory experiments. Therefore, developing computational methods that can accurately predict PPIs is currently an active research area.

A number of computational approaches for PPI prediction have been developed over the years. These methods differ in feature information used for PPI prediction, for example genomic data, phylogenetic trees.

A recent trend in computational approaches for predicting PPIs is to frame this problem in a supervised learning setting. That is, information about proteins and labels for protein pairs as interacting or not, supervise the estimation of a function that can predict whether an interaction exists or not between two proteins. PPI prediction can thus be seen as a pattern recognition problem, i.e., find patterns in the interacting protein pairs that do not exist in the non-interacting pairs. This can be further framed as a binary classification problem which takes as input a set of features for a protein pair and gives as output a label: interact or non-interact. Binary classification has been studied extensively in the machine learning community, and many algorithms designed to solve it have been also applied for predicting PPIs, including Bayesian networks [9], kernel-based methods [1, 31], logistic regression [14, 27], SVMs [26] decision trees and random forest based methods [33, 22, 2], metric or kernel learning [31] and [7, 6, 5]. Very recently, other machine learning paradigms, such as, active learning, multi-task learning, and semi-supervised learning, have also been employed for improving the prediction of PPIs [18, 24, 11].

In addition to information about proteins and interactions between them, PPI networks are characterized by several topological properties [10, 15, 4, 21, 28]. Network topology can uncover important biological information that is independent of other available biological information [25, 13]. One of the most important topological properties is the existence of a few nodes in the networks, called hubs, which have many links with the other nodes, while most of the nodes have just a few links. This characteristic is present in PPI networks and also in other real-world networks, such as the internet and citation networks. Topology only has been shown to be able to predict protein functions [17] and PPIs [12] and to complement sequence information in various biological tasks, like for example, homology detection [16]. Summarizing, we can distinguish two types of information that can be used for predicting PPIs: first, information about proteins and labels for protein pairs as interacting or not, and second, information about topological properties of PPI networks. These two sources of information can complement each other and are both valuable for constructing models which can accurately predict interactions between proteins.

In this contribution, we present a principled way of combining topology and feature information for constructing models for predicting PPIs. We combine models that have been previously used for modeling each type of information separately. We use a random graph generator for addressing the topology information and a naive Bayes model for addressing the feature information. We show that by making a few simplifying assumptions, both topological and protein information can be incorporated and we show experimentally that this improves the prediction accuracy in two PPI networks.

2 Models and Methods

The approach that we use to combine topology and feature information is graphically summarized in Figure 1. It consists of a random graph generator model and a naive Bayes model which are combined using Bayes' rule to finally arrive to a logistic regression model (we will ignore for the moment the details of this figure but come back to it throughout the section). The random graph generator gives rise to networks which based on topology can all be plausible hypotheses for the PPI network that we want to reconstruct. Incorporating the actual data will reduce this set of plausible hypotheses to just a few, out of which we can pick the one which has the highest likelihood. We implement this in a Bayesian framework by treating our random graph model as a prior and define a probability model for the features given the absence/presence of an edge and combine these two using Bayes' rule, to finally arrive at a model incorporating both topological and feature information. The way in which each of these models is constructed and then combined is detailed in the rest of this section.

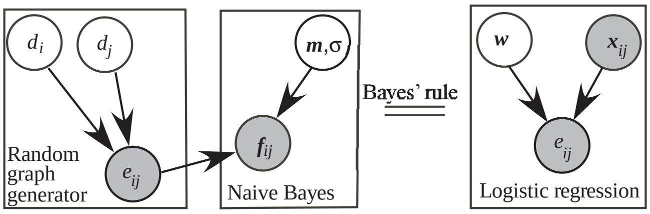

Fig. 1. Graphical representation of the model which combines topology and feature information. Left box: random graph generator model. Center box: naive Bayes model. Right box: the result of applying Bayes' rule, the model which combines topology and feature information.

2.1 Topological Properties of PPI Networks

We will focus on one essential topological characteristics of PPI networks: the node degree distribution. The degree of a node represents the number of connections the node has with the other nodes in the network. The probability distribution of these degrees over the whole network, $p(k)$, is defined as the fraction of nodes in the network with degree k,

$$p(k) = \frac{N_k}{N} ,$$

where N is the total number of nodes in the network and N_k is the number of nodes with degree k. The majority of real-world networks have a node degree distribution that is highly right-skewed, which means that most of the nodes have low degrees, while a small number of nodes, known as “hubs”, have high degrees. The degree of hubs is typically several order of magnitudes larger than the average degree of a node in the network.

2.2 Random Graph Generator

The first step of our approach is to define a model for generating networks with the node degree distribution similar to the one of PPI networks (the left-hand side box of Figure 1). The random graph generator that we define here is inspired by the general random graph method [3]. The general random graph method assigns each node with its expected degree and edges are inserted according to a probability proportional to the product of the degrees of the two endpoints, i.e., the probability of an edge between two nodes i and j is proportional to the product of the expected degrees of the nodes i and j. We introduce a latent variable, d_i, related to the degree of node i, i.e., d_i is roughly proportional to the degree of node i. Let e_{ij} be a random variable with two possible values: $e_{ij} = 1$ if a link is present between nodes i and j, and $e_{ij} = -1$ if there is no link. In Figure 1, the random variables d_i and d_j are represented by white color circles because they are unobserved while e_{ij} is represented by a gray color circle because it is observed.

Our model generates links in the network as follows,

$$p(e_{ij}|d_i, d_j) \propto (\sqrt{d_i d_j})^{e_{ij}} = \exp\left[e_{ij}\frac{1}{2}(\log d_i + \log d_j)\right] , \tag{1}$$

$$\begin{aligned} p(e_{ij} = 1|d_i, d_j) &\propto \sqrt{d_i d_j} \\ p(e_{ij} = -1|d_i, d_j) &\propto \tfrac{1}{\sqrt{d_i d_j}} \end{aligned}$$

$$\begin{aligned} p(e_{ij} = 1|d_i, d_j) &= \tfrac{p(e_{ij}=1|d_i,d_j)}{p(e_{ij}=1|d_i,d_j)+p(e_{ij}=-1|d_i,d_j)} \\ &= \tfrac{\sqrt{d_i d_j}}{\sqrt{d_i d_j}+\frac{1}{\sqrt{d_i d_j}}} = \tfrac{d_i d_j}{1+d_i d_j} , \end{aligned}$$

$$\begin{aligned} p(e_{ij} = -1|d_i, d_j) &= \tfrac{p(e_{ij}=-1|d_i,d_j)}{p(e_{ij}=1|d_i,d_j)+p(e_{ij}=-1|d_i,d_j)} \\ &= \tfrac{\frac{1}{\sqrt{d_i d_j}}}{\sqrt{d_i d_j}+\frac{1}{\sqrt{d_i d_j}}} = \tfrac{1}{1+d_i d_j} , \end{aligned}$$

In order to generate networks with a desired topology and for computational reasons which will become clear later, we consider a log-normal distribution for d_i,

$$p(\log d_i) = \mathcal{N}(\log d_i; m_0, \sigma_0^2) , \tag{2}$$

where m_0 is a scaling parameter, and the parameter σ_0 controls the shape of the distribution. These parameters can be fit such that the networks randomly generated with the model from Equation (1) have the desired topology. We have defined d_i to be roughly proportional to the degree of node i, thus a log-normal distribution for d_i results in a distribution for the degree of node i which is approximately log-normal, which is similar to what is observed in practice.

In summary, the random graph generator for a given topology performs the following steps.

1. Choose m_0 and σ_0 the parameters of the log-normal distribution for d_i.
2. Draw from this distribution a random sample $(d_1, \ldots, d_N)$ of size N the number of nodes in the network.
3. Based on this sample construct the network by inserting edges with probability given in Equation (1).

2.3 Bayesian Framework for Combining Topology and Feature Information

In order to combine the topology and feature information, we treat the random graph model as a prior and define a probability model for the protein pairs features given the absence/presence of an interaction. We make use of a a naive Bayes model to express the likelihood of a protein pairs feature given the absence/presence of an interaction. The likelihood is thus computed as a product of 1-dimensional Gaussian distributions, each Gaussian distribution expressing the probability of a feature component f_{ij}^k given the edge variable e_{ij} and the parameters mean m_k and variance σ,

$$p(\boldsymbol{f}_{ij}|e_{ij}, \boldsymbol{m}, \sigma) = \prod_{k=1}^{D} \mathcal{N}(f_{ij}^k; m_k e_{ij}, \sigma) \propto \prod_{k=1}^{D} \exp\left(-\frac{(f_{ij}^k - e_{ij} m_k)^2}{2\sigma^2}\right) . \quad (3)$$

We refer to the center box of Figure 1 for a graphical representation of this model. The naive Bayes model defined above treats the features as independent, which might not be the case in practice. Despite this simplifying assumption, the naive Bayes model is known to be a competitive classification method, with similar performance as the closely related logistic regression algorithm.

The posterior distribution for e_{ij} which combines topology and feature information is computed using Bayes' rule as the product between the prior defined in Equation (1) and the likelihood terms defined in Equation (3), i.e.,

$$\begin{aligned} p(e_{ij}|\boldsymbol{f}_{ij}, d_i, d_j) &\propto p(e_{ij}|d_i, d_j)p(\boldsymbol{f}_{ij}|e_{ij}, d_i, d_j) \\ &\propto \exp\left(e_{ij}\frac{1}{2}(\log d_i + \log d_j) - \frac{\sum_k (f_{ij}^k - e_{ij}m_k)^2}{2\sigma^2} \right) \quad (4) \\ &\propto \exp\left(e_{ij}\frac{1}{2}(\log d_i + \log d_j) + \frac{e_{ij}\sum_k f_{ij}^k m_k}{\sigma^2} \right) \quad (5) \\ &\propto \exp\left(e_{ij}(\sum_{k=1}^{D} \frac{f_{ij}^k m_k}{\sigma^2} + \frac{1}{2}\log d_i + \frac{1}{2}\log d_j) \right) \quad (6) \end{aligned}$$

where when going from (4) to (5) we discarded the square terms. In the above, we can ignore any term that does not depend on e_{ij}, since it will only affect the normalization. This includes the term $e_{ij}^2 m_k^2/\sigma^2$, since $e_{ij} \in \{-1, 1\}$. The normalization term does play a role and, when incorporated, leads to Equation (8) below. The unknown quantities of our model are $\frac{m_k}{\sigma^2}$, $k = \{1, \dots, D\}$ and $\log d_i$, $i = \{1, \dots, N\}$, and these will be estimated based on the available training data in a learning procedure that we describe below.

The first step is to adjoin the unknown quantities in a single random variable, that is

$$\boldsymbol{w} = [\frac{m_1}{\sigma^2}, \dots, \frac{m_D}{\sigma^2}, \frac{1}{2}\log d_1, \dots, \frac{1}{2}\log d_N] \,, \quad (7)$$

and the same for the information available, that is protein features and topological information

$$\boldsymbol{x}_{ij} = [\boldsymbol{f}_{ij}, \boldsymbol{t}_{ij}] \,,$$

where $\boldsymbol{t}_{ij}$ is the position vector having 1 on positions i and j and 0 everywhere else. Then, the normalized probability that there is an interaction between the proteins i and j from Equation (6) can be rewritten as

$$p(e_{ij}|\boldsymbol{x}_{ij}, \boldsymbol{w}) = \frac{1}{1 + \exp(-2e_{ij}\boldsymbol{w}^T\boldsymbol{x}_{ij})} \,. \quad (8)$$

Note that in the sum

$$\boldsymbol{w}^T\boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + \sum_{k=1}^{N} w^{D+k} t_{ij}^k \,, \quad (9)$$

the first term on the right-hand side originates from the protein features information and the second term from the topological information.

The unknown parameter $\boldsymbol{w}$ is learned in a Bayesian framework which consists in setting a prior distribution for it, and updating this prior based on observations. The update is performed using Bayes' rule given below

$$p(\boldsymbol{w}|\text{observations}) \propto \prod_{o=1}^{n_{\text{obs}}} p(e_{ij}^o|\boldsymbol{x}_{ij}^o, \boldsymbol{w})p(\boldsymbol{w}) \,. \quad (10)$$

where n_{obs} is the size of the training data, i.e., the number of known interacting/non-interacting protein pairs, and $p(e^o_{ij}|\boldsymbol{x}^o_{ij}, \boldsymbol{w})$ is given in Equation (8). $p(\boldsymbol{w})$ is the prior and we choose it to be a Gaussian distribution

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,.$$

The hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the prior are chosen such that the topological information is included in the model. This is implemented by making the correspondence with the prior for the latent variables d_i. Recall from Equation (7) that $w^{i+D} = \frac{1}{2}\log d_i \,, i = 1, \ldots, N$ and from Equation (2) that $\log d_i$ is normally distributed, consequently w^{i+D} will also be normally distributed, i.e.,

$$w^{i+D} \sim \mathcal{N}\left(\frac{m_0}{2}, \frac{\sigma_0^2}{4}\right) \,, i = 1, \ldots, N \,.$$

The vectors $\boldsymbol{x}_{ij}$ are sparse because their components $\boldsymbol{t}_{ij}$ of dimension N contain only two non-zero elements on positions i and j. This sparsity property can be exploited for making the computations more efficient. Predictions can be done for an unknown interaction between a pair of proteins i', j' characterized by the feature vector $\boldsymbol{x}_{i'j'}$. These predictions can be done either averaging the posterior over $\boldsymbol{w}$ in Equation (8) or by using a point estimate of this posterior, let $\boldsymbol{w}^*$ be the mean of $p(\boldsymbol{w}|\text{observations})$, and computing $p(e_{i'j'}|\boldsymbol{x}_{i'j'}, \boldsymbol{w}^*)$ using Equation (8).

We refer back to the graphical sketch of our model in Figure 1 at the beginning of this section. The box on the left-hand side, corresponds to the random graph generator model. The observation e_{ij}, which expresses the presence or absence of an edge between nodes i and j, depends on the latent variables d_i and d_j which are related to the degrees of nodes i and j. The random graph generator model incorporates feature information through the naive Bayes model with unknown parameters $\boldsymbol{m}$ and σ, represented in the center box. The combination of the two models is obtained using Bayes' rule. The result is shown in the right-hand side box. The unknown quantities d_i, d_j, and $\boldsymbol{m}$, σ are combined in the node $\boldsymbol{w}$ which is unobserved, and $\boldsymbol{f}_{ij}$ together with $\boldsymbol{t}_{ij}$ which is implicitly expressed by indices i and j form the observed quantity $\boldsymbol{x}_{ij}$.

In the experimental evaluation from Section 3 we will compare four models. All the models are based on Equation (10) with a Gaussian prior and likelihood terms of the form given in Equation (8) and they vary in the way of computing the dot product from Equation (9) and on the parameters of the Gaussian prior.

1. Model 1 (Features+Topology): is the model we propose in this work. It makes use of the following dot product

$$\boldsymbol{w}^T\boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f^k_{ij} + \sum_{k=1}^{N} w^{D+k} t^k_{ij} \,, \tag{11}$$

   and a Gaussian prior with mean $\boldsymbol{\mu}^{1:D} = 0$, $\boldsymbol{\mu}^{D+1:N} = -1.5$ and covariance matrix equal to the identity matrix.

2. Model 2 (Features only): uses only information about proteins, and the dot product is computed as

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + w^{D+1} \,. \tag{12}$$

   The second term on the right-hand side of Equation (12) is a bias term to address the unbalancedness of the data. This bias term also corresponds to the second term on the right-hand side of Equation (11); for an edge e_{ij} the contributions in Equation (11) are $w^{D+i} + w^{D+j}$ while in Equation (12) we constraint $w^{D+i} = \frac{1}{2} w^{D+1}$, $\forall i = 1, \ldots, N$. This observation also motivates the choice of the prior for this model: mean $\boldsymbol{\mu}^{1:D} = 0$ and $\mu^{D+1} = -3$ and covariance equal to the identity matrix.
3. Model 3 (Topology only): uses only topology information and the dot product is computed as

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{N} w^k t_{ij}^k \,.$$

   The Gaussian prior is of dimension N with mean equal to the vector $\boldsymbol{\mu}^{1:N} = -1.5$ and covariance matrix equal to the identity matrix. The choice for $\boldsymbol{\mu}^{1:N} = -1.5$ corresponds to the log-normal distribution with $m_0 = -3$, thus to a network with a node degree distribution similar to the one observed in PPI networks.
4. Model 4 (Topology-enriched features): uses the information about proteins and about topology in the following form

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + w^{D+1} \log(\hat{d}_i + 1) + w^{D+2} \log(\hat{d}_j + 1) \,,$$

   where $\hat{d}_i$ and $\hat{d}_j$ are the estimated degrees of nodes i and j computed on the training data. Basically, the features $\boldsymbol{f}_{ij}$ for a pair of proteins i and j are being extended by adding two new columns corresponding to the degrees of nodes i and j computed on the training set. For computational reasons we considered the logarithms of node degrees to which we added 1. The idea behind this model is similar to the one used in [29, 24], i.e., the topological features are added to protein features resulting in an enriched set of features. The features are being standardized and the parameters of the Gaussian prior are set to $\boldsymbol{\mu}^{1:D+2} = 0$ and covariance equal to the identity matrix.

3 Results

In this section we discuss the results of the experimental evaluation of the framework proposed here. We compare the performance obtained using information about proteins only, with the performance obtained using topology information only and with the performance obtained with the combination of the two.

3.1 Data Sets

We used two data sets. Details for each of them are given below.

Yeast Data This data set was borrowed from [5] and it consists of the high confidence physical interactions between proteins highlighted in [30]. The PPI network has 984 nodes (proteins) connected by 2438 links (interactions). We consider all the protein pairs not present in the 2438 interactions as non-interacting. The yeast PPI graph is very sparse, as a result the data is highly unbalanced, with less than 1% from the total examples belonging to the positive class. Each protein has associated a vector of dimension 157 representing gene expression values in various experiments. We constructed the features for protein pairs by summing the individual protein features.

Human Data This data set was created and made available by [23] and consists of protein pairs with an associated label: interact or non-interact. Each pair of proteins is characterized by a 27-dimensional feature vector. The features were constructed based on Gene Ontology (GO) cell component (1), GO molecular function (1), GO biological process (1), co-occurrence in tissue (1), gene expression (16), sequence similarity (1), homology based (5) and domain interaction (1), where the numbers in brackets correspond to the number of elements contributed by the feature type to the feature vector. Unlike positive interactions, non-interacting pairs are not experimentally reported. Thus, a common strategy is to consider as non-interacting pairs a randomly drawn fraction from the total set of potential protein pairs excluding the pairs known to interact. The resulting data set has 14,608 interacting pairs and 432,197 non-interacting pairs. The PPI graph consists of $24,380$ nodes connected by $14,608$ edges. As in the case of the yeast data set, the PPI graph of the human data is very sparse, the interacting pairs represent less then 1% from the possible links in the graph.

Both data sets are highly unbalanced, with 1% and 5% positive pairs for yeast data and human data, respectively. There are classification methods that were designed to address the unbalancedness of data [19]. Specifically, for protein interactions, there are some studies [32, 20] that investigate how to construct non-interacting protein pairs (negative samples).

3.2 Experimental Setup

The experimental setup considered a part of the data for training and the rest for testing. The training data was used to learn the models and the testing data was used to evaluate how good these models can predict PPIs. We randomly sampled a training set containing 1%, 5%, 10% and 20% protein pairs and their labels as interacting or not from the yeast and human data set. The PPI prediction problem was thus transformed in a binary classification problem The training features were standardized to have mean zero and standard deviation of one. This data sample was used to train the classification model (i.e., learn the weight parameter of the logistic regression). The remaining protein pairs were used for testing the performance. These steps were repeated 10 times and average results are reported (mean ± standard deviation).

3.3 Evaluation Measure

Area under the receiver operating characteristic curve (AUC) was used as a measure for evaluating the performance. The receiver operator characteristic (ROC) curve plots the true positive rate against the false positive rate for different thresholds. The AUC statistic can be interpreted as the probability that a randomly chosen missing edge (a true positive) is given a higher score by the method than a randomly chosen pair of proteins without an interaction (a true negative).

3.4 Performance

Table 1 shows the comparison of the performance of the four models discussed. Model 1 represents the Bayesian framework for combining feature and topology information, Model 2 uses only protein information, Model 3 uses only topology information and Model 4 uses protein features which are enriched with node degrees. The comparison was performed for the yeast data (the four upper rows in Table 1) and human data sets (the four lower rows in Table 1). The protocol described in Section 3.2 was used and the averaged AUC scores with their standard deviations are reported. The statistical significance between Model 1 and Model 2 was assessed by using a Mann-Whitney U-test [8] on the AUC values obtained from the two models for 10 random splits of the data into training and testing. A 5% significance level has been considered. The * indicates that the results obtained for Model 1 are significantly better than the results obtained for Model 2.

The results show that the combination of the two sources of information, protein features and topology, gives a better performance than using only one type of information. In particular Model 1 (Features+Topology) performs significantly better than Model 2 (Features only) in most of the cases. Model 1 and Model 4 have a similar performance for human data, and Model 1 performs better than Model 4 for yeast data. An explanation for this is related to how the protein features were constructed in the two cases; for yeast data the features for a protein pair resulted from summing the feature vectors corresponding to the two proteins, while for human data the protein features are more related to the protein pair than to individual proteins. Model 3 (Topology only) uses only the information related to the topology, in particular the property of hub-proteins to interact with many other proteins. Note that you can have the pair of protein A and protein B in training set and the pair of protein A and protein C in the test set, and in this way the algorithm learns which proteins are hubs (and other topological information) and makes predictions based on topology.

The results vary also as a function of the size of the training data. For a small training set the network is not well defined, and we can see that in this case the

Table 1. AUC values (mean ± standard deviation) for the four models: Model 1 represents the Bayesian framework for combining feature and topology information, Model 2 uses only protein information, Model 3 uses only topology information, Mode 4 uses protein features which are enriched by node degrees. The * indicates that the results obtained for Model 1 are significantly better than the results obtained for Model 2. The four upper rows correspond to the yeast data set while the four lower rows correspond to the human data set.

% Train data	Model 1 Features+ Topology	Model 2 Features only	Model 3 Topology only	Model 4 Topology features
1%	0.639 ± 0.014	0.639 ± 0.018	0.577 ± 0.016	0.582 ± 0.022
5%	0.708 ± 0.006	0.697 ± 0.009	0.688 ± 0.010	0.689 ± 0.009
10%	0.731 ± 0.005*	0.712 ± 0.005	0.720 ± 0.006	0.717 ± 0.007
20%	0.746 ± 0.009*	0.719 ± 0.006	0.742 ± 0.009	0.737 ± 0.010
1%	0.863 ± 0.006*	0.851 ± 0.006	0.608 ± 0.014	0.822 ± 0.012
5%	0.909 ± 0.002*	0.859 ± 0.001	0.793 ± 0.007	0.899 ± 0.003
10%	0.931 ± 0.002*	0.861 ± 0.001	0.864 ± 0.005	0.931 ± 0.002
20%	0.952 ± 0.002*	0.862 ± 0.001	0.917 ± 0.003	0.954 ± 0.002

improvement is smaller, but, as we increase the training set, meaning that the knowledge about the network topology increases, the performance obtained by adding the topology information improves more.

4 Conclusion

We introduced a framework for predicting PPI by considering the network structure information. This is a Bayesian framework consisting of a prior distribution over the network topology and likelihood terms for observations about links in the network. In the Bayesian framework in general, and in our case when trying to add topological information, the computational complexity is an issue. In the framework presented here, we managed to find some simplifying assumptions which reduce the computational complexity and at the same time yield a good performance.

References

1. A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(1):38–46, 2005.
2. X.W Chen and M. Liu. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

3. F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.
4. C. Friedel and R. Zimmer. Inferring topology from clustering coefficients in protein-protein interaction networks. *BMC Bioinformatics*, 7:519, 2006.
5. P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc. Inferring biological networks with output kernel trees. *BMC Bioinformatics (PMSB06 special issue)*, 8(Suppl 2):S4, 2007.
6. P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Gradient boosting for kernelized output spaces. In *ACM International Conference Proceeding Series (Proceedings of the 24th International Conference on Machine Learning)*, volume 227, pages 289–296. ACM, 2007.
7. P. Geurts, L. Wehenkel, and d'Alché Buc F. Kernelizing the output of tree-based methods. In *Proceedings of the 23th international conference on Machine learning*, pages 345–352, 2006.
8. M. Hollander and D. Wolfe. *Nonparametric Statistical Methods*. John Wiley & Sons, 1999.
9. R. Jansen, H. Yu, and et al. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.
10. H. Jeong, S.P. Mason, A.-L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.
11. H. Kashima, Y. Yamanishi, T. Kato, M. Sugiyama, and K. Tsuda. Simultaneous inference of biological networks of multiple species from genome-wide data and evolutionary information. *Bioinformatics*, 25(22):2962–2968, 2009.
12. O. Kuchaiev, M. Rasajski, D.J. Higham, and N. Przulj. Geometric de-noising of protein-protein interaction networks. *PLOS Computational Biology*, 5(8), 2009.
13. Z.C Li, Y.H Lai, and et al. Identifying functions of protein complexes based on topology similarity with random forest. *Mol Biosyst*, (10):514–525, 2014.
14. N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao. Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, 5:154, 2004.
15. S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.
16. V. Memisevic, T. Milenkovic, and N. Przulj. Complementarity of network and sequence information in homologous proteins. *Journal of Integrative Bioinformatics*, 7(3):135, 2010.
17. T. Milenkovic and N. Przulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273, 2008.
18. T.P Mohamed, J.G Carbonell, and M.K Ganapathiraju. Active learning for human protein-protein interaction prediction. *BMC Bioinformatics*, 11(Suppl 1):S57, 2010.
19. M. Muntean, H. Valean, I. Ileana, and C. Rotar. Improving classification with support vector machine for unbalanced data. *Proceedings of 2010 IEEE International Conference on Automation, Quality and Testing, Robotics, THETA 17th edition*, pages 234–239, 2010.
20. Y. Park and E.M. Marcotte. Revisiting the negative example sampling problem for predicting protein-protein interactions. *Bioinformatics*, 27(21):3024–3028, 2011.
21. N. Przulj, D. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric?. *Bioinformatics*, 20(18):3508–3515, 2004.
22. Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. Random forest similarity for protein-protein interaction prediction from multiple sources. In Russ B. Altman, Tiffany A. Jung, Teri E. Klein, A. Keith Dunker, and Lawrence Hunter, editors, *Pacific Symposium on Biocomputing*. World Scientific, 2005.

23. Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. A mixture of feature experts approach for protein-protein interaction prediction. *BMC Bioinformatics*, 8(Suppl 10):S6, 2007.
24. Y. Qi, O. Tastan, J.G Carbonell, J. Klein-Seetharaman, and J. Weston. Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins. *Bioinformatics*, 26(18):i645–i652, 2010.
25. A. Sarajlic, V. Janjic, N. Stojkovic, D. Radak, and N. Przulj. Network topology reveals key cardiovascular disease genes. *PLoS ONE*, 8(8):e71537, 08 2013.
26. M.G Shi, J.F Xia, X.L Li, and D.S Huang. Predicting protein-protein interactions from sequence using correlation coefficient and high-quality interaction dataset. *Amino Acids*, 38(3):891–899, 2010.
27. E. Sprinzak, Y. Altuvia, and H. Margalit. Characterization and prediction of proteinprotein interactions within and between complexes. *PNAS*, 103(40):1471814723, 2006.
28. R. Tanaka, T.M Yi, and J. Doyle. Some protein interaction data do not exhibit power law statistics. *FEBS Letters*, 579:5140–5144, 2005.
29. O. Tastan, Y. Qi, J.G Carbonell, and J. Klein-Seetharaman. Prediction of interactions between hiv-1 and human proteins by information integration. *Proceedings of the Pacific Symposium on Biocomputing*, 14:516–527, 2009.
30. 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(6887):399–403, 2002.
31. Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(1):363–370, 2004.
32. J. Yu, M. Guo, C.J Needham, Y. Huang, L. Cai, and D. Westhead. Simple sequence-based kernels do not predict protein-protein interactions. *Bioinformatics*, 26(20):2610–2614, 2010.
33. L.V Zhang, S. Wong, O. King, and F. Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, 5:38, 2004.