Protein-Protein Interaction Prediction via Structured Matrix Completion

Qingming Tang *
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, Illinois 60637
qmtang@ttic.edu

Lifu Tu *
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, IL
lifu@ttic.edu

Aly Azeem Khan
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, IL
aakhan@ttic.edu

Jinbo Xu †
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, Illinois 60637
jinbo.xu@gmail.com

ABSTRACT

This paper considers how to computationally predict unknown protein-protein interactions (PPIs) given the experimentally verified PPIs. Matrix completion, a very popular machine learning technique that can be used to to infer the missing part of a matrix, has been introduced to recover the missing interactions of an incomplete PPI network. The benefit of Matrix completion is that it does not rely on unavailable negative samples, which are crucial for existing supervised classification methods. However, current matrix completion solutions for recovering missing PPIs fail to capture the important topology information of the underlying network. That is, the underlying network is a sparse network with skewed degree distribution.

In this paper, we design a structured matrix completion method that is suitable for capturing the skewed degree distribution of the underlying true PPI network. Theoretical analysis and extensive experimental results on known PPIs of three species (Plasmodium falciparum, Rattus norvegicus and Caenorhabditis elegans) show that our method outperforms related state-of-the-art protein-protein prediction approaches. We also tested the predicted networks of Plasmodium falciparum in terms of GO similarities. In turns out that our predicted network is with the highest GO score. To further demonstrate the power of our algorithm in predicting new PPIs, we compare the predicted Rattus norvegicus PPI networks using relatively old release from BioGrid with the much newer releases. Comparing with other methods, our predicted networks are much more similar with the newer releases. Our code is available upon request and will be public available in our website after the paper published.

1. INTRODUCTION

Proteins are the fundamental units of life, which play significant role in cellular processes such as composing cellular structure and promoting chemical reactions. The multiplicity of functions that one protein plays in most cellular processes and biochemical event are attributed to its interaction with other proteins. Thus, understanding protein-protein interactions (PPIs) is critical in both scientific research and practical applications such as identifying pharmacological targets and drug design. Though advanced high-throughput technologies [20, 38] are applied, the number of discovered PPIs is still very limited and far from complete [33, 34]. Further, reported by most experimental techniques, there exist experimental bias toward certain protein types and cellular localizations. Thus relaible computational methods to predict putative PPIs is helpful.

Supervised machine learning methods, such as random forest [8, 30], Bayesian network [17] and support vector machine (SVM) with different kernels [3, 12], have been widely employed to infer missing PPIs based on known interactions. These methods utilize biological features, e.g., PSSM [19] computed from protein sequences to train the classifier to distinguish the positive interacted pairs from the negative non-interacted pairs. However, all these supervised methods suffer from two problems: One is how to choose the negative training samples. Choosing negative samples is not easy because it is very hard to say that two proteins do not interact unless we have very strong evidence. That is, we cannot conclude that any pair of proteins that do not exist in the observed set, which comprise the positive samples, do not interact in real biological processes. Even if we can construct negative samples that can represent the non-interacting protein pairs, the training samples would still be unbalanced. Because the number of interacting protein pairs are much smaller than the number of non-interacting pairs. In machine learning community, it is widely known that the unbalanced

[1]Equal Contribution.
[2]Correspondence should be addressed to Dr. Xu.

training samples will cause the trained classifiers biased [28]. Matrix completion [6, 7, 21] refers to inferring the missing values of entries of one matrix based on the observed entries. A bunch of exiting approaches such as matrix factorization [23, 11, 26] and low rank/norm approximation [5, 16, 37] can be used to complete the matrix. Recently, matrix completion has been employed to protein-protein interaction task [40, 9, 41, 29]. Treat the target PPI network as a symmetric matrix, and fill the value of the ij-th entry as 1 if and only if protein i and j is observed to interact with each other. Otherwise, fill an entry with a question mark. In such scenario, predicting unknown PPIs can be modeled as guessing the values of those question marks. Then the PPI prediction is transformed to the matrix completion problem. After using matrix completion algorithm to predict the missing entries in the whole matrix, we can further choose the inferred entries with top K largest scores as true PPIs. Compared with those classification based approaches, matrix completion methods does not require negative samples. Among existing matrix completion based PPI prediction approaches, [40] shows state-of-the-art performance in terms of prediction accuracy by using non-negative tri-factorization [11]. However, existing matrix completion based prediction methods ignore the important topology information that the underlying PPI network is not a "random" graph. That is, the degree (number of interacting proteins) distribution of a real PPI network is significantly skewed. There are a bunch of papers discussing the degree property [13, 32, 2, 25, 18, 1, 39] of PPI network. Though it is still debatable about the "scale-free" property of PPI network, but the skewness of degree distribution is for sure, as shown in Fig 1, which describes the degree distribution based on latest version of verified PPIs of Plasmodium falciparum, Rattus norvegicus and Caenorhabditis elegans from BioGrid respectively.

Experimentally verified PPIs, a sample of the complete PPIs, convey the degree information of the complete PPI network only if the size of known PPIs are not too small . We define the observation rate of one protein as the number of experimentally verified interacting proteins divided by the number of true interacting proteins. If we have already known the observation rate of each protein, we can use this information to greatly improve the prediction accuracy as we can confidently control the number of predicted interacting proteins of each protein. We design a structured matrix completion framework which would iteratively and simultaneously learn the observation rate of each protein and the value of each question mark. In each iteration, we are solving a structured matrix factorization problem which would enforce the predicted PPI network consistent with the estimated observation rates. The rates reflect our guess of the degree distribution. In turn, the predicted PPI network, supervised by the degree information, would further effect our estimation of the observation rate. The following part of the paper is organized as follows. In section two, we introduce our mathematical model and describe the optimization process for our model. In section three, we would list the data sources we used throughout the paper. In section four, we would compare our method with existing matrix completion solutions in different settings. We demonstrate that our method significantly outperforms existing solutions in prediction accuracy. We also show the power of our method in predicting new PPIs by comparing the predicted PPIs based on old release of PPI network with much newer and larger PPI networks. Finally, we conclude the paper in section five.

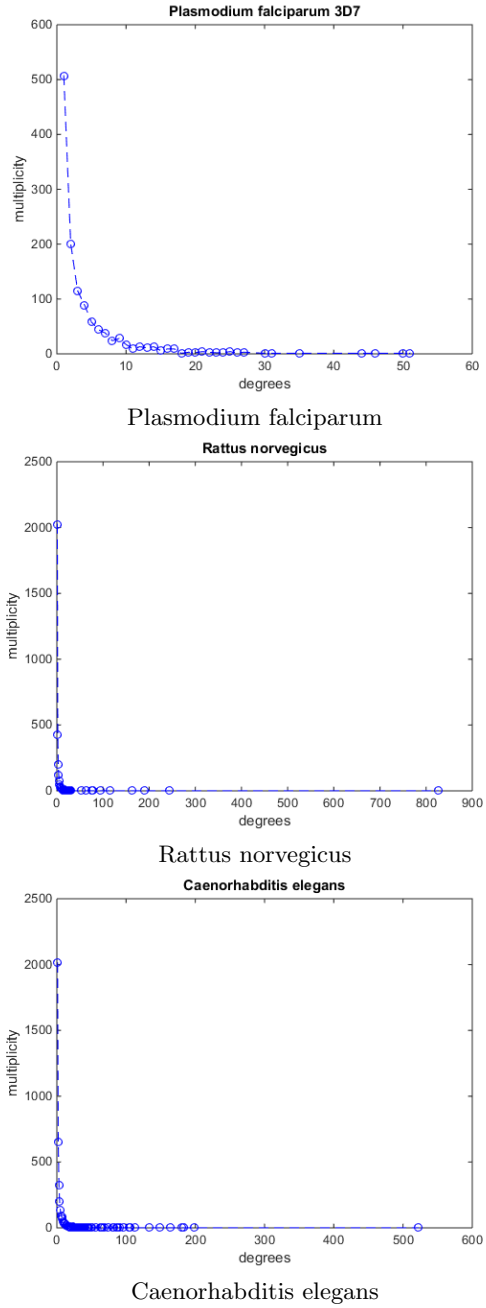


Figure 1: Degree distribution of three species

2. METHODS

We first briefly formalize the PPI prediction problem. Given a PPI network consists of p proteins, we can represent the network as $G = (V, \Omega, \mathbf{E})$, where $V = \{1, 2, ..., p\}$ is the

vertex set of the network corresponds to p proteins and $\mathbf{E}$ is the adjacency matrix we need to complete. There are some already observed 1s in $\mathbf{E}$ while all the other entries in $\mathbf{E}$ are left as question marks. These observed entries form the set Ω.

2.1 Non-Negative matrix completion

Non-negative matrix factorization (NMF) [23] and its extension Tri-factorization (NMTF) [11] have been introduced to predict those question marks. Shortly, the procedure is to iteratively apply matrix factorization to current estimation $\mathbf{E}^{(i)}$, and then use the product of the estimated factors of $\mathbf{E}^{(i)}$ as the $(i+1)$-th estimation $\mathbf{E}^{(i+1)}$ (See [40] for details). For the i-th iteration, the objective function can be written as follows (when using NMTF)

$$\min_{\mathbf{H}\geq 0,\mathbf{S}\geq 0} |\mathbf{E}^{(\mathbf{i})} - \mathbf{HSH}'|^2_\Omega \tag{1}$$

Here, $|^2_\Omega$ means the Frobenius norm constrained on Ω.

2.2 Modeling degree distribution given observation rate

As shown in Fig 1, the degree distribution of observed PPIs of the three species (Plasmodium falciparum, Rattus norvegicus and Caenorhabditis elegans) are very skewed. Most PPIs centralize around just a small portion of the proteins. We can infer that the degree distribution of the complete PPI network, though not identically the same with the network of the observed PPIs, is also very skewed. However, NMF and NMTF are not specifically designed for such kind of skewed data. That is, the degree distribution information is not used for optimizing (17). Though NMTF has shown to be related to k-means clustering [11], NMTF can not quantitatively characterize how many PPIs are expected to be recovered around each protein, which is crucial for recovering the missing PPIs in such a skewed scenario. In this section, we would discuss how to build a prior to enforce the predicted PPI network to follow the expected degree distribution under some assumptions.

Denote the observation rate of the p proteins $(V_1, V_2, ..., V_p)$ as $(o_1, o_2, ..., o_p)$, in other words, o_i equals to the number of proteins been observed to interact with protein V_i (except protein V_i itself) divided by the number of all proteins that truly interact with protein V_i. Typically, if we have known the actual value of observation rate of each protein, then we can specifically control the number of interacting proteins to be reported in the final prediction, which would enhance the prediction accuracy.

2.2.1 Expected degree distribution inducing prior

There are some existing papers working on designing a prior/norm to induce a scale-free (or hub) graph in different scenarios [24, 10, 27]. There are two difficulties to directly apply these methods. One is that the underlying PPI may not be a perfect scale-free network. Another is, all these methods are trying to pose a global constraint, which is one property of scale-free network, to regularize the main objective function rather than specifically controlling the degree of each node. Actually, global constrain is always not an ideal choice to enforce a specific distribution because there are many possible distributions that favor same global constraint [36]. But how can we specifically control the degree of each protein?

Now assuming that we are describing the p nodes network $\mathbf{X}$ (p dimension symmetric matrix), the observation rate has been given by $(o_1, o_2, ..., o_p)$, the number of observed interactions of each node are given by $(q_1, q_2, ..., q_p)$, then we design the following prior for regularizing the matrix factorization problem like (1),

$$S_a(\mathbf{X}, o, q) = \sum_{i=1}^{p} \frac{a[1]X_{i,[1]} + a[2]X_{i,[2]} + ... + a[p-1]X_{i,[p-1]}}{a[\lceil \frac{q_i}{o_i} \rceil]} \tag{2}$$

Here, $a[.]$ is a monotonically increasing positive discrete function, that is

$$0 < a[1] < a[2] < ... < a[p-1] \tag{3}$$

There are a bunch of choices for $a[.]$ only if it increases with a moderate rate. Through out the paper, we would use the mapping $a[i] = \log(i+1)^{0.15}$ for $1 \leq i < p$. $X_{i,[j]}$ denotes the j-th strongest interaction pair around node i, that is, the j-th largest value among $X_{i,k}$ for $1 \leq k \leq p$ and $k \neq j$. When the degree information $d = \{d_1, d_2, ..., d_p\}$ is given rather than the ratio and observation, we directly written (6) as

$$S_a(\mathbf{X}, d) = \sum_{i=1}^{p} \frac{a[1]X_{i,[1]} + a[2]X_{i,[2]} + ... + a[p-1]X_{i,[p-1]}}{a[d_i]} \tag{4}$$

There are a bunch of properties of Eq. 2, as follows

1 For each node i, there are exactly $\lceil \frac{q_i}{o_i} \rceil$ number of putative interactions with penalty smaller than or equal to 1.

2 For each node i, the stronger putative interactions are penalized with smaller penalty

3 If $\lceil \frac{q_i}{o_i} \rceil$ is larger than $\lceil \frac{q_j}{o_j} \rceil$, then the k-th largest pair around i is assigned with smaller penalty than the k-th largest pair around j.

Based on the three properties, we have the following **Theorem** 1

THEOREM 1. *Given degree distribution $\{d_1, d_2, ..., d_p\}$ of p variables and a network $\mathbf{X}$ satisfying $\sum_{i=1}^{p} |X_{-i}|_0 = \sum_{i=1}^{p} d_i$ but does not strictly following the degree distribution, where $X_{-i} = \{X_{i,1}, ..., X_{i,i-1}, X_{i,i+1}, X_{i,p}\}$, there exists another network $\mathbf{X}^*$ with the same sparsity level, such that $S_a(\mathbf{X}^*, d) < S_a(\mathbf{X}, d)$.*

The detailed proof of (1) can be found in supplementary material. **Theorem** 1 shows that Eq 2 is a structured sparsity inducing prior which favors the network follow the distribution defined by given degree distribution, or o and q. When it is used as a prior to regularize the main objective function, like matrix factorization in our matrix completion problem, it would also drive the solution of matrix factorization follows the degree distribution defined by $(o_1, o_2, ..., o_p)$ and $(q_1, q_2, ..., q_p)$.

2.2.2 Matrix factorization with expected degree distribution inducing prior

We now consider how to solve (1) if the observation rate $(o_1, o_2, ..., o_p)$ is given. To apply the expected degree distribution inducing prior $S_a(\mathbf{X}, o, q)$ to a matrix factorization problem like (1), the objective function is modified as

$$|\mathbf{HSH}' - \mathbf{E}|^2_\Omega + \alpha S_a(\mathbf{HSH}', o, q) \tag{5}$$

Where α is a parameter controlling the sparsity level. Solving Eq 5 is not easy, we can relax the optimization problem of minimizing Eq 5 as the following problem

$$\begin{aligned} &|\mathbf{HSH}' - \mathbf{E}|^2_\Omega + \alpha S_a(\mathbf{X}, o, q) \\ &s.t. \quad \mathbf{HSH}' = \mathbf{X} \end{aligned} \tag{6}$$

Which can be solved by applying ADMM [14, 4], a widely used decomposition optimization framework. Using ADMM, solving Eq. 6 can be summarized as iteratively doing the following three steps till convergence, where ρ is a positive number.

$$\begin{aligned} &\mathbf{H}^{(t+1)}, \mathbf{S}^{(t+1)} \\ &= \min_{\mathbf{H}\geq 0, \mathbf{S}\geq 0} |\mathbf{HSH}' - \mathbf{E}|^2_\Omega + \frac{\rho}{2}|\mathbf{HSH}' - \mathbf{X}^{(t)} + \mathbf{U}^{(t)}|^2_F \end{aligned} \tag{7}$$

$$\begin{aligned} &\mathbf{X}^{(t+1)} \\ &= \min_{\mathbf{X}\geq 0} \frac{\rho}{2}|\mathbf{H}^{(t+1)}\mathbf{S}^{(t+1)}\mathbf{H}^{(t+1)\prime} - \mathbf{X} + \mathbf{U}^{(t)}|^2_F + \alpha S_a(\mathbf{X}, o, q) \end{aligned} \tag{8}$$

$$\mathbf{U}^{(t+1)} = \mathbf{U}^{(t)} + \mathbf{H}^{(t+1)}\mathbf{S}^{(t+1)}\mathbf{H}^{(t+1)\prime} - \mathbf{X}^{(t+1)} \tag{9}$$

Solve Eq 7

Denote $\mathbf{X}^{(t)} - \mathbf{U}^{(t)}$ as $\mathbf{Z}^{(t)}$, by mathematically combining the similar terms of Eq 7, solving 7 is equivalent to minimize the following problem

$$\begin{aligned} &\sum_{ij\in\Omega} (\mathbf{HSH}'_{ij} - \mathbf{E}_{ij})^2 + \frac{\rho}{2} \sum_{1\leq i,j\leq p} (\mathbf{HSH}'_{ij} - \mathbf{Z}^{(t)}_{ij})^2 \\ &= \sum_{ij} (\mathbf{1}_{ij\in\Omega} + \frac{\rho}{2})(\mathbf{HSH}'_{ij})^2 \\ &- \sum_{ij} (2\mathbf{E}_{ij}\mathbf{1}_{ij\in\Omega} + \rho\mathbf{Z}^{(t)}_{ij})(\mathbf{HSH}'_{ij}) + const \end{aligned} \tag{10}$$

Minimizing (10) is equivalent to solving the following problem

$$\min_{\mathbf{H}\geq 0, \mathbf{S}\geq 0} |\mathbf{HSH}' - \mathbf{K}|^2_F \tag{11}$$

Where $[K_{ij}] = [\frac{\mathbf{E}_{ij}\mathbf{1}_{ij\in\Omega} + \frac{\rho}{2}\mathbf{Z}^{(t)}_{ij}}{\mathbf{1}_{ij\in\Omega} + \frac{\rho}{2}}]$.

Solve Eq 8

The intuition to solve Eq 8 is to divide the problem into p smaller problems as follows, and solve each of the problem separately.

$$\min_{\mathbf{X}_i\geq 0} \big(\frac{\rho}{2}|\mathbf{X}_i - \mathbf{A}_i|^2_F + \alpha \sum_{k=1}^{p-1} b_i(k) X_{i,[k]}\big) \tag{12}$$

Where $\mathbf{A} = \mathbf{H}^{(t+1)}\mathbf{S}^{(t+1)}\mathbf{H}^{(t+1)\prime} + \mathbf{U}^{(t)}$, $\mathbf{X}_i$ and $\mathbf{A}_i$ are the i-th column of corresponding matrices. $b_i(.)$ is a new mapping defined as $b_i(k) = \frac{a(k)}{a[\lceil \frac{q_i}{o_i} \rceil]}$ for $1 \leq i < p$. However, the p smaller problems like Eq 12 are not independent as $\mathbf{X}$ is a symmetric matrix. To resolve this issue, we can borrow the idea from [10] to relax the symmetric constraint as follows,

$$\begin{aligned} &\min_{\mathbf{X}\geq 0} \frac{\rho}{2}|\mathbf{X} - \mathbf{A}|^2_F + \alpha S_a(\mathbf{X}, o, q) \\ &s.t. \mathbf{X} = \mathbf{X}^T \end{aligned} \tag{13}$$

Then we can apply dual decomposition [22], another common framework for solving complicated optimization problem, to solve Eq 13. By introducing the Lagrangian term $< \beta, (\mathbf{X} - \mathbf{X}^T) >$ (point-wise product of two matrices), we are actually minimizing the following objective function in each iteration of dual decomposition.

$$\min_{\mathbf{X}\geq 0} \frac{\rho}{2}|\mathbf{X} - \mathbf{A}|^2_F + \alpha S_a(\mathbf{X}, o, q) + < (\beta - \beta^T), \mathbf{X} > \tag{14}$$

Here, β is a p by p matrix, and its entries will be adjusted according to the difference between $\mathbf{X}$ and $\mathbf{X}^T$ accordingly as dual decomposition goes on. Actual, by completing the square terms, Eq 14 can be transformed to

$$\min_{\mathbf{X}\geq 0} \frac{\rho}{2}|\mathbf{X} - \mathbf{B}|^2_F + \alpha S_a(\mathbf{X}, o, q) + const \tag{15}$$

Which can be decomposed as p independent problems like Eq 12. In the paper [36], an interesting $O(p\log(p))$ algorithm is presented to solve a problem like Eq 12.

2.3 Update observation rate

We have presented how to solve a matrix factorization problem regularized by degree constrained prior in last section when observation rates are known. However, observation rates of proteins, which are latent variables, are unknown. We propose an iterative method to simultaneously predict the observation rates and the PPI network.

As we know, a typical method to evaluate a prediction method is to look the AUC (or other similar metric) curve. Assume that we would like to output the top K entries as our prediction. In this situation, we can assume the number of true edges are K, and we can initially set the observation rate as $o^{(0)} = (o^{(0)}_1, o^{(0)}_2, ..., o^{(0)}_p) = (\frac{|q|}{K}, \frac{|q|}{K}, ..., \frac{|q|}{K})$. $q = (q_1, q_2, ...q_p)$ is the vector representing the number of observations of the each protein.

Though this initial guess will not be true even if under the assumption of random sampling, we can update the observation rate based on current estimation of the PPI network. That is, $o^{(1)}$ can be updated based on $\mathbf{H}^{(1)}\mathbf{S}^{(1)}\mathbf{H}^{(1)\prime}$ predicted from Eq 16

$$\mathbf{H}^{(1)}, \mathbf{S}^{(1)} = \min |\mathbf{HSH}' - \mathbf{E}|^2_\Omega + \alpha S_a(\mathbf{HSH}', o^{(0)}, q) \tag{16}$$

Given $\mathbf{H}^{(1)}\mathbf{S}^{(1)}\mathbf{H}^{(1)\prime}$, we can rank all the putative interactions (there are $\frac{p(p-1)}{2}$ interactions), and select top K putative interactions. Denote $(d^{(1)}_1, d^{(1)}_2, ..., d^{(1)}_p)$ as the distribution of the K interactions over p nodes, then we can update the observation rate as

$$o^{(1)} = (o^{(1)}_1, o^{(1)}_2, ..., o^{(1)}_p) = (\frac{q_1}{d^{(1)}_1}, \frac{q_2}{d^{(1)}_2}, ..., \frac{q_p}{d^{(1)}_p}) \tag{17}$$

The update for both the observation rates and the PPI network would continue until convergence. The intuition of the iterative algorithm is similar to EM algorithm. Where we don't know both the observation rate (which reflects the degree distribution) and the real PPI network, but each information can improve the prediction of each other. Thus

we alternatively guess each of the information, and then stop when the two guesses agree with each other.

2.4 Reweighed model and algorithm framework

We describe the whole algorithm in Algorithm 1. There are two points we need to pay attention to.

1 Traditional matrix factorization method can not be directly applied to do matrix completion because those methods are based on Frobenius norm rather than Ω. Thus, we set the initial target matrix as $\mathbf{E}^{(0)}$, whose question marks are 0, and then iteratively update the value of these question marks as shown in the algorithm.

2 Unlike previous solutions [40, 9, 41, 29] for PPI prediction that based on matrix completion, we would adjust the weight of of the observations ($(i,j) \in \Omega$) if they temporally rank out of top K. There are two explanations. One is the estimator should be a consistent one. If the observations are out of top K, then the estimator itself is not consistent. The second reason is that the real PPI is very sparse, so when we look at the top K entries, where K is typically a small number comparing with $\frac{p(p-1)}{2}$, the observations should be in top K.

Data: $\mathbf{E}$,$\mathbf{\Omega}$,$q = (q_1, q_2, ..., q_p)$,K
Result: Predict missing values of $\mathbf{E}$ and $o = (o_1, o_2, ...o_p)$
Initialize $\mathbf{E}^{(0)} = \mathbf{E} \times \mathbf{1}_{ij \in \Omega}$, $converge = false$, $t = 0$,
$o_i^{(0)} = \frac{|q|_1}{K}$ for $1 \leq i \leq p$;
while *converge is false* **do**
  $\mathbf{H}^{(t+1)}, \mathbf{S}^{(t+1)} =$
  $\min_{\mathbf{H}\geq 0, \mathbf{S}\geq 0} \left(|\mathbf{HSH}' - \mathbf{E}^{(t)}|_\Omega^2 + \alpha S_a(\mathbf{HSH}', o^{(t)}, q) \right)$;
  Select top K entries from $\mathbf{H}^{(t+1)}\mathbf{S}^{(t+1)}\mathbf{H}^{(t+1)\prime}$;
  Update $o^{(t+1)}$;
  if $|\mathbf{S}^{(t+1)} - \mathbf{S}^{(t)}|_1 < 10^{-3}$ *and* $|\mathbf{H}^{(t+1)} - \mathbf{H}^{(t)}|_1 < 10^{-3}$ **then**
    $converge = true$;
  else
    $\mathbf{E}^{(t+1)} = \mathbf{H}^{(t+1)}\mathbf{S}^{(t+1)}\mathbf{H}^{(t+1)\prime}$;
  end
  for $(i,j) \in \Omega$ **do**
    if $\mathbf{E}_{i,j}^{(t+1)}$ *ranks out of top* K **then**
      $\mathbf{E}_{i,j}^{(t+1)} = \mathbf{E}_{i,j}^{t+1} \times 1.1$;
    end
  end
  $t = t + 1$;
end

Algorithm 1: Iterative Structured Matrix Completion

3. MATERIALS AND DATA SOURCES

We construct our PPI networks based on the verified PPIs compiled by BioGRID databse [35]. We evaluate our methods with related matrix completion solutions via comparing predicted PPIs based on release 3.3.124 of three species Plasmodium falciparum, Rattus norvegicus and Caenorhabditis elegans respectively. In this release, there are 2541 verified PPIs for Plasmodium falciparum, 5030 verified PPIs for Rattus norvegicus and 8725 verified PPIs for Caenorhabditis elegans respectively.

To demonstrate the power of our algorithm in predicting unknown PPIs, we download five releases (3.3.124, 3.2.114, 3.2.104, 3.1.94 and 3.1.84) of Rattus norvegicus, and compare the predicted network using one of the first four releases with its successive releases. The detailed information of each release of Rattus norvegicus are summarized in **Table** 1.

Table 1: Information about six releases of Rattus norvegicus

Release	3.1.74	3.1.84	3.1.94	3.2.104	3.2.114	3.3.124
# Edges	?	969	2322	4250	4599	5051

4. EXPERIMENTAL RESULTS AND DISCUSSION

We implemented two methods, one is structured matrix completion based on NMTF (Denoted as **SNMTFC** for short) and another is structured matrix completion based on NMF (Denoted as **SNMFC** for short). As [40, 9, 41, 29] have claimed nice performance (in prediction accuracy) of matrix completion solutions compared to existing supervised classification methods and kernel based methods [31, 3, 15], we just compare with **NMFC** (matrix completion based on NMF) and **NMTFC** (matrix completion based on NMTF) in this paper.

There are three parts of this section. In the first part, we would compare the prediction accuracy of our algorithms with **NMFC** and **NMTFC** respectively, given partial observations of known PPIs of the three species listed in Section 3. In the second part, we would test our algorithms using different releases of PPI network of Rattus norvegicus to show the capability of our algorithm in predicting new PPIs. That is, using older releases as the partial observations of much newer releases, and see if our algorithm can better assisting scientists to find out useful PPIs. In the third part, we would calculate the GO similarities of the predicted Plasmodium falciparum PPI networks using latest version of known PPIs of the four algorithms. We would show that our algorithms also outperform **NMFC** and **NMTFC** in the view of function analysis.

4.1 Improved prediction accuracy

In this section, we compare our methods with **NMFC** and **NMTFC** in two different observation levels. Say, using 30% and 10% of the known PPIs as training set respectively, and test over the remaining 70% and 90% known PPIs accordingly. We compare the four methods by looking at how many remaining known PPIs are recovered when predicting top $0.1\% \times \frac{p(p-1)}{2}$, $0.15\% \times \frac{p(p-1)}{2}$, $0.2\% \times \frac{p(p-1)}{2}$, $0.3\% \times \frac{p(p-1)}{2}$, $0.4\% \times \frac{p(p-1)}{2}$ and $0.5\% \times \frac{p(p-1)}{2}$ edges respectively.

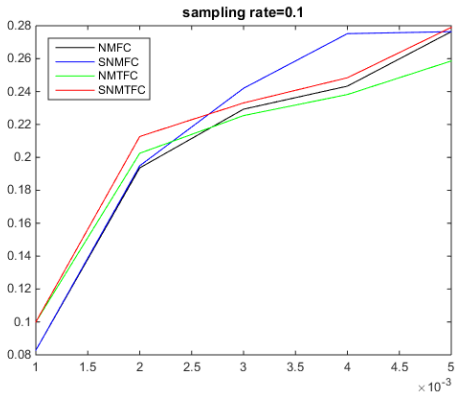
In order to achieve best performance, we tune the parameter k ($k \leq 50$) for **NMFC** and **NMTFC** in each data set respectively, such that both algorithms achieve best performance in the each data set. This can be done by cross-validation as shown in [40]. Here, k is the dimension of factors, which is crucial for matrix factorization. We use the

same k for **SNMFC** and **NMFC**, and the same k for **SNMTFC** and **NMTFC**. Beside k, the number of iterations used to complete a matrix is also a crucial hyper parameter. Ideally, we would expect all methods would converge. Practically, considering the efficiency issue, we set maximum number of iterations to 10 for all the four methods. If one algorithm does not converge in 10 iterations, we just pick up the iteration that achieves best performance for comparison. k and maximum number of iterations are common parameters for all the four algorithms. There is another parameter α for **SNMFC** and **SNMTFC** specifically. Obviously, α can also be tuned by using cross validation. However, we do not strictly tune α, and just set $\alpha = 0.001$ for all the test cases.

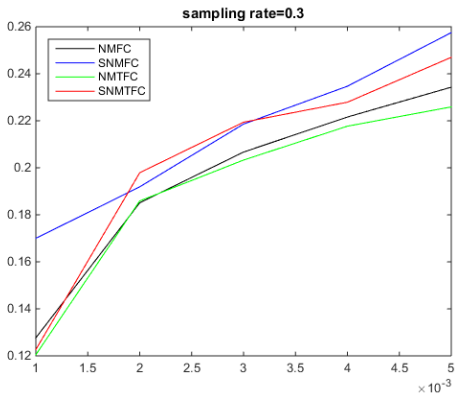
For the three species Plasmodium falciparum, Rattus norvegicus and Caenorhabditis elegans, we repeat the experiments in each setting (one specific observation level and one specific recall) five times (that is, we randomly generate the training set at 30% and 10% level for every recall five times), and take average of the results for every setting. In **Figure** 2, **Figure** 3 and **Figure** 4, the y-axis denotes the ratio of remaining known PPIs been predicted while x-axis denotes the ratio of PPIs been reported, that is the reported number of PPIs divided by $\frac{p(p-1)}{2}$. According to the figures, the **SNMFC** performs much better than **NMFC** and **SNMTFC** also outperforms **NMTFC** in terms of prediction accuracy under both observation levels. This fact demonstrates the powerful effect of our structured sparsity inducing prior.

4.2 Capability to predict new protein interactions

To demonstrate the power of our algorithm in predicting new PPIs, we can predict putative PPIs based on some old release of PPI network, and compare the predicted network with recent larger and experimentally verified PPI networks. We predict PPIs of Rattus norvegicus based on the released PPI network of version 3.2.114, 3.2.104, 3.1.94, 3.1.84 and 3.1.74 respectively using all the four methods. Denote the PPIs of the six releases (3.1.74, 3.1.84, 3.1.94, 3.2.104, 3.2.114 and 3.3.124) as P_1, P_2, P_3, P_4, P_5 and P_6 respectively, and $|X|$ as the number of edges in the network X. We evaluate the predicted network by how many edges ranked in top $|P_i| - |P_j|$ are in P_i, where $1 \leq j < i \leq 6$. According to the **Tables** 2, 3, 4 and 5, our methods are much powerful in predicting new PPIs comparing with **NMFC** and **NMTFC**.

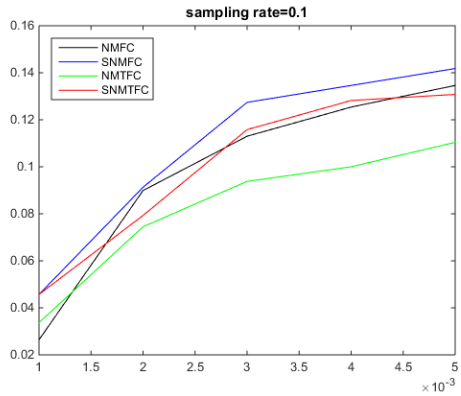


Correctly predicted vs # predicted with 90% PPIs as training

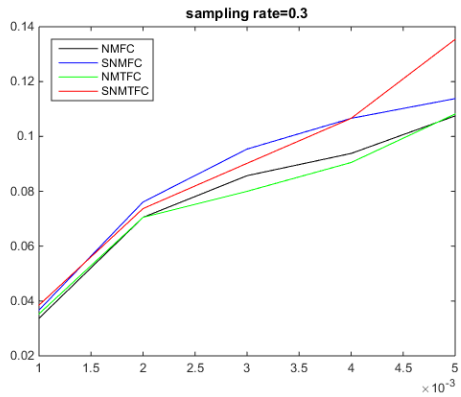


Correctly predicted vs # predicted with 70% PPIs as training

Figure 2: Caenorhabditis Elegans Comparison

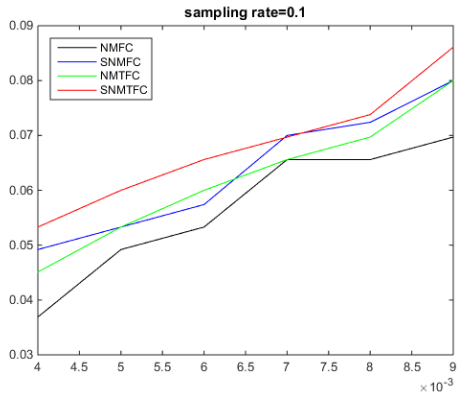


Correctly predicted vs # predicted with 90% PPIs as training

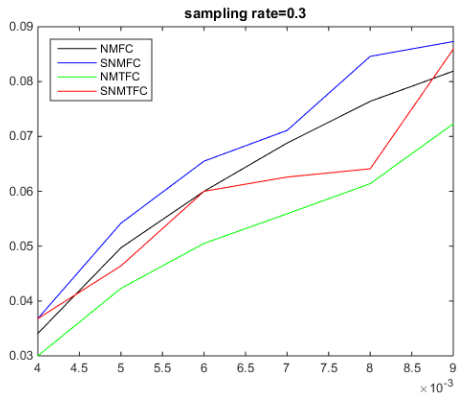


Correctly predicted vs # predicted with 70% PPIs as training

Figure 3: Rattus Norvegicus Comparison



Correctly predicted vs # predicted with 90% PPIs as training



Correctly predicted vs # predicted with 70% PPIs as training

Figure 4: Plasmodium falciparum Comparison

Table 2: Results when considering top $0.003 \times \frac{p(p-1)}{2}$ edges. Each entry means Accuracy of NMTFC vs Accuracy of SNMTFC

-	3.1.84	3.1.94	3.2.104	3.2.114	3.3.124
3.174	0/0	0/0	0/0	0/0	0/0
3.1.84	-	0.0109/0.0326	0.0423/0.0508	0.0399/0.450	0.0378/0.0415
3.1.94	-	-	0.0388/0.1025	0.0350/0.0911	0.0349/0.0893
3.2.104	-	-	-	0.0674/0.1124	0.0515/0.0735
3.2.114	-	-	-	-	0.0345/0.0345

Table 3: Results when considering top $0.004 \times \frac{p(p-1)}{2}$ edges. Each entry means Accuracy of NMTFC vs Accuracy of SNMTFC

-	3.1.84	3.1.94	3.2.104	3.2.114	3.3.124
3.174	0/0	0.0147/0.0588	0.0079/0.0394	0.0074/0.0370	0.0070/0.0352
3.1.84	-	0.0217/0.0652	0.0538/0.0538	0.0507/0.0507	0.0481/0.0481
3.1.94	-	-	0.0831/0.2271	0.0724/0.1986	0.0719/0.1874
3.2.104	-	-	-	0.0899/0.0899	0.0735/0.0735
3.2.114	-	-	-	-	0.0345/0.0345

Table 4: Results when considering top $0.005 \times \frac{p(p-1)}{2}$ edges. Each entry means Accuracy of NMTFC vs Accuracy of SNMTFC

-	3.1.84	3.1.94	3.2.104	3.2.114	3.3.124
3.174	0/0.05	0.0294/0.0588	0.0315/0.0394	0.0296/0.0370	.0282/0.0352
3.1.84	-	0.0326/0.1196	0.071/0.1115	0.0688/0.1087	0.0653/0.11
3.1.94	-	-	0.0997/0.15245	0.0864/0.1332	0.085/0.1373
3.2.104	-	-	-	0.0899/0.1348	0.0735/0.1176
3.2.114	-	-	-	-	0.0517/0.0690

Table 5: Results when considering top $0.01 \times \frac{p(p-1)}{2}$ edges. Each entry means Accuracy of NMTFC vs Accuracy of SNMTFC

-	3.1.84	3.1.94	3.2.104	3.2.114	3.3.124
3.174	0.1/0.1	0.0588/0.0735	0.0551/0.0551	0.0519/0.0519	0.0493/0.0493
3.1.84	-	0.0435/0.0652	0.0385/0.0538	0.0362/0.0507	0.0362/0.0507
3.1.94	-	-	0.097/0.1684	0.0911/0.1422	0.0937/0.1512
3.2.104	-	-	-	0.1798/0.2247	0.1618/0.1912
3.2.114	-	-	-	-	0.1552/0.1724

4.3 GO similarities and putative unknown P-PIs

We examine the predicted networks of the three species (using latest release).

Fill in comments about predicted putative PPIs here.

5. CONCLUSION

PPI network is kind of biological network with skewed degree distribution. We propose a structured sparsity inducing prior to capture the degree distribution information of the underlying network by enforcing different level of sparsity for each node. Based on this degree prior, we design two structured matrix completion algorithms based on **NMF** and **NMTF**. Our algorithms simultaneously estimate the missing interactions and the degree distribution of the proteins to enhance prediction accuracy. The degree may not be interesting for a random graph with uniformed degree distribution, but it does convey valuable information for a graph with skewed distribution.

We tested our algorithms using complied datasets from BioGrid. We demonstrate the superior performance of algorithms comparing with others through both simulation and function analysis. More interestingly, we compared the predicted networks of our algorithms using old release of PPIs with much recent release of PPIs. This experiment does confidently show the ability of our algorithm in predicting real PPIs. It turns out that our algorithm does outperform other matrix completion solutions in terms of the ability of predicting new PPIs.

6. ACKNOWLEDGMENTS

The work is supported by NSF/BIO DBI-1262603 and NSF/CCF AF-1149811.

7. REFERENCES

[1] R. Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

[2] A.-L. Barabási, Z. Dezső, E. Ravasz, S.-H. Yook, and Z. Oltvai. Scale-free and hierarchical structures in complex networks. In *Modeling of Complex Systems: Seventh Granada Lectures*, volume 661, pages 1–16. AIP Publishing, 2003.

[3] A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(suppl 1):i38–i46, 2005.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[7] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[8] X.-W. Chen and M. Liu. Prediction of protein–protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

[9] O. M. W. Dai and N. S. Prasad. Low-rank matrix completion for inference of protein-protein interaction networks. In *ICNAAM 2010: International Conference of Numerical Analysis and Applied Mathematics 2010*, volume 1281, pages 1531–1534. AIP Publishing, 2010.

[10] A. Defazio and T. S. Caetano. A convex formulation for learning scale-free networks via submodular relaxation. In *Advances in Neural Information Processing Systems*, pages 1250–1258, 2012.

[11] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[12] Y. Guo, L. Yu, Z. Wen, and M. Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic acids research*, 36(9):3025–3030, 2008.

[13] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell. Protein-protein interaction networks and biologyŮwhat's the connection? *Nature biotechnology*, 26(1):69–72, 2008.

[14] M. R. Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[15] M. Hue, M. Riffle, J.-P. Vert, and W. S. Noble. Large-scale prediction of protein-protein interactions from structures. *BMC bioinformatics*, 11(1):144, 2010.

[16] P. Jain, R. Meka, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

[17] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.

[18] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

[19] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

[20] J. K. Joung, E. I. Ramm, and C. O. Pabo. A bacterial two-hybrid selection system for studying protein–dna and protein–protein interactions. *Proceedings of the National Academy of Sciences*, 97(13):7382–7387, 2000.

[21] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[22] T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1288–1298. Association for Computational Linguistics, 2010.

[23] D. D. Lee and H. S. Seung. Algorithms for

non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[24] Q. Liu and A. T. Ihler. Learning scale free networks by reweighted l1 regularization. In *International Conference on Artificial Intelligence and Statistics*, pages 40–48, 2011.

[25] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002.

[26] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[27] K. Mohan, M. Chung, S. Han, D. Witten, S.-I. Lee, and M. Fazel. Structured learning of gaussian graphical models. In *Advances in neural information processing systems*, pages 620–628, 2012.

[28] Y. L. Murphey, H. Guo, and L. A. Feldkamp. Neural learning from unbalanced data. *Applied Intelligence*, 21(2):117–128, 2004.

[29] S. Pinkert, J. Schultz, and J. Reichardt. Protein interaction networksŮmore than mere modules. *PLoS Computational Biology*, 6(1):e1000659, 2010.

[30] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. Random forest similarity for protein-protein interaction prediction from multiple sources. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 531–542, 2004.

[31] J. Qiu, M. Hue, A. Ben-Hur, J.-P. Vert, and W. S. Noble. A structural alignment kernel for protein structures. *Bioinformatics*, 23(9):1090–1098, 2007.

[32] M. Salathé, R. M. May, and S. Bonhoeffer. The evolution of network topology by selective removal. *Journal of the Royal Society Interface*, 2(5):533–536, 2005.

[33] B. A. Shoemaker and A. R. Panchenko. Deciphering protein–protein interactions. part i. experimental techniques and databases. *PLoS computational biology*, 3(3):e42, 2007.

[34] B. A. Shoemaker and A. R. Panchenko. Deciphering protein–protein interactions. part ii. computational methods to predict protein and domain interaction partners. *PLoS computational biology*, 3(4):e43, 2007.

[35] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl 1):D535–D539, 2006.

[36] Q. Tang, S. Sun, and J. Xu. Learning networks by node specific degree prior. *arXiv preprint arXiv:1503.02129*, 2015.

[37] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640):15, 2010.

[38] G. Verbeck, B. Ruotolo, H. Sawyer, K. Gillig, and D. Russell. A fundamental introduction to ion mobility mass spectrometry applied to the analysis of biomolecules. *Journal of biomolecular techniques: JBT*, 13(2):56, 2002.

[39] 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.

[40] H. Wang, H. Huang, C. Ding, and F. Nie. Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization. *Journal of Computational Biology*, 20(4):344–358, 2013.

[41] Q. Xu, E. W. Xiang, and Q. Yang. Protein-protein interaction prediction via collective matrix factorization. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 62–67. IEEE, 2010.