Protein Function Prediction using Dependence Maximization

Guoxian Yu[1], Carlotta Domeniconi[2], Huzefa Rangwala[2], and Guoji Zhang[3]

[1] School of Comp. & Inf. Sci., Southwest University, Chongqing, China
[2] Department of Comp. Sci., George Mason University, Fairfax, VA, USA
[3] School of Sci., South China University of Technology, Guangzhou, China
guoxian85@gmail.com, {carlotta,rangwala}@cs.gmu.edu, magjzh@scut.edu.cn

Abstract. Protein function prediction is one of the fundamental tasks in the post genomic era. The vast amount of available proteomic data makes it possible to computationally annotate proteins. Most computational approaches predict protein functions by using the labeled proteins and assuming that the annotation of labeled proteins is complete, and without any missing functions. However, partially annotated proteins are common in real-world scenarios, that is a protein may have some confirmed functions, and whether it has other functions is unknown.

In this paper, we make use of partially annotated proteomic data, and propose an approach called *Pro*tein Function Prediction using *D*ependency *M*aximization (ProDM). ProDM works by leveraging the correlation between different function labels, the 'guilt by association' rule between proteins, and maximizes the dependency between function labels and feature expression of proteins. ProDM can replenish the missing functions of partially annotated proteins (a seldom studied problem), and can predict functions for completely unlabeled proteins using partially annotated ones. An empirical study on publicly available protein-protein interaction (PPI) networks shows that, when the number of missing functions is large, ProDM performs significantly better than other related methods with respect to various evaluation criteria.

1 Introduction

Proteins are macromolecules that serve as the fundamental building blocks and functional components of a living cell. The knowledge of protein functions can promote the development of new drugs, better crops and synthetic biochemicals [14]. With the development of high-throughput biotechnologies, it is easy to collect various proteomic data, but the functions of these proteomic data cannot be determined at the same pace. The availability of vast amount of proteomic data enables researchers to computationally annotate proteins. Thus various computational models have been developed to reduce the cost associated with experimentally annotating proteins in the wet lab.

Numerous computational approaches have been proposed for protein function prediction. Some approaches assume that two proteins with similar sequences should have similar functions. These methods use a kernel function (i.e., string kernel [12]) to measure the similarity between the sequences of a pair of proteins and predict their functions. A protein often interacts with other proteins to accomplish certain tasks. Some algorithms take advantage of this knowledge and use protein-protein interaction (PPI) networks to automatically make predictions [5,18,20,24]. Further, some approaches integrate multiple data types (i.e., PPI networks, protein sequences, and gene co-expression networks) for protein function prediction [13,21].

Proteins have multiple functions and each function can be viewed as a label. These function labels are typically correlated. Traditional protein function prediction approaches often formulate the problem as a multiple binary classification problem [12] and ignore the correlation between labels. To avoid this limitation, multi-label learning is widely used for protein function prediction [10,15,24]. Multi-label learning can make use of label correlations to boost the prediction accuracy and assign more than one function to a protein [20,22]. Other approaches train a binary classifier for each function label, and then organize these classifiers in a hierarchical (tree or direct acyclic graph) structure according to the Function Catalogue (FunCat) [16][4] or Gene Ontology [2][5] database [14]. In this paper, we focus on protein function prediction using multi-label learning and function correlation.

All these approaches assume that the available annotations for the labeled proteins are complete. In practice, we may just have a subset of the functions of a protein, and whether some functions are missing is unknown. In other words, proteins may not be completely annotated [4], i.e., function annotations may be only partial. This kind of multi-label learning problem is called *multi-label weak-label learning* [19], a much less studied problem in the literature [4,22]. Unlike traditional multi-label learning methods [10,21,24], we study protein function prediction using incomplete annotations and propose a technique called *Pro*tein Function Prediction using *D*ependency *M*aximization (ProDM). ProDM can replenish the missing functions of partially annotated proteins and predict the function of completely unlabeled proteins using the partially annotated ones. Our empirical study on publicly available PPI datasets shows that ProDM performs better than other related approaches on these two prediction problems, and it is also computationally efficient.

2 Related Work

Various network-based methods have been proposed for protein function prediction [18]. Schwikowski et al. [17] make predictions for a protein based on the functions of its interacting proteins. They observed that the interacting proteins are likely to share similar functions, which is recognized as the 'guilt by association' rule. Chua et al. [6] found that indirectly interacting proteins share

[4] http://mips.helmholtz-muenchen.de/proj/funcatDB/

[5] http://www.geneontology.org/

few functions, and extended the PPI network by integrating the level-1 (direct) and level-2 (indirect) neighbors using different weights. These methods use a threshold on the predicted likelihood to attach more than one function to a protein. However, these methods do not take into account the correlation among functions.

More recently, multi-label learning approaches [23] have been introduced for protein function prediction. Pandey et al. [15] incorporated function correlations within a weighted k-nearest neighbor classifier, and observed that incorporating function correlations can boost the prediction accuracy. Jiang et al. [10] applied the learning with local and global consistency model [25] on a tensor graph to predict protein functions. Zhang et al. [24] included a function correlation term within the manifold regularization framework [3] to annotate proteins. Jiang et al. [9] conducted label propagation on a bi-relation graph to infer protein functions. To avoid the risk of overwriting functions during label propagation, Yu et al. [21] introduced a Transductive Multi-label Classifier (TMC) on a directed bi-relation graph to annotate proteins. Chi et al. [5] considered the fact that proteins' functions can influence the similarity between pairs of proteins and proposed an iterative model called Cosine Iterative Algorithm (CIA). In each iteration of CIA, the most confidently predicted function of an unlabeled protein is appended to the function set of this protein. Next, the pairwise similarity between training proteins and testing proteins is updated based on the similar functions within the two sets for each protein. CIA uses the updated similarity, function correlations, and PPI network structures to predict the functions on the unlabeled proteins in the following iteration.

All the above multi-label learning approaches focus on utilizing function correlation in various ways and assume that the function annotations on the training proteins are complete and accurate (without missing functions). However, due to various reasons (e.g., the evolving Gene Ontology scheme, or limitations of experimental methods), we may be aware of some of the functions of a protein, but don't know whether other functions are associated with the same protein. Namely, proteins are partially annotated. Learning from partially (or incomplete) labeled data is different from learning from partial labels [7]. In the latter case, one learns from a set of candidate labels of an instance, and assumes that only one label in this set is the ground-truth label. Learning from partially labeled data is also different from semi-supervised and supervised learning, as they both assume complete labels. In this paper, we study how to leverage partially annotated proteins, a less studied scenario in protein function prediction and multi-label learning literature [4,19,22].

Several multi-label weak-label learning approaches have been proposed. Sun et al. [19] introduced a method called WEak Label Learning (WELL). WELL is based on three assumptions: (i) the decision boundary for each label should go across low density regions; (ii) any given label should not be associated to the majority of samples; and (iii) there exists a group of low rank-based similarities, and the approximate similarity between samples with different labels can be computed based on these similarities. WELL uses convex optimization and quadratic programming to replenish the missing labels of a partially labeled sample. As such, WELL is computationally expensive. Buncak et al. [4] annotated unlabeled images using partially labeled images, and proposed a method called MLR-GL. MLR-GL optimizes the ranking loss and group Lasso in a convex optimization form. Yu et al. [22] proposed a method called *Pro*tein function prediction using *W*eak-label *L*earning (ProWL). ProWL can replenish the missing functions of partially annotated proteins, and can predict the functions of completely unlabeled proteins using the partially annotated ones. However, ProWL depends heavily on function correlations and performs the prediction for one function label at a time.

To alleviate these drawbacks associated with ProWL, we develop a new protein function prediction approach called *Pro*tein function prediction using *D*ependency *M*aximization (ProDM). ProDM uses function correlations, the 'guilt by association' rule [17], and maximizes the dependency between the features and function labels of proteins, to complete the prediction for all the function labels at one time. In our empirical study, we observe that ProDM performs better than the other competitive methods in replenishing the missing functions, and performs the best (or comparable to the best) in predicting function for completely unlabeled proteins.

3 Problem Formulation

For the task of *replenishing* missing functions, we have available n partially annotated proteins. The goal is to replenish the missing functions using such partially annotated proteins. For the task of *predicting* the functions of completely unlabeled proteins, we have a total of $n = l + u$ proteins, where the first l proteins are partially annotated and the last u proteins are completely unlabeled. The goal here is to use the l partially annotated proteins to annotate the u unlabeled ones.

Let $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ be the currently available function set, with $y_{ic} = 1$ if protein i has the c-th function, and $y_{ic} = 0$ otherwise. At first, we can define a function correlation matrix $M^{'} \in \mathbb{R}^{C \times C}$ based on cosine similarity as follows:

$$M^{'}_{st} = \frac{Y^T_{.s} Y_{.t}}{\|Y_{.s}\| \|Y_{.t}\|} \tag{1}$$

where $M^{'}_{st}$ is the correlation between functions s and t, and $Y_{.s}$ represents the s-th column of Y. There exists a number of ways (e.g., Jaccard coefficient [24] and Lin's similarity [15]) to define function correlation. Here, we use the cosine similarity for its simplicity and wide application [5,20,22]. If Y is represented in a probabilistic function assignment form, Eq. (1) can also be applied.

From Eq. (1), we can see that $M^{'}_{st}$ measures the fraction of times function s and t co-exist in a protein. We normalize $M^{'}$ as follows:

$$M_{st} = \frac{M^{'}_{st}}{\sum_{c=1}^{C} M^{'}_{sc}} \tag{2}$$

M_{st} can be viewed as the probability that a protein has function t given that it is annotated with function s.

Now, let's consider the scenario with incomplete annotations and extend the observed function set Y to $\tilde{Y} = YM$. Our motivation in using $\tilde{Y}$ is to append the missing functions using the currently known functions and their correlations. More specifically, suppose the currently confirmed functions Y_i for the i-th protein have a large correlation with the c-th function (which may be missing), then it is likely that this protein will also have function c. Based on this assumption, we define the first part of our objective function as follows:

$$\Psi_1(\mathbf{f}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{c=1}^{C}(f_{ic} - \tilde{y}_{ic})^2 = \frac{1}{2}\sum_{i=1}^{n}\|F - \tilde{Y}\|_2^2 \tag{3}$$

where f_{ic} is the predicted likelihood of protein i with respect to the c-th function, $\tilde{y}_{ic}$ is the extended function annotation of protein i with respect to the c-th function, and $F = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n]$ is the prediction for the n proteins.

Since a protein has multiple functions, and the overlap between the function sets of two proteins can be used to measure their similarity, the larger the number of shared functions, the more similar the proteins are. This function induced similarity between proteins was used successfully in Chi et al. [5] and Wang et al. [20]. The function annotations of a protein can be used to enrich its feature representation. Thus, we define the function-based similarity matrix $W^f \in \mathbb{R}^{n\times n}$ between n proteins as follows:

$$W_{ij}^f = \frac{\mathbf{y}_i^T\mathbf{y}_j}{\|\mathbf{y}_i\|\|\mathbf{y}_j\|} \tag{4}$$

Note that W_{ij}^f measures the pairwise similarity (induced by the function sets of two proteins) between proteins i and j, whereas M_{st} in Eq. (2) describes the pairwise function correlations.

We now define a composite similarity W between pairwise proteins as:

$$W = W^p + \eta W^f \tag{5}$$

where $W^p \in \mathbb{R}^{n\times n}$ describes the feature induced similarity between pairs of proteins. Here W^p can be set based on the amino acid sequence similarity of a protein pair (i.e., string kernel [12] for protein sequence data), or by using the frequency of interactions found in multiple PPI studies (i.e., PPI networks in BioGrid[6]), or the weighted pairwise similarity based on reliability scores from all protein identifications by mass spectrometry (e.g., Krogan et al. [11][7]). η is a predefined parameter to balance the tradeoff between W^p and W^f. It is set to $\eta = \sum_{i=1,j=1}^{n,n} W_{ij}^p / \sum_{i=1,j=1}^{n,n} W_{ij}^f$.

The second part of our objective function leverages the knowledge that proteins with similar amino acid sequences are likely to have similar functions. In

[6] http://thebiogrid.org/

[7] http://www.nature.com/nature/journal/v440/n7084/suppinfo/nature04670.html

other words, we capture the ‘guilt by association’ rule [17], which states that interacting proteins are more likely to share similar functions. This rule is widely used in network-based protein function prediction approaches [5,17,18,22]. As in learning with local and global consistency [25], we include a smoothness term as the second part of our objective function:

$$\begin{aligned}\Psi_2(\mathbf{f}) &= \frac{1}{2}\sum_{i,j=1}^{n} \| \frac{\mathbf{f}_i}{\sqrt{D_{ii}}} - \frac{\mathbf{f}_j}{\sqrt{D_{jj}}} \|^2 W_{ij} \\ &= tr(F^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})F) \\ &= tr(F^T LF) \end{aligned} \tag{6}$$

where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$. I is an $n \times n$ identity matrix, $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, and $tr(\cdot)$ is the matrix trace operation.

Here, we assume the function labels of a protein depend on the feature representation of this protein. We encode this assumption as the third part of our objective function. To capture the dependency between the function labels and the features of proteins we take advantage of the Hilbert-Schmidt Independence Criterion (HSIC) [8]. HSIC computes the squared norm of the cross-covariance operator over the feature and label domains in Hilbert Space to estimate the dependency. We choose HSIC because of its computational efficiency, simplicity and solid theoretical foundation. The empirical estimation of HSIC is given by:

$$HSIC(F, Y, p_{\mathbf{xy}}) = \frac{tr(KHSH)}{(n-1)^2} = \frac{tr(HKHS)}{(n-1)^2} \tag{7}$$

where $H, K, S \in \mathbb{R}^{n \times n}$, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is used to measure the kernel induced similarity between two samples, $S_{ij} = s(\mathbf{f}_i, \mathbf{f}_j)$ is used to describe the label induced similarity between two samples, $H_{ij} = \delta_{ij} - \frac{1}{n}$, $\delta_{ij} = 1$ if $i = j$, otherwise $\delta_{ij} = 0$, $p_{\mathbf{xy}}$ is the joint distribution of $\mathbf{x}$ and $\mathbf{y}$. HSIC makes use of kernel matrices to estimate the dependency between labels and features of samples, thus it can also be applied in the case that there is no explicit feature representation for the n samples, as in the case of PPI network data. Although there are many other ways to initialize K and S, in this paper, we set $K = W$ and $S_{ij} = \mathbf{y}_i^T \mathbf{y}_j$ for its simplicity and its strong empirical performance. Alternative initializations of K and S will be investigated in our future study.

3.1 The Algorithm

By integrating the three objective functions introduced above, we obtain the overall objective function of ProDM:

$$\Psi(F) = tr(F^T LF) + \alpha\|F - \tilde{Y}\|_2^2 - \beta tr(HKHFF^T) + \gamma tr(F^T F) \tag{8}$$

where $\alpha > 0$ and $\beta > 0$ are used to balance the tradeoff between the three terms. Our motivation to minimize $\Psi(F)$ is three-fold: (i) two proteins with similar sequences (or frequently interacting) should have similar functions, which corresponds to the smoothness assumption in label propagation [25]; (ii) predictions in F should not change too much from the extended function labels $\tilde{Y}$; and (iii) the dependency between the function labels and the features of a protein should be maximized. In Eq. (8) we also add a term $tr(F^T F)$ (weighted by $\gamma > 0$) to enforce the sparsity of F, since each function is often associated with a relatively small number of proteins.

ProWL [22] makes use of function correlations and the 'guilt by association' rule to replenish the missing functions of partially annotated proteins. In addition, ProDM incorporates the assumption of dependency maximization. ProWL relies on the function correlation matrix M to extend the observed function annotations and to define the weight of each function label of a protein (see Eq. (3) in [22]). In contrast, ProDM exploits the function correlations to expand the incomplete function sets. As the number of missing functions increases, the function correlation matrix M becomes less reliable [22]. Therefore, when the number of missing functions is large, ProDM outperforms ProWL. In addition, ProWL predicts each function label separately and computes the inverse of a matrix for each label. ProDM, instead, predicts all C labels at once, and computes the inverse of a matrix only once. As a result, ProDM is faster than ProWL. These advantages of ProDM with respect to ProWL are corroborated in our experiments.

Eq. (8) can be solved by taking the derivative of $\Psi(F)$ with respect to F:

$$\frac{\partial \Psi(F)}{\partial F} = 2(LF + \alpha(F - \tilde{Y}) - \beta HKHF + \gamma F) \tag{9}$$

By setting $\frac{\partial \Phi(F)}{\partial F} = 0$, we obtain:

$$F = \alpha(L + \alpha I - \beta HKH + \gamma I)^{-1} \tilde{Y} \tag{10}$$

In Eq. (10), the complexity of the matrix multiplication HKH is $O(n^3)$ and the complexity of the matrix inverse operation is $O(n^3)$. Thus, the time complexity of ProDM is $O(n^3)$. In practice, though, L, H, and K are all s-parse matrices, and Eq. (10) can be computed more efficiently. In particular, the complexity of sparse matrix multiplication is $O(nm_1)$, where m_1 is the number of nonzero elements in K. In addition, instead of computing the inverse of $(L + \alpha I - \beta HKH + \gamma I)$ in Eq. (10), we can use iterative solvers (i.e., Conjugate Gradient (CG)). CG is guaranteed to terminate in n iterations. In each iteration, the most time-consuming operation is the product between an $n \times n$ sparse matrix and a label vector (one column of $\tilde{Y}$). Thus, in practice, the time complexity of ProDM is $O(m_1 n + t m_2 n C)$, where C is the number of function labels, m_2 is the number of nonzero elements in $(L + \alpha I - \beta HKH + \gamma I)$, and t is the number of CG iterations. CG often terminates in no more than 20 iterations.

4 Experimental Setup

Datasets We investigate the performance of ProDM on replenishing missing functions and predicting protein functions on three different PPI benchmarks.

The first dataset, *Saccharomyces Cerevisiae* PPIs (ScPPI), is extracted from BioGrid[8]. We annotate these proteins according to FunCat [16] database and use the largest connected component of ScPPI for experiments, which includes 3041 proteins. FunCat organizes function labels in a tree structure. We filtered the function labels and used the 86 informative functions. Informative functions [10,24] are the ones that have at least 30 proteins as members and within the tree structure these functions do not have a particular descendent node with more than 30 proteins. The weight matrix W^p of ScPPI is specified by the number of PubMed IDs, where 0 means no interaction between two proteins, and $q > 0$ implies the interaction is supported by q distinct PubMed IDs. The second dataset, KroganPPI is obtained from the study of Krogan et al. [11][9]. We use its largest connected component for the experiments and annotate these proteins according to FunCat. After the preprocessing, KroganPPI contains 3642 proteins annotated with 90 informative functions. The weight matrix of W^p is specified by the provider. The third dataset, HumanPPI is obtained from the study of Mostafavi et al. [13][10]. HumanPPI is extracted from the multiple data types of Human Proteomic data. The proteins in HumanPPI are annotated according to the Gene Ontology [2]. Similarly to [10,13], we use the largest connected components of HumanPPI and the functions that have at least 30 annotated proteins. The weight matrix W^p of HumanPPI is specified by the provider. The characteristics of these processed datasets are listed in Table 1.

Table 1. Dataset Statistics (Avg±Std means average number of functions for each protein and its standard deviation)

Dataset	#Proteins	#Functions	Avg±Std
ScPPI	3041	86	2.49 ± 1.70
KroganPPI	3642	90	2.20 ± 1.60
HumanPPI	2950	200	3.80 ± 3.77

Comparative Methods We compare the proposed method with: (i) ProWL [22], (ii) WELL [19][11], (iii) MLR-GL [4][12], (iv) TMC [21], and (v) CIA [5]. The first three approaches are multi-label learning models with partially labeled data, and the last two methods are recently proposed protein function prediction algorithms based on multi-label learning and PPI networks. WELL and MLR-GL need an input kernel matrix. We substitute the kernel matrix with W^p, which is semi-definite positive and can be viewed as a Mercer kernel [1]. WELL was proposed to replenish the missing functions of partially annotated proteins. We adopt it here to predict the functions of completely unlabeled proteins by

[8] http://thebiogrid.org/

[9] http://www.nature.com/nature/journal/v440/n7084/suppinfo/nature04670.html

[10] http://morrislab.med.utoronto.ca/∼sara/SW/

[11] http://lamda.nju.edu.cn/code_WELL.ashx

[12] http://www.cse.msu.edu/∼bucakser/MLR_GL.rar

including the unlabeled proteins in the input kernel matrix. MLR-GL is targeted at predicting the functions of completely unlabeled proteins using partially annotated proteins. We adapt it to replenish the missing functions of partially annotated proteins by using all the proteins as training and testing set. As was done for MLR-GL, we also adapt TMC to replenish the missing functions. Due to the iterative procedure of CIA, it cannot be easily adapted to replenish missing functions. The parameters of WELL, MLR-GL, ProWL, TMC, and CIA are set as the authors specified in their code, or reported in the papers. For ProDM, we search for optimal α values in the range [0.5, 1] with step size 0.05, and β values in the range [0.01, 0.1] with step size 0.01. In our experiments, we set α and β to 0.99 and 0.01, respectively, since we observed that the performance with respect to the various metrics does not change as we vary α and β around the fixed values. Similarly to ProWL, we set γ to 0.001.

Experimental Protocol In order to simulate the incomplete annotation scenario, we assume the annotations on the currently labeled proteins are complete and mask some of the ground truth functions. The masked functions are considered missing. For presentation, we define a term called *Incomplete Function* (IF) ratio, which measures the ratio between the number of missing functions and the number of ground truth functions. For example, if a protein has five functions (labels), and two of them are masked (two 1s are changed to two 0s), then the IF ratio is $2/5 = 40\%$.

Evaluation Criteria Protein function prediction can be viewed as a multi-label learning problem and evaluated using multi-label learning metrics [10,22]. Various evaluation metrics have been developed for evaluating multi-label learning methods [23]. Here we use five metrics: *MicroF1*, *MacroF1*, *HammingLoss*, *RankingLoss* and adapted *AUC* [4]. These metrics were also used to evaluate WELL [19], MLR-GL [4], and ProWL [22]. In addition, we design *RAccuracy* to evaluate the performance of replenishing missing functions. Suppose the predicted function set of n proteins is F_p, the initial incomplete annotated function set is F_q, and the ground truth function set is Y. *RAccuracy* is defined as follows:

$$RAccuracy = \frac{|(Y - F_q) \cap F_p|}{|(Y - F_q)|}$$

where $|(Y - F_q)|$ measures how many functions are missing among n proteins and $|(Y - F_q) \cap F_p|$ counts how many missing functions are correctly replenished. To maintain consistency with other evaluation metrics, we report *1-HammLoss* and *1-RankLoss*. Thus, similarly to other metrics, the higher the values of *1-HammLoss* and *1-RankLoss*, the better the performance.

5 Experimental Analysis

5.1 Replenishing Missing Functions

We performed experiments to investigate the performance of ProDM on replenishing the missing functions of n partially labeled proteins. To this end, we consider all the proteins in each dataset as training and testing data. To perform comparisons against the other methods, we vary the IF ratio from 30% to 70%, with an interval of 20%. A few proteins in the PPI networks do not have any functions. To make use of the 'guilt by association' rule and keep the PPI network connected, we do not remove them and test the performance of replenishing missing functions on the proteins with annotations. We repeat the experiments 20 times with respect to each IF ratio. In each run, the missing functions are randomly masked for each protein according to the IF ratio. $F \in \mathbb{R}^{n \times C}$ in Eq. (10) is a predicted likelihood matrix. *MicroF1*, *MacroF1*, *1-HammLoss* and *RAccuracy* require F to be a binary indicator matrix. Here, we consider the functions corresponding to the r largest values of $\mathbf{f}_i$ as the functions of the i-th protein, where r is determined by the number of ground-truth functions of this protein. To simulate the incomplete annotation scenario, we assume the given functions of the i-th protein in a dataset are ground-truth functions, and mask some of them to generate the missing functions. The experimental results are reported in Tables 2-4. In these tables, best and comparable results are in **boldface** (statistical significance is examined via pairwise t-test at 95% significance level).

Table 2. Results of replenishing missing functions on **ScPPI**

Metric	IF Ratio	ProDM	ProWL	WELL	MLR-GL	TMC
MicroF1	30%	**93.88±0.12**	86.28±0.14	60.49±0.54	23.67±0.50	91.80±0.20
	50%	**79.09±0.28**	68.36±0.36	47.42±0.74	26.98±0.49	77.09±0.28
	70%	**71.67±0.51**	60.09±0.51	42.06±0.04	27.15±0.59	69.79±0.44
MacroF1	30%	**94.05±0.18**	86.28±0.18	55.35±0.52	24.06±0.79	90.98±0.24
	50%	**78.39±0.33**	67.81±0.36	43.80±0.55	27.45±0.72	74.72±0.35
	70%	**70.05±0.45**	59.45±0.62	38.25±0.87	27.98±0.72	67.34±0.52
1-HammLoss	30%	**99.65±0.01**	99.20±0.01	97.71±0.03	95.58±0.03	99.52±0.01
	50%	**98.79±0.02**	98.17±0.02	96.95±0.04	95.77±0.03	98.67±0.02
	70%	**98.36±0.03**	97.69±0.03	96.64±0.00	95.78±0.03	98.25±0.03
1-RankLoss	30%	**99.67±0.02**	95.16±0.02	94.78±0.07	44.38±0.39	99.65±0.02
	50%	96.80±0.12	91.95±0.24	90.41±0.24	41.43±0.66	**97.06±0.10**
	70%	**94.92±0.17**	88.03±0.24	89.01±0.26	38.06±0.77	94.52±0.29
AUC	30%	**98.79±0.05**	94.92±0.04	93.09±0.04	55.63±0.38	**98.77±0.04**
	50%	95.63±0.14	92.07±0.16	88.24±0.24	54.01±0.66	**95.97±0.10**
	70%	**93.09±0.22**	88.85±0.20	86.08±0.35	52.60±0.46	**93.04±0.29**
RAccuracy	30%	**49.24±1.28**	38.05±1.07	23.94±1.55	46.18±1.04	46.01±1.52
	50%	**46.57±0.71**	32.14±0.92	18.83±1.01	35.59±0.91	42.46±0.76
	70%	**44.18±1.03**	31.41±1.03	17.12±0.12	33.89±0.74	41.42±0.82

From these Tables (2-4), we can observe that ProDM performs much better than the competitive methods in replenishing the missing functions of proteins across all the metrics. Both ProDM and ProWL take advantage of function correlations and of the 'guilt by association' rule, but ProDM significantly outperforms ProWL. The difference in performance between ProDM and ProWL confirms our intuition that maximizing the dependency between functions and features of proteins is effective. The performance of WELL is not comparable to that of ProDM. The possible reason is that the assumptions used in WELL may be not suitable for the PPI network datasets. The performance of MLR-GL varies because it is targeted at predicting functions of unlabeled proteins using partially annotated proteins, whereas here it is adapted for replenishing missing functions. TMC is introduced to predict functions for completely unlabeled proteins using completely labeled ones; TMC sometimes outperforms ProWL and

Table 3. Results of replenishing missing functions on **KroganPPI**

Metric	IF Ratio	ProDM	ProWL	WELL	MLR-GL	TMC
MicroF1	30%	**95.51±0.13**	93.05±0.08	61.04±0.27	14.78±0.23	88.67±0.12
	50%	**79.46±0.22**	68.39±0.27	48.54±0.67	16.18±0.29	70.93±0.22
	70%	**70.23±0.35**	60.25±0.29	43.72±0.19	16.09±0.34	61.82±0.31
MacroF1	30%	**95.70±0.18**	94.57±0.15	58.24±0.20	13.71±0.28	88.41±0.12
	50%	**78.92±0.25**	71.51±0.32	52.09±1.08	15.12±0.34	69.20±0.33
	70%	**69.01±0.40**	62.30±0.46	48.79±0.52	14.92±0.35	60.20±0.44
1-HammLoss	30%	**99.78±0.01**	99.66±0.00	98.08±0.01	95.81±0.01	99.44±0.01
	50%	**98.99±0.01**	98.44±0.01	97.47±0.03	95.87±0.01	98.57±0.01
	70%	**98.53±0.02**	98.04±0.01	97.23±0.01	95.87±0.02	98.12±0.02
1-RankLoss	30%	**99.75±0.02**	99.61±0.02	96.50±0.03	39.88±0.37	99.52±0.02
	50%	**96.87±0.12**	94.55±0.12	91.60±0.09	39.99±0.27	96.20±0.16
	70%	**94.37±0.14**	91.02±0.25	89.89±0.06	38.48±0.39	93.28±0.19
AUC	30%	**98.87±0.04**	98.58±0.04	94.90±0.05	45.49±0.28	98.59±0.05
	50%	**95.47±0.12**	92.55±0.15	88.88±0.14	46.65±0.32	94.63±0.18
	70%	**91.91±0.16**	86.90±0.35	85.87±0.10	46.45±0.37	90.58±0.24
RAccuracy	30%	**44.97±1.63**	14.90±0.98	9.24±0.66	30.90±1.48	23.89±1.30
	50%	**42.20±0.63**	11.04±0.77	7.03±0.22	23.83±0.71	27.89±0.61
	70%	**36.25±0.75**	14.89±0.61	7.68±0.44	21.69±0.80	27.06±0.65

Table 4. Results of replenishing missing functions on **HumanPPI**

Metric	IF Ratio	ProDM	ProWL	WELL	MLR-GL	TMC
MicroF1	30%	**96.60±0.14**	95.12±0.14	86.21±0.10	15.76±0.30	91.90±0.15
	50%	**88.48±0.41**	77.18±0.24	64.93±0.26	16.36±0.21	77.98±0.27
	70%	**79.20±0.55**	61.91±0.30	51.91±0.46	16.10±0.29	69.05±0.31
MacroF1	30%	**96.21±0.16**	94.76±0.16	87.95±0.03	15.79±0.27	91.43±0.15
	50%	**87.49±0.46**	76.86±0.30	70.43±0.18	16.00±0.26	77.05±0.31
	70%	**77.58±0.53**	62.19±0.30	59.05±0.37	15.45±0.26	67.67±0.35
1-HammLoss	30%	**99.87±0.01**	99.81±0.01	99.48±0.00	96.80±0.01	99.69±0.01
	50%	**99.56±0.02**	99.13±0.01	98.67±0.01	96.82±0.01	99.16±0.01
	70%	**99.21±0.02**	98.55±0.01	98.17±0.02	96.82±0.01	98.83±0.01
1-RankLoss	30%	**99.81±0.02**	99.74±0.03	97.19±0.03	54.78±0.32	99.73±0.02
	50%	**98.73±0.07**	96.90±0.21	87.55±0.44	58.09±0.29	98.31±0.12
	70%	**97.50±0.15**	93.56±0.41	83.97±0.08	58.35±0.36	96.76±0.21
AUC	30%	**98.65±0.04**	98.52±0.05	93.51±0.13	54.32±0.22	98.44±0.04
	50%	**97.37±0.09**	95.86±0.15	83.05±0.20	55.90±0.21	96.82±0.10
	70%	**95.48±0.14**	91.31±0.28	76.12±0.48	55.69±0.26	94.64±0.18
RAccuracy	30%	**80.39±0.80**	71.86±0.79	20.50±0.59	30.92±1.09	53.35±0.83
	50%	**73.14±0.96**	46.78±0.55	18.23±0.62	23.92±0.56	48.66±0.63
	70%	**63.28±0.97**	32.76±0.53	15.09±0.82	21.41±0.45	45.36±0.55

WELL. This is because the missing functions can be appended in the bi-relation graph. In fact, TMC also makes use of function correlations and the 'guilt by association' rule, but it still loses to ProDM. The reason is that ProDM maximizes the dependency between proteins' functions and features. The margin in performance achieved by ProDM with respect to ProWL and TMC demonstrates the effectiveness of using *dependency maximization* in replenishing the missing functions of proteins.

We also observe that, as more functions are masked, ProWL downgrades much more rapidly than ProDM. As the IF ratio increases, the function correlation matrix M becomes less reliable. ProWL uses M to estimate the likelihood of missing functions and to weigh the loss function. ProDM only utilizes M to estimate the probability of missing functions and makes additional use of dependency maximization. Thus ProDM is less dependent on M. Taking *RAccuracy* on ScPPI as an example, ProDM on average is 33.55% better than ProWL, 49.60% better than WELL, 19.31% better than MLR-GL, and 8.21% better than TMC. These results confirm the effectiveness of ProDM in replenishing the missing functions. Overall, this experimental results confirm the advantages of combining the 'guilt by association' rule, function correlations, and dependency maximization.

5.2 Predicting Unlabeled Proteins

We conduct another set of experiments to study the performance of ProDM in predicting the function of completely unlabeled proteins using partially labeled ones. In this scenario, $l < n$ proteins are partially annotated and $n - l$ proteins are completely unlabeled. At first, we partition each dataset into a *training* set (accounting for 80% of all the proteins) with partial annotations and into a *testing* set (accounting for the remaining 20% of all the proteins) with no annotations. We run the experiments 20 times for each dataset. In each round, the dataset is randomly divided into training and testing datasets. We simulate the setting of missing functions (IF ratio=50%) in the training set as done in the experiments in Section 5.1, but r is determined as the average number of functions (round to the next integer) of all proteins. From Table 1: r is set to 3 for ScPPI and KroganPPI, and to 4 for HumanPPI. The results (average of 20 independent runs) are listed in Tables 5-7. Since *RAccuracy* is not suitable for the settings of predicting completely unlabeled proteins, the results for this metric are not reported.

Table 5. Prediction results on completely unlabeled proteins of **ScPPI**

Metric	ProDM	ProWL	WELL	MLR-GL	TMC	CIA
MicroF1	**32.78±1.37**	30.06±1.15	16.75±2.03	24.15±1.40	3.67±0.38	20.78±0.38
MacroF1	**31.91±1.48**	**31.33±1.74**	5.19±0.71	26.25±1.50	2.00±0.39	26.27±0.39
1-HammLoss	**95.73±0.10**	95.56±0.09	94.69±0.16	95.19±0.09	93.89±0.05	94.96±0.05
1-RankLoss	**73.13±2.72**	60.37±1.64	73.57±0.05	41.56±1.06	28.29±0.70	21.82±0.70
AUC	**78.40±1.57**	**78.63±0.74**	77.00±0.53	61.47±1.26	55.72±0.84	63.38±0.84

Table 6. Prediction results on completely unlabeled proteins of **KroganPPI**

Metric	ProDM	ProWL	WELL	MLR-GL	TMC	CIA
MicroF1	**22.55±1.35**	**22.40±0.97**	14.35±1.25	13.58±0.86	3.32±0.52	13.78±0.52
MacroF1	**18.26±1.53**	**17.68±1.11**	1.47±0.30	12.80±0.92	2.05±0.41	13.85±0.41
1-HammLoss	**96.40±0.08**	**96.40±0.08**	96.04±0.03	95.99±0.07	95.52±0.06	95.99±0.06
1-RankLoss	66.69±1.19	**75.41±0.88**	**75.43±0.22**	48.40±1.13	61.26±0.89	18.43±0.89
AUC	72.26±0.73	**74.78±0.73**	**74.16±0.12**	58.80±1.10	61.35±0.68	59.45±0.68

Table 7. Prediction results on completely unlabeled proteins of **HumanPPI**

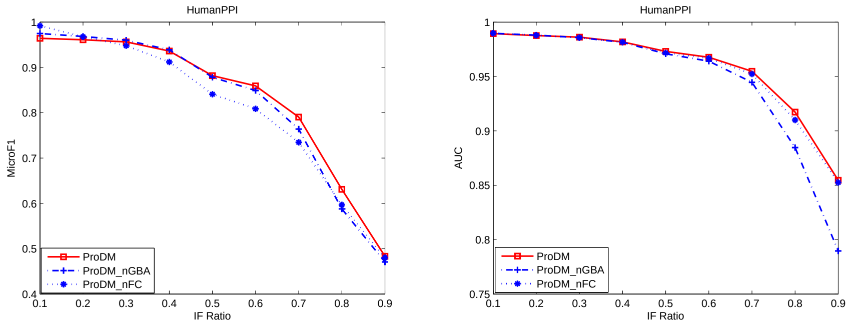
Metric	ProDM	ProWL	WELL	MLR-GL	TMC	CIA
MicroF1	**24.57±1.03**	23.18±1.24	16.43±1.78	12.87±0.76	1.91±0.28	12.86±0.28
MacroF1	**20.58±1.18**	19.32±0.90	15.55±1.30	11.95±0.78	1.61±0.26	9.90±0.26
1-HammLoss	**97.17±0.05**	97.11±0.09	96.85±0.10	96.73±0.05	96.33±0.07	96.72±0.07
1-RankLoss	**76.70±1.07**	**76.64±2.01**	62.98±1.82	67.89±1.44	50.93±0.77	33.87±0.77
AUC	**78.82±1.19**	77.41±0.92	62.30±1.38	66.23±0.85	51.78±1.21	67.08±1.21

From Tables 5-7, we can observe that ProDM achieves the best (or comparable to the best) performance among all the comparing methods on various evaluation metrics. ProDM and ProWL have similar performance in the task of predicting the functions of completely unlabeled proteins. One possible reason is that F is initially set to $\tilde{Y}$ and $\{\tilde{\mathbf{y}}_j\}_{j=l+1}^{n}$ are zero vectors. WELL works better than MLR-GL in replenishing the missing functions, and it loses to MLR-GL in predicting the functions of unlabeled proteins. One possible cause is that WELL is targeted at replenishing missing functions, and here it's adjusted to predict functions on completely unlabeled proteins. MLR-GL predicts protein functions under the assumption of partially annotated proteins, and it is outperformed by ProDM. MLR-GL optimizes the ranking loss and the group Lasso loss, whereas ProDM optimizes an objective function based on the function correlations, the 'guilt by association' rule, and the dependency between the function labels and the features of proteins. We can claim that ProDM is more faithful to the characteristics of proteomic data than MLR-GL. For the same reasons, ProDM often outperforms WELL, which takes advantage of low density separation and low-rank based similarity to capture function correlations and data distribution.

TMC sometimes performs similar to ProDM in the task of replenishing the missing functions. However, TMC is outperformed by other methods when making predictions for completely unlabeled proteins. A possible reason is that TMC assumes the training proteins are fully annotated, and the estimated function correlation matrix M may be unreliable when IF ratio is set to 50%. CIA also exploits function-based similarity and PPI networks to predict protein functions, but it's always outperformed by ProDM and by ProWL. There are two possible reasons. First, CIA does not account for the weights of interaction between two proteins. Second, CIA mainly relies on the function induced similarity W^f, and when training proteins are partially annotated, this similarity becomes less reliable. CIA performs better than TMC. One reason might be that CIA exploits a neighborhood count algorithm [17] to initialize the functions on unlabeled proteins in the kick-off step of CIA, whereas TMC does not. All these results show the effectiveness of ProDM in predicting unlabeled proteins by considering the partial annotations on proteins.

5.3 Component Analysis

To investigate the benefit of using the 'guilt by association' rule and of exploiting function correlations, we introduce two variants of ProDM, namely ProDM_nGBA and ProDM_nFC. ProDM_nGBA corresponds to *Pro*tein function prediction using *D*ependency *M*aximization with *n*o '*G*uilt *B*y *A*ssociation' rule. Specifically, ProDM_nGBA is based on Eq. (8) without the first term; that is, ProDM_nGBA uses only the partial annotations and function correlations to replenish the missing functions. ProDM_nFC corresponds to *Pro*tein function prediction using *D*ependency *M*aximization with *n*o *F*unction *C*orrelation. In ProDM_nFC, Y is used in Eq. (8) instead of $\tilde{Y}$. We increase the IF ratio from 10% to 90% at intervals of 10%, and record the results of ProDM, ProDM_nGBA and ProDM_nFC with respect to each IF ratio. For brevity, in Figure 1 we just report the results with respect to *MicroF1* and *AUC* on HumanPPI.



(a) HumanPPI (MicroF1)

(b) HumanPPI (AUC)

Fig. 1. The benefit of using both the 'guilt by association' rule and function correlations (ProDM_nFC is ProDM with no function correlation, and ProDM_nGBA is ProDM with no 'guilt by association' rule)

From Figure 1, we can observe that ProDM, ProDM_nGBA, and ProDM_nFC have similar performance when few functions are missing. This indicates that both the 'guilt by association' rule and function correlations can be utilized to replenish the missing functions. However, as the number of missing function increases, ProDM generally outperforms ProDM_nGBA and ProDM_nFC. The reason is that ProDM, unlike ProDM_nGBA and ProDM_nFC, makes use of *both* the 'guilt by association' rule and function correlations. This fact shows that it's important and reasonable to integrate these two components in replenishing missing functions.

5.4 Run Time Analysis

Table 8. Runtime Analysis (seconds)

Dataset	ProDM	ProWL	WELL	MLR-GL	TMC
ScPPI	60.77	83.09	1687.09	22.66	2.29
KroganPPI	80.60	134.94	3780.24	32.40	3.62
HumanPPI	64.02	194.62	5445.97	50.68	3.49
Total	178.37	412.65	10913.30	105.74	9.40

In Table 8 we record the average run time of each of the methods on the three datasets. The experiments are conducted on Windows 7 platform with Intel E31245 processor and 16GB memory. TMC assumes the training proteins are accurately annotated, and it takes much less time than the other methods. MLR-GL relaxes the convex-concave optimization problem into a Second Order Cone Programming (SOCP) [4] problem, and it ranks 2nd (from fast to slow). ProDM takes less time than ProWL, since ProDM infers the functions of a protein in one step, whereas ProWL divides the prediction into C subproblems. WELL uses eigen-decomposition and convex optimization, and it costs much more than the other methods. As such, it is desirable to use ProDM for protein function prediction.

6 Conclusions

In this paper, we study protein function prediction using partially annotated proteins and introduce the ProDM method. ProDM integrates the maximization of dependency between features and function labels of proteins, the 'guilt by association' rule, and function correlations to replenish the missing functions of partially annotated proteins, and to predict the functions of completely unlabeled proteins. Our empirical study on three PPI networks datasets shows that the proposed ProDM performs significantly better than the competitive methods. In addition, we empirically demonstrate the benefit of integrating the 'guilt by association' rule, function correlations, and dependency maximization in protein function prediction.

7 Acknowledgements

The authors want to thank the anonymous reviewers for helpful comments. This work is partially supported by NSF IIS-0905117 and NSF Career Award IIS-1252318, NSFC (61003174, 61070090), Natural Science Foundation of Guangdong Province (S2012010009961), Specialized Research Fund for the Doctoral Program of Higher Education (20110172120027), and China Scholarship Council (CSC).

References

1. A. Aizerman, E.M. Braverman, and L.I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
2. M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
3. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
4. S.S. Bucak, R. Jin, and A.K. Jain. Multi-label learning with incomplete class assignments. In *Proceedings of 24th IEEE Conference on Computer Vision and Pattern Recognition*, pages 2801–2808, 2011.
5. X. Chi and J. Hou. An iterative approach of protein function prediction. *BMC Bioinformatics*, 12(1):437, 2011.
6. H.N. Chua, W. Sung, and L. Wong. Using indirect protein interactions for the prediction of gene ontology functions. *BMC Bioinformatics*, 8(Suppl 4):S8, 2007.
7. T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.
8. A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*, pages 63–77. Springer, 2005.

9. J.Q. Jiang. Learning protein functions from bi-relational graph of proteins and function annotations. In *Algorithms in Bioinformatics*, pages 128–138. Springer, 2011.
10. J.Q. Jiang and L.J. McQuay. Predicting protein function by multi-label correlated semi-supervised learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(4):1059–1069, 2012.
11. N.J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A.P. Tikuisis, et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.
12. C.S. Leslie, E. Eskin, A. Cohen, J. Weston, and W.S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.
13. S. Mostafavi and Q. Morris. Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, 26(14):1759–1765, 2010.
14. G. Pandey, V. Kumar, and M. Steinbach. Computational approaches for protein function prediction. Technical Report TR 06-028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 2006.
15. G. Pandey, C. Myers, and V. Kumar. Incorporating functional inter-relationships into protein function prediction algorithms. *BMC Bioinformatics*, 10(1):142, 2009.
16. A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Güldener, G. Mannhaupt, M. Münsterkötter, et al. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.
17. B. Schwikowski, P. Uetz, S. Fields, et al. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18(12):1257–1261, 2000.
18. R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(1), 2007.
19. Y. Sun, Y. Zhang, and Z. Zhou. Multi-label learning with weak label. In *Proceedings of 24th AAAI Conference on Artificial Intelligence*, 2010.
20. H. Wang, H. Huang, and C. Ding. Function-function correlated multi-label protein function prediction over interaction networks. In *Proceedings of the 16th International Conference on Research in Computational Molecular Biology*, pages 302–313, 2012.
21. G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, and Z. Yu. Transductive multi-label ensemble classification for protein function prediction. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1077–1085, 2012.
22. G. Yu, G. Zhang, H. Rangwala, C. Domeniconi, and Z. Yu. Protein function prediction using weak-label learning. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 202–209, 2012.
23. M. Zhang and Z. Zhou. A review on multi-label learning algorithms. *IEEE Transctions on Knowledge and Data Engineering*, 99(99):1, 2013.
24. X. Zhang and D. Dai. A framework for incorporating functional interrelationships into protein function prediction algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(3):740–753, 2012.
25. D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Proceedings of Advances in Neural Information Processing Systems*, pages 321–328, 2003.