Transductive Multi-label Ensemble Classification for Protein Function Prediction

Guoxian Yu
School of Comp. Sci. & Tech.
South China Univ. of Tech.
Guangzhou,510006,China
guoxian.yu@mail.scut.edu.cn

Carlotta Domeniconi
Dept. of Computer Science
George Mason University
Fairfax,22030,VA
carlotta@cs.gmu.edu

Huzefa Rangwala
Dept. of Computer Science
George Mason University
Fairfax,22030,VA
rangwala@cs.gmu.edu

Guoji Zhang
School of Sciences
South China Univ. of Tech.
Guangzhou,510640,China
magjzh@scut.edu.cn

Zhiwen Yu
School of Comp. Sci. & Tech.
South China Univ. of Tech.
Guangzhou,510006,China
zhwyu@scut.edu.cn

ABSTRACT

Advances in biotechnology have made available multitudes of heterogeneous proteomic and genomic data. Integrating these heterogeneous data sources, to automatically infer the function of proteins, is a fundamental challenge in computational biology. Several approaches represent each data source with a kernel (similarity) function. The resulting kernels are then integrated to determine a composite kernel, which is used for developing a function prediction model. Proteins are also found to have multiple roles and functions. As such, several approaches cast the protein function prediction problem within a multi-label learning framework. In our work we develop an approach that takes advantage of several unlabeled proteins, along with multiple data sources and multiple functions of proteins. We develop a graph-based *transductive multi-label classifier* (TMC) that is evaluated on a composite kernel, and also propose a method for data integration using the ensemble framework, called *transductive multi-label ensemble classifier* (TMEC). The TMEC approach trains a graph-based multi-label classifier for each individual kernel, and then combines the predictions of the individual models. Our contribution is the use of a bi-relational *directed* graph that captures relationships between pairs of proteins, between pairs of functions, and between proteins and functions. We evaluate the ability of TMC and TMEC to predict the functions of proteins by using two yeast datasets. We show that our approach performs better than recently proposed protein function prediction methods on composite and multiple kernels.

KDD'12, August 12–16, 2012, Beijing, China.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology- Classifier Design and Evaluation; J.3 [**Life and Medical Sciences**]: Biology and Genetics

General Terms

Algorithms, Performance, Experimentation

Keywords

Multi-label Ensemble Classifier, Directed Bi-relation Graph, Protein Function Prediction

1. INTRODUCTION

Rapid advances in biotechnology have resulted in a variety of high throughput experimentally obtained genomic and proteomic datasets. Examples include protein-protein interaction networks, microarrays, genome sequences, protein structures, and genetic interaction networks, each of which provide a complementary view of the underlying mechanisms within a living cell. Manually annotating a protein, i.e. determining the "protein function", using the vast volumes of data available from heterogeneous sources is challenging and low throughput. As such, various computational methods have been developed for predicting protein functions by integrating the available biological data [17, 18].

Kernel based approaches [12, 14] have been very popular for developing several bioinformatics tools. The data is represented by means of a kernel function, $\mathcal{K}$, that computes pairwise similarities between proteins (or genes). Kernel functions capture the underlying biological complexity associated with the data. For each data source, a unique kernel function is defined, and each kernel function captures a different notion of similarity. For example, for protein sequences a string kernel [13] can be defined, and for protein-protein interaction (PPI) data a random walk kernel [23] function can be used. Both the sequence and PPI datasets are now defined by different kernel functions, each of which captures similarities between protein pairs within different feature spaces (or embeddings). Several methods [17, 24] have been developed to take advantage of the complementary embeddings induced by different data sources. Many approaches

[12, 15, 27] use a weighted combination (optimal or ad hoc) of multiple kernels derived from the different sources. This kind of approaches can be coined as *data integration*. In addition, supervised ensemble approaches [20, 21] have been developed to integrate the multiple data sources.

Several supervised [20] and semi-supervised [24, 27, 29] approaches have been introduced for predicting the function of proteins. Transductive or semi-supervised approaches are able to take advantage of the large number of available unannotated (unlabeled) proteins to improve the accuracy of function prediction algorithms. Further, proteins are generally involved in more than one biological process, and as such are annotated with multiple functions. Several approaches [9, 10, 19, 30] formulate the protein function prediction problem within a multi-label framework. By treating the dependencies between the different function classes, multi-label approaches achieve superior classification performance in comparison to single-labeled prediction methods.

In this paper we develop a protein function prediction method called *Transductive Multi-label Classifier* (TMC) that treats the relationship between proteins and functions as a directed bi-relation graph. To integrate the heterogeneous sources of protein data, we also develop an ensemble based classifier called *Transductive Multi-label Ensemble Classifier* (TMEC). We can summarize our key contributions as follows:

1. We analyze the use of an undirected bi-relation graph [28] within the label propagation framework, and discuss the advantage of a *directed* bi-relation graph instead.

2. We design a TMC on the directed bi-relation graph, and a weighted ensemble of TMCs (TMEC) trained on different graphs to take advantage of multiple complimentary data sources.

3. We compare *classifier integration* against *data integration*, and show the former is often more effective for protein function prediction.

We performed comprehensive experiments evaluating the performance of TMC and TMEC on two yeast protein function prediction benchmark sets. One of the datasets contains 5 kernels, whereas the other one contains 44 kernels. Our results show that the use of a directed bi-relation graph achieves higher accuracy than the undirected one. Our approaches outperform state-of-the-art protein function prediction approaches; namely, two transductive multi-label classification approaches [9, 30] and a transductive classifier [15].

The rest of the paper is organized as follows. In Section 2, we review related work on protein function prediction using multi-label learning and multiple data sources. In Section 3, we introduce the directed bi-relation graph and the corresponding training procedure. We also describe our ensemble approach to make use of multiple data sources. Section 4 details the experimental protocol, and Section 5 discusses the experimental results. In Section 6, we provide conclusions along with directions for future work.

2. PRIOR WORK

Numerous computational approaches have been developed for protein function prediction. They vary in terms of methodology, input data, and even problem definition. We refer the reader to a comprehensive review on this topic [18], and discuss here only the work most relevant to the scope of the paper.

2.1 Multi-label Learning in Protein Function Prediction

Traditional function prediction methods represent proteins in a feature space and train binary classification models, treating each functional class independently [20]. Given the fact that proteins are involved in multiple functions, and structured relationships are prevalent within protein function annotation databases (e.g., GO[1] is a directed acyclic graph), researchers have formulated the protein function prediction as a multi-label classification problem [26]. RankSVM [7] incorporates a ranking loss function within the minimization function. The work of Chen et. al. [4] improves the RankSVM approach by explicitly including label correlations within the objective function using a hypergraph of interrelated proteins. Specifically, a hypergraph is defined where proteins are connected to each other if they share the same function. Since, the protein function databases like Gene Ontology [5] and FunCat [22] represent proteins within a hierarchy (or a directed acyclic graph), several approaches incorporate the parent-children relationship within the algorithm. The work of Barutcuoglu et. al. [1], first trains independent binary SVM classification models for each of the GO function classes, and then integrates the prediction by incorporating GO's hierarchical structure within a Bayes formulation. Pandey et. al. [19] defined a pairwise similarity term between the different GO labels and incorporated it within a weighted multi-label kNN classifier.

Using protein-protein interaction data, several semi-supervised multi-label learning algorithms have been developed. MCSL [10] uses an objective function that is similar to the local and global consistency method [31]. The approach in [10] incorporates a pairwise function correlation term within the regularization penalty resulting in a Kronecker matrix. To address the computational complexity associated with storing a Kronecker matrix in memory, PfunBG [9] uses an *undirected* bi-relational graph [28] to represent the relationships between proteins and functions. The bi-relational graph captures three types of relationships: (i) protein-protein similarities, (ii) function-function similarities and (iii) protein-function associations. The GRF method [30] uses Jaccard coefficients to measure pairwise function-function similarities, which are incorporated within a manifold regularization framework [2]. All the methods described, MCSL, PfunBG and GRF, utilize pairwise label correlations within a semi-supervised learning framework, but are developed for protein function prediction using a single data source only.

2.2 Data Integration in Protein Function Prediction

Several protein function prediction approaches that capture the complementary information associated with multiple heterogeneous data sources have been developed [17]. Within the kernel optimization framework, Lanckriet et. al. [12] represent each data source as an individual kernel and determine the optimal weighted combination of kernels by solving a semi-definite problem (SDP). This optimization does not scale with large number of proteins, and the work of Tsuda et.

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

al. [27] use a dual problem and gradient descent to determine the optimal combination of kernels. Shin et. al. [24] seek to determine weights using the EM algorithm [6], by iterating over reducing the prediction error and combining weights. Mostafavi et. al. [16] proposed a heuristic ridge-regression algorithm to combine multiple kernels. We refer the reader to a recent survey of multiple kernel learning methods [8].

The composite kernel determined as a linear combination of individual kernels obtained from the methods described above can be used within SVM-based or graph-based classifiers for protein function prediction. These methods determine the set of weights per function class, which results in increased time complexity. On the other hand, Mostafavi et. al. [15] proposed a method that simultaneously learns a set of weights for a group of correlated functions. This approach was called "Simultaneous Weighting (SW)". SW optimizes a set of weights for a group of functions, constructs a composite kernel for the functional groups, and then uses a graph-based semi-supervised classification scheme. Tsuda et. al. [27], Lewis [14] and Gönen et. al. [8] observe that a composite kernel combined with optimized weights has similar performance to a composite kernel combined with equal weights, i.e. without optimization.

In this paper we develop approaches that differ from traditional kernel integration methods [12, 15, 27]. Traditional methods first combine the kernels, and then annotate proteins using the composite kernels. In contrast, our approach first trains a transductive multi-label classifier (TMC) on each of the kernels representing a data source, and then integrates the predictions using an ensemble classification framework (TMEC). Our experimental evaluation demonstrates that, for the protein function prediction problem, TMEC outperforms methods that train classifiers on composite kernels. In addition, we observe that TMEC outperforms the TMC trained on an individual kernel. This result confirms that the ensemble approach is effective, and therefore the transductive multi-label classifiers trained on individual kernels are complementary to each other.

3. PROBLEM FORMULATION

We consider R different kinds of features that describe the same set of N proteins with C functions. Different kinds of features provide different representations for proteins (*e.g.* as vectors, trees, or networks). We assume that the first l proteins have known functions, and the remaining u proteins are not annotated ($l + u = N$). The R different sources of proteins are transformed into R kernels $[K_r]_{r=1}^{R}(K_r \in R^{N\times N})$, one kernel per source. Our objective is to first train a TMC on a directed bi-relation graph adapted from the kernel K_r, and then combine these TMCs into an ensemble classifier, thus giving TMEC. Finally, we use TMEC to annotate the u proteins. In this section, we first review the bi-relation graph and analyze its drawbacks when used for label propagation. Next, we propose a directed bi-relation graph and train TMC on this graph. We then define our TMEC on multiple kernels.

3.1 Transductive Multi-label Classification on a Directed Bi-relation Graph

Most graph-based multi-label learning methods explicitly incorporate a label correlation term into the general framework of graph-based semi-supervised learning to handle multi-labels [10, 30]. Unlike these methods, Wang *et al.* [28] design an undirected bi-relation graph for image classification and apply a random walk on it [25]. This graph includes both images and labels as nodes. For consistency, hereinafter, we use proteins instead of images and functions instead of labels. A bi-relation graph is composed of three kinds of edges: between proteins, between functions, and between proteins and functions. For the latter, if protein i has function c, an edge is set between them.

A random walk on a graph is often described by a transition probability matrix. For a random walk on a bi-relation graph, the transition probability matrix W is defined as:

$$W = \begin{bmatrix} \beta W_{PP} & (1-\beta)W_{PF} \\ (1-\beta)W_{FP} & \beta W_{FF} \end{bmatrix} \quad (1)$$

where W_{PP} and W_{FF} are the transition probability matrices of the intra-subgraphs of proteins and functions, respectively. W_{PF} and W_{FP} are the inter-subgraph transition probability matrices between proteins and functions, and β controls the relative importance of the intra- and the inter- subgraphs. W_{PP} and W_{FF} are computed as:

$$W_{PP} = D_{PP}^{-1} S_{PP} \quad W_{FF} = D_{FF}^{-1} S_{FF} \quad (2)$$

where $S_{PP} \in R^{N\times N}$ is the similarity matrix of proteins, and S_{FF} is the correlation matrix between functions. D_{PP} and D_{FF} are the diagonal matrices of the row sums of S_{PP} and S_{FF}, respectively. The correlation between functions can be defined in various ways [9, 19, 30]. Here we define the correlation between functions m and n using the cosine similarity as follows:

$$S_{FF}(m,n) = \frac{\mathbf{f}_m^T \mathbf{f}_n}{\|\mathbf{f}_m\| \|\mathbf{f}_n\|} \quad (3)$$

where $\mathbf{f}_m \in R^N (1 \le m \le C)$ is the m-th function vector on all proteins: if protein i has function m, then $\mathbf{f}_m(i) = 1$, otherwise $\mathbf{f}_m(i) = 0$. W_{PF} and W_{FP} are calculated as:

$$W_{PF} = D_{PF}^{-\frac{1}{2}} S_{PF} D_{FP}^{-\frac{1}{2}} \quad W_{FP} = D_{FP}^{-\frac{1}{2}} S_{FP} D_{PF}^{-\frac{1}{2}}, \quad (4)$$

where S_{PF} is the relation matrix between proteins and functions and S_{FP} is the transpose of S_{PF}. D_{PF} is the diagonal matrix of the column sums of S_{PF} and D_{FP} is the diagonal matrix of the row sums of S_{PF}. We observe that if protein i has function c then $S_{PF}(i,c) = 1$; otherwise $S_{PF}(i,c) = 0$.

Similarly to hypergraph-based multi-label learning [4], also in the bi-relation graph the c-th function node and the proteins annotated with this function are considered as a group:

$$G_c = v_c^F \cup \{v_i^P | S_{PF}(i,c) = 1\} \quad (5)$$

where v_c^F is the c-th function node and v_i^P is the i-th protein node of the bi-relation graph. In the bi-relation graph, instead of computing the node-to-node relevance between a function node and an unannotated protein node, the relevance between a protein and a group G_c is considered. For the c-th function, the distribution vector $\tilde{Y}_c$ is:

$$\tilde{Y}_c = \begin{bmatrix} \gamma \tilde{Y}_c^P \\ (1-\gamma)\tilde{Y}_c^F \end{bmatrix} \in R^{N+C} \quad (6)$$

where $\tilde{Y}_c^P(i) = \frac{1}{\sum_{i=1}^{N} S_{PF}(i,c)}$ if $S_{PF}(i,c) = 1$ and $\tilde{Y}_c^P(i) = 0$ otherwise; $\tilde{Y}_c^F(j) = 1$ if $j = c$, and $\tilde{Y}_c^F(j) = 0$ otherwise. γ controls the probability for the random walker to jump from a protein subgraph to a function subgraph.

Based on these preliminaries, an iterative objective function, also applied in our TMC, is defined on this bi-relation graph as follows:

$$F^{(t+1)}(j) = (1-\alpha)\sum_{i=1}^{N+C} F^{(t)}(i)W(i,j) + \alpha\tilde{Y}_j \qquad (7)$$

where $F^{(t)}(j)$ is the predicted likelihood score vector of the j-th protein in the t-th iteration, α is a scalar value to balance the tradeoff between the initial function set and the predicted function set. From Eq. (7), we can see that the functions of a protein are predicted by the functions of its connected nodes. This makes Eq. (7) a direct protein function prediction method [23].

However, the application of Eq. (7) for protein function prediction has a major drawback. Suppose i is a protein vertex annotated with a function vertex j. The function j may be overwritten by the functions of the proteins connected to i, thus causing the loss of reliable information. As such, the functions of initially annotated proteins may be changed in the iterative label propagation. This phenomenon is similar to the one occurring in the local and global consistency method [31], and should be avoided.

A formal analysis of the above phenomenon is as follows. For simplification, the parameter β in Eq. (1) is not included in the following. Eq. (7) can be rewritten as follows:

$$\begin{bmatrix} F_P^{(t+1)} \\ F_F^{(t+1)} \end{bmatrix} = (1-\alpha)\begin{bmatrix} W_{PP} & W_{PF} \\ W_{FP} & W_{FF} \end{bmatrix}\begin{bmatrix} F_P^{(t)} \\ F_F^{(t)} \end{bmatrix} + \alpha\begin{bmatrix} \tilde{Y}_P \\ \tilde{Y}_F \end{bmatrix}$$

$$= (1-\alpha)\begin{bmatrix} W_{PP}F_P^{(t)} + W_{PF}F_F^{(t)} \\ W_{FP}F_P^{(t)} + W_{FF}F_F^{(t)} \end{bmatrix} + \alpha\begin{bmatrix} \tilde{Y}_P \\ \tilde{Y}_F \end{bmatrix}$$

$$F_P^{(t+1)} = (1-\alpha)(W_{PP}F_P^{(t)} + W_{PF}F_F^{(t)}) + \alpha\tilde{Y}_P \qquad (8)$$

$$F_F^{(t+1)} = (1-\alpha)(W_{FP}F_P^{(t)} + W_{FF}F_F^{(t)}) + \alpha\tilde{Y}_F \qquad (9)$$

From Eqs. (8-9), we can see that W_{PF} propagates function information from function to protein nodes, and W_{FP} propagates function information from proteins to function nodes. In a bi-relation graph, we expect that information propagates from function to protein nodes, but not vice versa. Thus, we change the undirected bi-relation graph into a directed one. The resulting directed bi-relation graph is shown in Figure 1. In this graph, information can be propagated in the intra-subgraphs W_{PP} and W_{FF}, and in the inter-subgraph W_{PF}, but not in the inter-subgraph W_{FP}. Therefore, we define a new transition probability matrix W_d on the directed bi-relation graph as follows:

$$W_d = \begin{bmatrix} W_{PP} & W_{PF} \\ \mathbf{0} & W_{FF} \end{bmatrix} \qquad (10)$$

where $\mathbf{0} \in R^{C\times N}$. In the case where proteins have incorrect annotations (i.e., noisy labels), it will be advantageous to use undirected relationships between proteins and functions.

Based on Eq. (7), setting $F^{(1)} = \tilde{Y}$, we have:

$$F^{(t+1)} = ((1-\alpha)W_d)^{(t)}\tilde{Y} + \alpha\sum_{k=1}^{t}((1-\alpha)W_d)^k\tilde{Y} \qquad (11)$$

Since $0 < \alpha < 1$ and $0 \le (1-\alpha)W_d < 1$, the first term in Eq. (11) is bound to 0, and the second term (excluding $\tilde{Y}$) is a geometric series with the following limit:

$$\lim_{t\to\infty}\sum_{k=1}^{t}((1-\alpha)W_d)^k = (I-(1-\alpha)W_d)^{-1} \qquad (12)$$

where $I \in R^{(N+C)\times(N+C)}$ is the identity matrix. Thus the equilibrium solution F of Eq. (7) is:

$$F = \alpha(I-(1-\alpha)W_d)^{-1}\tilde{Y} \qquad (13)$$

The predicted $F(j)$ is a real value vector with size C, where each entry reflects the likelihood that protein j has the corresponding function. Thus, we also refer to $F(j)$ as the predicted likelihood score vector of protein j. From Eq. (13), we can see that F is determined by W_d and a well-structured bi-relation graph can produce a competent F.

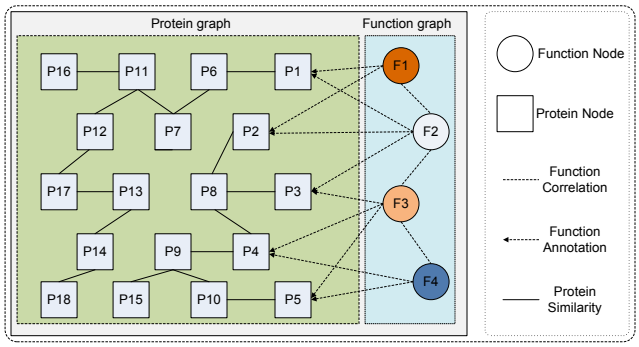


Figure 1: Directed bi-relation graph

3.2 Transductive Multi-label Ensemble Classification

TMC avoids the risk of overwriting the information given by function nodes. However, because of the noisy edges and isolated proteins present in a single bi-relation graph, it is still limited in providing a confident likelihood score vector $F(j)$ from a single data source. To avoid this limitation, we can leverage the various graphs (or kernels) associated with the same set of proteins (*e.g.*, PPI network, gene interaction network, and co-participation network in a protein complex) [17, 24]. These graphs are, to some extent, independent to one another, and also carry partially complementary information. Some researchers have advocated integrating such multiple kernels into a composite kernel [15, 16, 17]. The experimental results showed the annotator trained on the composite kernel has a superior likelihood score vector than the annotator trained on the single kernel. But the classifiers used on the composite kernel are binary and cannot make use of the correlation between the functions.

Here we predict protein functions using multiple kernels derived from multiple sources by performing *classifiers integration*. More specifically, we first transform each kernel into a directed bi-relation graph. We then train a TMC on each of the graphs. Finally, we combine these TMCs into a transductive multi-label ensemble classifier, TMEC. TMEC is described in **Algorithm 1**. In Eq. (14), we combine the F_r values using a weighted majority vote. The motivation is that different kernels have different qualities and have different levels of confidence on the predicted functions of a protein. For example, if kernel K_1 is more

confident on annotating protein i with function m, and K_2 is more confident on annotating protein i with function n, then K_1 will have more influence on determining the m-th function of protein i, and K_2 will have more influence on determining the n-th function of protein i.

Algorithm 1 TMEC: Transductive Multi-label Ensemble Classification

Input:

$\{K_r\}_{r=1}^R$ from R data sources
$Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_l]$
α, β, γ

Output:

Predicted likelihood score vectors $\{F(j)\}_{j=l+1}^N$

1: Specify $\widetilde{Y}$ using Eq.(6)
2: **for** $r = 1$ to R **do**
3: Set $W_{PP} = K_r$ and construct a directed bi-relation graph using Eqs.(2-4) and Eq.(10)
4: Get the rth annotator F_r using Eq.(13)
5: **end for**
6: Ensemble the R annotators $\{F_r\}_{r=1}^R$ as:

$$F(j) = \frac{1}{R}\sum_{r=1}^{R} F_r(j) \tag{14}$$

Due to the different structures across different kernels, the base classifiers F_r in Eq. (14) are diverse. In addition, because of the complementary information between different kernels, the predicted likelihood score vectors $F_r(j)$ are also complementary to each other. In contrast, the annotator trained on the composite graph cannot make use of these predicted likelihood score vectors. In ensemble learning, diversity between base classifiers is paramount to gain a consensus classifier with a good generalization ability [11]. For these reasons, TMEC can annotate proteins with higher confidence than the annotator trained on the composite kernel. Our experimental results in Section 5 confirm this advantage. An additional merit of TMEC is that it does not require to have all the data sources available beforehand, and each TMC can be trained individually or sequentially. In addition, new data sources can be appended into TMEC without repeating the entire training process.

4. EXPERIMENTAL SETUP

4.1 Dataset Description

We evaluate the performance of our algorithm on two previously defined protein function prediction benchmarks. Both these datasets provide heterogeneous protein information for the yeast (*Saccharomyces cerevisiae*) organism. The first dataset obtained from Tsuda *et. al.* [27] [2] consists of 3588 proteins, annotated with 13 highest-level functional classes from the MIPS Comprehensive Yeast Genome Data (MIPS CYGD)[3]. This dataset, called **yeast1** here, represents pairwise protein similarities as five different kernel functions. These kernels are obtained from the following sources: Pfam domain structure (K_1), co-participation within protein complexes (K_2), PPI network (K_3), genetic interaction network (K_4), and cell cycle and gene expression measurements (K_5). We refer the reader to the original paper [27] for more details.

The second dataset was obtained from the study by Mostafavi *et. al.* [15][4]. This dataset includes 3904 proteins annotated with 1188 protein functions, according to the biological process terms of the Gene Ontology [5]. The dataset provides 44 different kernel functions, most of which are protein-protein interactions obtained from different experiments [15]. As done in the previous study [15], we filtered the dataset to include only those GO functions that had at least 100 proteins and at most 300 proteins. This resulted in a dataset containing 1809 proteins annotated with 57 different functions. We refer to this dataset as **yeast2**.

4.2 Evaluation Metrics

We evaluate the protein function prediction problem as a multi-label classification problem. Given C different protein functions, our approach results in a predicted likelihood vector that assigns a protein j to a function c with probability $F(j, c)$. To convert the probabilistic assignment into a hard function assignment, we choose the top k most probable function assignments. For each protein, the k largest likelihood scores are chosen as relevant functions [3, 30].

We compute the Macro $F1$ score (MacroF1) across all the protein functions:

$$MacroF1 = \frac{1}{C}\sum_{c=1}^{C}\frac{2p_c r_c}{p_c + r_c}$$

where p_c and r_c are the precision and recall of the c-th function.

We also compute the Ranking loss. Ranking loss evaluates the average fraction of function label pairs that are not correctly ordered:

$$RankingLoss = \frac{1}{u}\sum_{i=l+1}^{N}\frac{1}{|\mathbf{y}_i||\bar{\mathbf{y}}_i|}|\{(y_1, y_2) \in \mathbf{y}_i \times \bar{\mathbf{y}}_i | F(i, y_1) \leq F(i, y_2)\}|$$

where $\mathbf{y}_i$ is the function set of protein i, and $\bar{\mathbf{y}}_i$ is the complementary set of $\mathbf{y}_i$. The performance is perfect when $RankingLoss = 0$, and the smaller the value, the better the performance. To keep consistency with $MacroF1$, we use $1 - RankingLoss$ instead of $RankingLoss$.

5. EXPERIMENT ANALYSIS

In this section, we evaluate our TMC and TMEC by comparing them against PfunBG [9], GRF [30] and SW [15]. PfunBG and GRF are recently proposed semi-supervised multi-label classifiers based on PPI networks for protein function prediction. SW is a recently introduced efficient protein function prediction method based on composite kernels. TMC and PfunBG have a similar objective function. The main difference between them is that TMC is trained on the *directed* bi-relation graph and PfunBG is trained on the *undirected* bi-relation graph. Here, we view function and protein nodes as equal, and set the parameters β and γ in the bi-relation graph equal to 0.5. Similarly to the local and global consistency method [31] and PfunBG [9], α in Eq. (7) is set to 0.01.

[2] http://www.cbrc.jp/ tsuda/code/eccb05.html
[3] http://mips.helmholtz-muenchen.de/genre/proj/yeast/
[4] http://morrislab.med.utoronto.ca/ sara/SW/

Yeast1 has 5482 annotations on 3588 proteins, thus on average each protein is annotated with 1.52 functions. Among the 3588 proteins, 3489 (97%) proteins have $\leq$3 functions, so we evaluate all the methods on yeast1 with $k \in \{1, 2, 3\}$. Yeast2 has 7874 functions on 1809 proteins, thus on average each protein has 4.35 functions. Among the 1809 proteins, 1333 proteins (73.69%) have $\leq$5 functions and 1656 proteins (91.54%) have $\leq$10 functions. Therefore, we evaluate all methods on yeast2 with k from 1 to 10.

As the authors reported in [14] and [27], the composite kernel combined with different weights has similar performance to the composite kernel combined with equal weights. Therefore, for simplicity, TMC, PfunBG and GRF are all trained on the composite kernel combined with equal weights. In addition, we take the recently proposed SW, which gives different weights to different kernels, as a baseline method. In Subsection 5.2, we investigate the performance of TMC on each single kernel. In the following experiments, unless otherwise specified, all the results are the average of 100 runs. In each run, 80% of the proteins are randomly selected for training, and the remaining 20% are used for testing.

5.1 Directed Bi-relation Graph vs. Undirected Bi-relation Graph

To investigate the difference between directed and undirected bi-relation graphs, we compare our TMC against PfunBG and GRF on the composite kernel, and TMEC against ensemble PfunBG (PfunBG-MK) and ensemble GRF (GRF-MK) on yeast1 and yeast2. The base classifiers of PfunBG-MK and GRF-MK on these 5 kernels are combined in the same way as TMEC in Eq. (14). Table 1 and Table 2 list the corresponding macro F1 scores and RankingLoss on the composite kernel and multiple kernels of yeast1. Table 3 reports the RankingLoss on the composite kernel of yeast2, and Table 4 records the RankingLoss on the multiple kernels of yeast2. Figure 2 plots macro F1 scores on the composite kernel and multiple kernels of yeast2. In these tables, results reported in boldface are significantly better, with significance level 95% in 100 runs using paired t-test. Standard deviations are also reported. The same t-test is also used to analyze the results in Figure 2. In these figures, the highest bar in each index indicates a significantly better result.

From these tables and figure, we can observe that, in most cases, TMC trained on the directed bi-relation graph is capable of achieving a higher performance than PfunBG trained on the undirected bi-relation graph. These results show the advantage of using a directed bi-relation graph. SW takes advantage of regression to seek the optimal combining weights, and binary classification to predict protein functions; it achieves a higher RankingLoss on yeast1, but it loses to PfunBG and TMC on the macro F1 score. As to yeast2, TMC significantly outperforms the other three methods on MacroF1 and RankingLoss. The ensemble classifiers are capable of achieving a better performance than the corresponding classifiers trained on the composite kernel. This result supports the use of multi-label ensemble classification for protein function prediction.

5.2 Multi Kernels vs. Single Kernel

In this section, we conduct additional experiments to investigate the difference between the TMC on the composite kernel, the TMC on a single kernel from one data source, and the TMEC on multiple kernels. Since the average number of functions of a protein is 1.52 in yeast1 and 4.35 in yeast2, we set $k = 2$ on yeast1 and $k = 5$ on yeast2. The results are plotted in Figures 3 and 4. The first two bars in each figure represent TMEC and TMC on the composite kernel; the remaining bars describe the results of TMC trained on a single kernel. The highest bar in each index indicates that the corresponding result is significantly better than the others bars for that index.

From Figures 3 and 4, we can observe that TMC trained on the composite kernel always outperforms TMC trained on a single kernel. This can be attributed to the complementary information across different kernels. TMEC performs better than TMC trained on the single kernel, and it outperforms TMC trained on the composite kernel (except in Figure 3(b)). The explanation is that TMEC not only takes advantage of the complementary information between different kernels, but also makes use of the structural difference among different kernels and the complementary likelihood score vectors. These results support the benefits of *classifier integration* over *data integration*.

5.3 Parameter Sensitivity Analysis

TMC and TMEC have three parameters: α, β and γ. α is a parameter common to label propagation methods and often is given a small value. In this section, we analyze the sensitivity of our methods with respect to β and γ. We vary β and γ between 0.1 and 0.9 with step size 0.1, and record the macro F1 score and Ranking loss. The experimental results are given in Figures 5 and 6.

From these figures, we can observe that TMEC is less sensitive than TMC to parameter selection. TMEC has wider ranges of effective parameter values than TMC. We also computed the maximum, minimum, average, and median values of MacroF1 and RankingLoss for the different values of β and γ. Due to space limit, we do not report them here. With respect to these four statistics, the values of MacroF1 and RankingLoss of TMEC are always better than that of TMC. This result again supports the advantage of *classifier integration* over *data integration*.

6. CONCLUSIONS

In this paper, we analyze the drawback of using undirected bi-relation graphs for protein function prediction. To avoid this limitation, we propose to use a directed bi-relation graph, and define a TMC on it. We further improve the performance by combining various TMCs trained on multiple data sources (TMEC). Different from traditional methods that make use of multiple data sources by data integration, TMEC takes advantage of multiple data sources by classifier integration. TMEC does not require to collect all the data sources beforehand. Our experimental results show that classifier integration is a valuable methodology to leverage multiple biological data sources.

The experimental results show that different kernels have different levels of quality. We will investigate a scheme that gives optimal weights to multi-label classifiers trained on different kernels, and then combine such classifiers using the weights for improved protein function prediction.

7. ACKNOWLEDGEMENT

This paper is partially supported by grants from NSF IIS 0905117, Natural Science Foundation of China (Project Nos.

Table 1: MacroF1 and 1-RankingLoss on composite kernel of yeast1.

Metric	MacroF1			1-RankingLoss
Method	$k=1$	$k=2$	$k=3$	
SW	35.24 ± 1.25	45.08 ± 1.76	46.05 ± 1.08	$\mathbf{84.35 \pm 0.73}$
GRF	44.81 ± 1.50	49.53 ± 1.31	45.89 ± 1.36	80.79 ± 1.03
PfunBG	45.44 ± 1.52	49.85 ± 1.26	47.32 ± 1.54	80.76 ± 0.84
TMC	$\mathbf{47.66 \pm 1.45}$	$\mathbf{50.22 \pm 1.25}$	47.17 ± 1.03	80.18 ± 0.93

Table 2: MacroF1 and 1-RankingLoss on multiple kernels of yeast1.

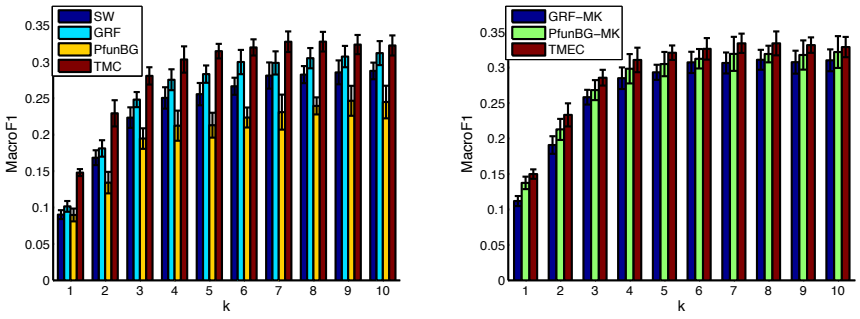
Metric	MacroF1			1-RankingLoss
Method	$k=1$	$k=2$	$k=3$	
GRF-MK	45.99 ± 1.59	50.65 ± 1.40	46.70 ± 1.30	80.93 ± 0.99
PfunBG-MK	47.79 ± 1.62	51.42 ± 1.27	47.40 ± 1.25	80.50 ± 0.87
TMEC	$\mathbf{49.75 \pm 1.60}$	$\mathbf{51.98 \pm 1.19}$	47.11 ± 1.01	80.79 ± 1.01

Table 3: 1-RankingLoss on composite kernel of yeast2.

Metric	1-RankingLoss			
Method	SW	GRF	PfunBG	TMC
	74.38 ± 1.42	71.66 ± 1.22	64.14 ± 0.89	$\mathbf{79.85 \pm 0.84}$

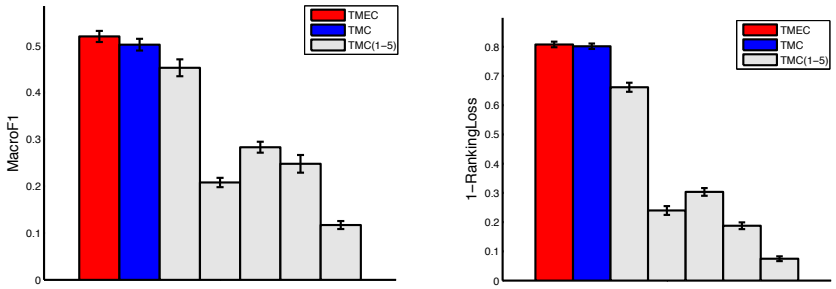
Table 4: 1-RankingLoss on multiple kernels of yeast2.

Metric	1-RankingLoss		
Method	GRF-MK	PfunBG-MK	TMEC
	72.97 ± 1.28	76.50 ± 1.28	$\mathbf{80.44 \pm 0.88}$



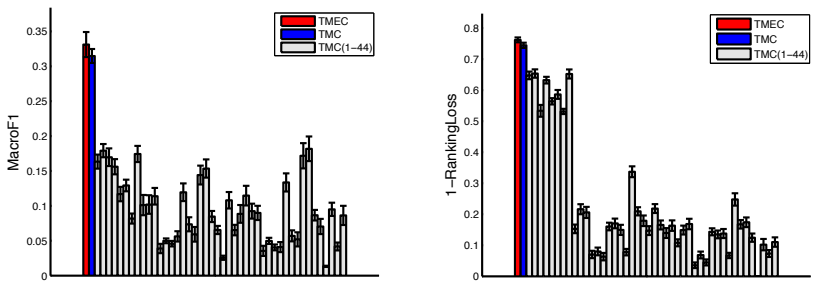
(a) composite kernel (b) multiple kernels

Figure 2: MacroF1 on the composite kernel and multiple kernels of yeast2.



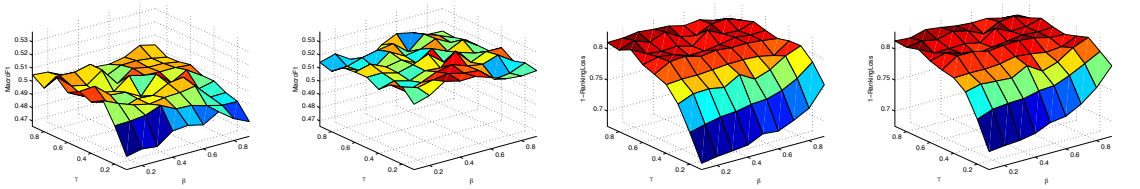
(a) MacroF1 (b) 1-RankingLoss

Figure 3: Difference between multiple kernels, composite kernel and single kernels of yeast1. TMC(1-5) denotes TMC on the single kernels of yeast1



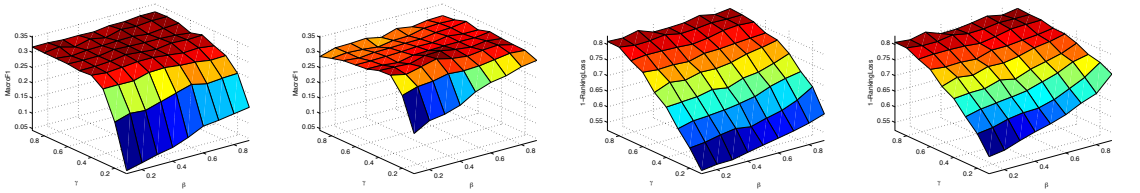
(a) MacroF1 (b) 1-RankingLoss

Figure 4: Difference between multiple kernels, composite kernel and single kernels of yeast2. TMC(1-44) denotes TMC on the single kernels of yeast2.



(a) TMC (MacroF1) (b) TMEC (MacroF1) (c) TMC (RankingLoss) (d) TMEC (RankingLoss)

Figure 5: MacroF1 and 1-RankingLoss on different β and γ (yeast1).



(a) TMC(MacroF1) (b) TMEC(MacroF1) (c) TMC(RankingLoss) (d) TMEC(RankingLoss)

Figure 6: MacroF1 and 1-RankingLoss on different β and γ (yeast2).

60973083, 61070090, 61003174 and 61170080), grants from the NSFC-Guangdong Joint Fund (Project No. U1035004 and U1135004) and China Scholarship Council (CSC).

8. REFERENCES

[1] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[2] 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.

[3] P. Bogdanov and A. Singh. Molecular function prediction using neighborhood features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(2):208–217, 2010.

[4] G. Chen, J. Zhang, F. Wang, C. Zhang, and Y. Gao. Efficient multi-label classification with hypergraph regularization. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1658–1665. IEEE, 2009.

[5] G. O. Consortium et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[7] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, 14:681–687, 2001.

[8] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[9] J. Jiang. Learning protein functions from bi-relational graph of proteins and function annotations. *Algorithms in Bioinformatics*, pages 128–138, 2011.

[10] J. Jiang and L. McQuay. Predicting protein function by multi-label correlated semi-supervised learning. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, (99):1–1, 2011.

[11] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[12] G. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[13] C. Leslie, E. Eskin, A. Cohen, J. Weston, and W. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467, 2004.

[14] D. Lewis. *Combining kernels for classification*. PhD thesis, Columbia University, 2006.

[15] S. Mostafavi and Q. Morris. Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, 26(14):1759–1765, 2010.

[16] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. Genemania: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(Suppl 1):S4, 2008.

[17] W. Noble and A. Ben-Hur. Integrating information for protein function prediction. *Bioinformatics-From Genomes to Therapies*, pages 1297–1314, 2007.

[18] 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.

[19] G. Pandey, C. Myers, and V. Kumar. Incorporating functional inter-relationships into protein function prediction algorithms. *BMC Bioinformatics*, 10(1):142, 2009.

[20] P. Pavlidis, J. Weston, J. Cai, and W. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.

[21] M. Re and G. Valentini. Ensemble based data fusion for gene function prediction. *Multiple Classifier Systems*, pages 448–457, 2009.

[22] 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.

[23] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(1), 2007.

[24] H. Shin, K. Tsuda, and B. Schölkopf. Protein functional class prediction with a combined graph. *Expert Systems with Applications*, 36(2):3284–3292, 2009.

[25] H. Tong, C. Faloutsos, and J. Pan. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346, 2008.

[26] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

[27] K. Tsuda, H. Shin, and B. Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(suppl 2):ii59, 2005.

[28] H. Wang, H. Huang, and C. Ding. Image annotation using bi-relational graph of images and semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 793–800. IEEE, 2011.

[29] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[30] X. Zhang and D. Dai. A framework for incorporating functional inter-relationships into protein function prediction algorithms. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, (99):1–1, 2011.

[31] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–328, 2004.