From Protein Sequence to Protein Function via Multi-Label Linear Discriminant Analysis

Hua Wang, Lin Yan, Heng Huang, and Chris Ding

Abstract—Sequence describes the primary structure of a protein, which contains important structural, characteristic, and genetic information and thereby motivates many sequence-based computational approaches to infer protein function. Among them, feature-base approaches attract increased attention because they make prediction from a set of transformed and more biologically meaningful sequence features. However, original features extracted from sequence are usually of high dimensionality and often compromised by irrelevant patterns, therefore dimension reduction is necessary prior to classification for efficient and effective protein function prediction. A protein usually performs several different functions within an organism, which makes protein function prediction a *multi-label classification* problem. In machine learning, multi-label classification deals with problems where each object may belong to more than one class. As a well-known feature reduction method, linear discriminant analysis (LDA) has been successfully applied in many practical applications. It, however, by nature is designed for *single-label classification*, in which each object can belong to exactly one class. Because directly applying LDA in multi-label classification causes ambiguity when computing scatters matrices, we apply a new Multi-label Linear Discriminant Analysis (MLDA) approach to address this problem and meanwhile preserve powerful classification capability inherited from classical LDA. We further extend MLDA by ℓ_1-normalization to overcome the problem of over-counting data points with multiple labels. In addition, we incorporate biological network data using Laplacian embedding into our method, and assess the reliability of predicted putative functions. Extensive empirical evaluations demonstrate promising results of our methods.

Index Terms—Protein function prediction, multi-label classification, linear discriminant analysis

1 Introduction

Sequence is the most fundamental form to describe a protein since it determines different characteristics of the protein such as its sub-cellular localization, structure and function. As a result, protein sequences have been heavily utilized to develop *in silico* approaches for automatic function prediction, which can be broadly categorized into the following three classes [1].

Homology-Based Approaches. Have demonstrated their usefulness together with the success of the sequence-sequence comparison systems such as FASTA [2] and Basic Local Alignment Search Tool (BLAST) [3], [4]. However, because a duplicate of an gene could adopt a new function in response to selective pressure during evolution [5], function transfer by homology on such gene and its product could produce erroneous results [6].

Subsequence-Based Approaches. Focus on seeking the most informative segments in protein sequences, such as motifs [7] and functional domains [8], because often only specific parts of a whole sequence are crucial for the protein to perform its functions. Treating these subsequences as features of a protein, these approaches construct models for the mapping of the features to protein functions and use the trained model to predict the function of a query protein [9], [10], [11].

Feature-Based Approaches. Subsequence-based approaches predict protein function from protein sequences in their raw form, i.e., as a string of characters. However, it is possible to transform these sequences into more biologically meaningful features, which makes it easier to distinguish between proteins from different functional classes [12]. With this recognition, feature-based approaches use standard classification algorithms to learn models of functional classes from the transformed set of features, and then utilize these models to make predictions for uncharacterized proteins [13], [14], [15], [16].

Adopting the perspective of feature-based approaches, in this paper we learn from protein sequences a prediction model to transform the extracted sequence features into a discriminative subspace, in which the classification, i.e., function prediction, can be carried out more effectively with less computational complexity.

2 Materials and Methods

Predicting protein function from sequence involves two types of data, i.e., protein sequences and the corresponding function annotations. Therefore, we first formulate these two types of data and formalize the protein function prediction problem.

2.1 Construction of Feature Vector for Protein Sequence

Sequence is a string of amino acids and depicts the primary structure of a protein. Traditional computational approaches make use of sequences to assess the similarity among proteins via sequence alignment algorithms, which attempt to match the amino acid strings to satisfy certain criteria: either

- *H. Wang is with the Department of Electrical and Computer Engineering, Colorado School of Mines, Golden, CO 80401. E-mail: yy.huawang@gmail.com.*
- *L. Yan, H. Huang, and C. Ding are with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019. E-mail: lin.h.yan@outlook.com, {heng, chqding}@uta.edu.*

Manuscript received 10 May 2015; revised 6 Oct. 2015; accepted 2 Nov. 2015. Date of publication 14 July 2016; date of current version 1 June 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2016.2591529

locally or globally. Numerous pairwise sequence alignment algorithms [17] and multiple sequence alignment algorithms [18] have been devised for each criterion. However, we have not yet notice any literature addressing the issue which criterion and which algorithm is optimal for protein function prediction. Moreover, genes evolve at different rates due to both uneven selection pressure on their functions and the inherent mutation rate of different species, which means that it is difficult to establish a similarity measure that is reliable in all cases. Rodents, for example, accumulate point mutations more rapidly than apes, and the evolutionary rates of proteins in different gene families may vary by several orders of magnitude. Therefore, independent from using sequence alignment, a more robust and effective way to quantify sequence is expected.

Considering the fact that a protein sequence is a string of characters (amino acids), we may use the *bag-of-words* model [19] in information retrieval to extract sequence features from the statistical point of view. In this model, a text, such as a sentence or a document, is represented as an collection of words. In the same way, a protein sequence can also be represented by a set of predefined terms. In the context of sequence analysis, k-mers are the most natural term set and largely used in many biological applications. k-mers consider k consecutive nucleotides (in DNA) or amino acids (in protein) as a unit, and their frequency, also called as distribution, are often used to characterize sequences. In this work, we use trimer ($k = 3$) as a descriptor for protein sequence which treats any three consecutive amino acids as a term.

Using trimers as terms, protein function prediction is analogous to document query, where a protein sequence is equivalent to a document in a text collection (all the protein sequences). We use term frequency-inverse document frequency (*tf-idf*) weight [20] to build the vector descriptions, $\mathbf{x}_i \in \mathbb{R}^p$, for proteins. This weight is broadly used in information retrieval, and statistically evaluates how important a term is with respect to a document in a collection.

Because there are 20 amino acids to constitute protein sequences, the dimensionality of x_i is $p = 20^3$. We sort the trimers in the standard amino acid order, and denote $\mathbf{tf}_{i,j}$ as the term frequency of the jth trimer appearing in the ith protein sequence. For example, because "Q" is the 7th amino acid and "S" is the 16th one, "QQS" is the $(7-1) \times 20 \times 20 + (7-1) \times 10 + 16 = 2,536$th trimer and its term frequency in the ith protein is denoted as $\mathbf{tf}_{i,2,356}$. Given a sample sequence as "AANEQQSANEQQSN", $\mathbf{tf}_{i,2,356} = 2$. Clearly, the value of $\mathbf{tf_{i,j}}$ correlates to the length (the total number of amino acids) of a protein. In general, a long protein would have a large value of $\mathbf{tf_{i,j}}$, which complicates the comparison between two different proteins. We hence normalize $\mathbf{tf_{i,j}}$ to solve this problem as following:

$$\mathbf{ntf}_{i,j} = \frac{\mathbf{tf_{i,j}}}{\sum_j \mathbf{tf_{i,j}}}, \tag{1}$$

where $\sum_j \mathbf{tf_{i,j}}$ computes the number of occurrences of all trimers in the ith protein sequence. Therefore, $\mathbf{ntf}_{i,j}$ measures the importance of a trimer in one individual protein, which, though, still suffers from a critical problem: all trimers are considered equally important when measuring pairwise protein similarities. In fact, however, certain terms have little or no discriminating power in determining relevance, which is same as that in document query. For instance, the stop words, such as "the" and "this", almost appear in every document in a collection, from which we can not determine the document category. In contrast, specific terms, such as "stock", usually only appear in the documents related to finance in the collection and is clearly more useful for category determination. To this end, we use inverse document frequency to measure the general importance of a trimer in the whole sequence data collection as

$$\mathbf{idf}_j = \log \frac{n}{\mathbf{df}_j}, \tag{2}$$

where n is the number of all the proteins, and $\mathbf{df}_j$ is the number of proteins in which the jth trimer appears (i.e., the number of proteins with $\mathbf{tf_{i,j}} \neq 0$). Finally, the tf-idf weight is defined as

$$\mathbf{tf-idf}_{i,j} = \mathbf{tf_{i,j}} \times \mathbf{idf}_j, \tag{3}$$

which is the jth component of $\mathbf{x}_i$, i.e., $\mathbf{x}_i(j) = \mathbf{tf-idf}_{i,j}$.

The protein sequence data used in this work are obtained from GenBank [21].

TABLE 1
Main Functional Categories in FunCat Annotation Scheme (Version 2.1) and the Corresponding Number of Annotated Proteins to Yeast Species

ID	Function Description	Size
01	Metabolism	1,397
02	Energy	336
10	Cell Cycle and DNA Processing	981
11	Transcription	1,009
12	Protein Synthesis	476
14	Protein Fate	1,125
16	Protein with Binding Function	1,019
18	Regulation of Metabolism and Protein Function	246
20	Transport Facilitation and Transport Routes	995
30	Cellular Communication and Signal Transduction	231
32	Cell Rescue, Defense and Virulence	515
34	Interaction with the Environment	446
38	Transposable Elements, Viral and Plasmid Proteins	59
40	Cell Fate	268
41	Development	67
42	Biogenesis of Cellular Components	827
43	Cell Type Differentiation	437

2.2 Function Annotation Data

We first use functional catalogue (FunCat) [22] for protein function annotation. FunCat is an annotation scheme for the functional description of proteins from prokaryotes, unicellular eukaryotes, plants and animals. Taking into account the broad and highly diverse spectrum of known protein functions, FunCat (version 2.1) consists of 27 main functional categories that cover general fields such as cellular transport, metabolism, cellular communication, etc. The main branches exhibit a hierarchical and tree-like structure with up to six levels of increasing specificity. 17 main function categories in FunCat annotation scheme are involved in annotating yeast genome as listed in Table 1. Together with the sequence data, we end up with 4,403 annotated proteins and 1,988 unannotated proteins for *Saccharomyces cerevisiae* species.

Besides, we also use Gene ontology (GO) [23] to annotate the proteins. GO is a major bioinformatics initiative to unify

the representation of gene and gene product attributes across all species, whose annotation categories covers three domains: cellular component, molecular function, and biological process. Following [24], we use the functional terms in molecular function and biological process.

2.3 Problem Formalization and Multi-Label Protein Function Prediction

Using protein sequences and the corresponding function annotations as input, we formalize protein function prediction as a classification problem with n training data points, m test data points and K target classes. A protein is characterized as a data point and the functions are treated as target classes. Each data point $\mathbf{x_i}$ is associated with a set of labels represented by a binary indicator vector $\mathbf{y}_i \in \{0,1\}^K$ such that $\mathbf{y}_i(k) = 1$ if data point $\mathbf{x_i}$ belongs to the kth class, and 0 otherwise. Given a labeled data set $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, the goal is to predict labels for the unlabeled data points $\{\mathbf{x_i}\}_{i=n+1}^{m+n}$. We write $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T$.

Necessity of Feature Reduction on Input Data. By formalizing protein function prediction as a classification problem, many classification algorithms in machine learning can be applied to infer protein functions. However, two problems prevent us from directly using $\mathbf{x_i}$ for classification. First, the dimensionality of the input data, $p = 20^3$, is relatively high, which significantly increases the computational complexity of the classification algorithms and makes the classification tasks computationally intractable due to "the curse of dimensionality" [25]. Second and more important, not all the features (trimers in current classification problem) are necessary for the classification. Same as classification problems in many other applications such as information retrieval and data mining, the class memberships only correlate to some patterns of much lower dimensionality hidden in the original data, and many of the features are irrelevant and sometimes even harmful. Therefore, feature reduction to reduce dimensionality and prune irrelevant information is necessary prior to classification. Among various feature reduction methods in statistical learning, Linear Discriminant Analysis [25] is a well known and widely used method to learn a *discriminative* transformation from the original high-dimensional space to a subspace with desired low dimensionality, in which the input data points from different classes are well separated.

Multi-Label Function Prediction and Its Difficulties to Use LDA. Because different regions of a protein sequence have different structural and functional characteristics, a protein usually performs multiple functions [8]. Therefore protein function prediction is a *multi-label classification* problem [26], [27]. Multi-label classification is an emerging topic in machine learning driven by the advances of modern technologies in the past two decades, in which each object may belongs to more than one classes. Therefore, although LDA has been applied successfully in many applications, it can not be directly used to predict protein function, because it is by nature a *single-label classification* method. Single-label classification refers to the traditional classification tasks in machine learning where each object belongs to exactly one class. The main difficulty to apply classical LDA to multi-label classification is how to measure the inter and intra class scatters. In single-label classification, data points are exclusively partitioned into several groups, hence the data scatters are naturally measured by the geometrical dispersion of the data points in the hyperspace. However, in multi-label case, because the partitions of data points overlap from one another, one data point could belong to different classes. Therefore, how much a data point with multiple labels should contribute to the between-class and within-class scatters remains unclear. By recognizing this, in this paper we apply the Multi-label Linear Discriminant Analysis (MLDA) approach proposed in our earlier work [26] to solve this problem by constructing the scatter matrices from class perspective, such that the scatter matrices are explicitly defined. We further extend MLDA by ℓ_1-normalization to overcome the problem of over-counting the data points with multiple labels in scatter matrices calculation.

Framework to Predict Protein Function Using MLDA. Using MLDA, we learn a transformation $U^{p\times r}$ from the training data $\{(\mathbf{x_i}, \mathbf{y}_i)\}_{i=1}^n$ to project the input data point $\mathbf{x_i}$ into a discriminative subspace as $\mathbf{q_i} \in \mathbb{R}^r$, where r is the dimensionality of the subspace and empirically selected as $r = K - 1$. In order to make use of the network data accumulated in various high-throughput technologies, we formulate them as graphs and convert them into vector form using Laplacian embedding [28]. Consequently, each protein acquires a description, $\mathbf{p_i}$, from network data. By concatenating the two transformed feature vectors for a protein, $\mathbf{q_i}$ and $\mathbf{p_i}$, we obtain a hybrid description $\mathbf{z_i}$, by which we conduct classification to predict protein function. In this work, we use K-nearest neighbor (KNN) method [25] ($K = 1$ in our implementation, which is abbreviated as 1NN in our paper.) due its simplicity and clear intuition. The key insight here is that, the hybrid descriptor $\mathbf{z_i}$ has a relatively small number of dimensions so that the classification can be computed efficiently. Moreover, because the discriminability of the data points in the projected subspace is enhanced by MLDA, together with the reinforcement using network data, the classification can be carried out more effectively. As another important contribution of this work, motivated by the computation process of MLDA, we propose to use the distance from a data point to the centroid of its predicted class to assess the reliability rank for a putative protein function, which is of great value for post-proteomic processes in biological experiments. We outline the framework to predict protein function using MLDA as in Table 2. In this work, and we focus on MLDA for feature reduction as it is the most essential part for a quality classification.

3 Multi-Label Linear Discriminant Analysis for Multi-Label Classification

Because the original feature vectors, $\mathbf{x_i}$, computed from protein sequence as in Section 2.1 is of high dimensionality and often non-discriminable, feature reduction is necessary for efficient and effective classification to infer protein function. As an successful feature reduction method in many practical applications, classical LDA, however, is by nature devised for single-label classification. To address this, in this section we will first analyze the problem to directly use classical LDA in multi-label classification scenarios, followed by applying our Multi-label Linear Discriminant Analysis approach [26] in for protein function prediction.

TABLE 2
Outlines to Predict Protein Function Using MLDA Approach

Input:
(a) Compute feature vector $\mathbf{x_i}$ for each protein sequence (Section 2.1).
(c) Construct label indicator $\mathbf{y}_i$ using annotation data (Section 2.2).

Multi-label protein function prediction using MLDA:
(a) Compute projected feature vector $\mathbf{q_i}$ for each protein from $\mathbf{x_i}$ using MLDA or NMLDA algorithm (Section 3).
(b) Compute embedded feature vector $\mathbf{p_i}$ for each protein from network data using Laplacian embedding (Section 4).
(c) Construct the hybrid feature vectors $\mathbf{z_i}$ by Eq. (21).
(d) Classify unannotated proteins $(\mathbf{z_i})_{i=m+1}^{n+m}$ by annotated proteins $(\mathbf{z_i}, \mathbf{y}_i)_{i=1}^{n}$ using KNN method, one function at time.

Output:
(a) Predicted functions for unannotated proteins, $(\mathbf{y}_i)_{i=n+1}^{n+m}$. One protein could acquire more than one putative functions.
(b) The reliability rank for putative protein functions (Section 5).

3.1 Review of Classical LDA

Classical LDA projects the original data from a high p-dimensional space to a much lower r-dimensional subspace, in which the classification task is much easier to perform and the results are more robust. Let the projection of LDA be

$$\mathbf{q}_i = U^T \mathbf{x_i}, \tag{4}$$

where $U \in \mathbb{R}^{p\times r}$ is the transformation matrix and $\mathbf{q}_i \in \mathbb{R}^r (r \ll p)$ is the projection of a data point $\mathbf{x_i}$ in the low-dimensional space, the goal of LDA is to find U such that different classes are more separated in the projection space. Let $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_n] \in \mathbb{R}^{r\times n}$, we have $Q = U^T X$.

For a single-label classification task, the training data $\{\mathbf{x_i}\}_{i=1}^n$ are partitioned into K exclusive pattern classes $\Pi = \{\pi_1, \ldots, \pi_K\}$ by prior knowledge, where π_k corresponds to the partition for the kth class and contains n_k data points. Thus, the between-class scatter matrix S_b and within-class scatter matrix S_w are computed as follows:

$$\begin{aligned} S_b(\mathbf{x}) &= \sum_{k=1}^{K} n_k (\mathbf{m_k} - \mathbf{m})(\mathbf{m_k} - \mathbf{m})^T, \\ S_w(\mathbf{x}) &= \sum_{k=1}^{K} \sum_{\mathbf{x_i}\in\pi_k} (\mathbf{x_i} - \mathbf{m_k})(\mathbf{x_i} - \mathbf{m_k})^T, \end{aligned} \tag{5}$$

where

$$\mathbf{m_k} = \frac{1}{n_k} \sum_{\mathbf{x_i}\in\pi_k} \mathbf{x_i}, \tag{6}$$

is the class mean (class centroid) of π_k and

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i}, \tag{7}$$

is the global mean (global centroid). Therefore, the total scatter matrix S_t is computed as

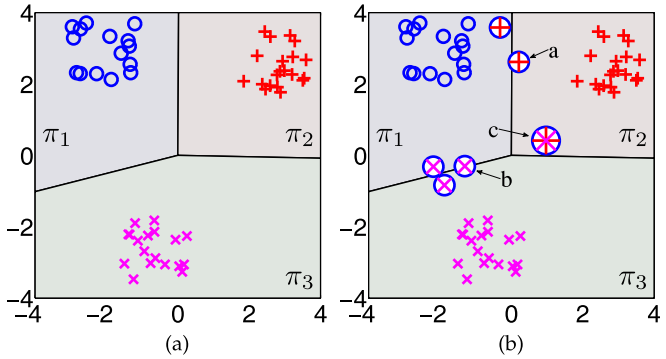


Fig. 1. Examples of classification problems. (a) A traditional single-label classification problem. Each data point clearly belongs to one cluster only. (b) A typical multi-label classification problem. Some data points belong to multiple classes. $\oplus$ denotes the data points belonging to both class π_1 and π_2, $\otimes$ denotes the data points belonging to both class π_1 and π_3, and the data points represented by $\circledast$ belong to all three classes. These data points with multiple labels cause the ambiguity in scatter matrices calculations.

$$S_t = \sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{m})(\mathbf{x_i} - \mathbf{m})^T = S_b + S_w. \tag{8}$$

The optimization criteria of LDA is that U is chosen such that data points from different classes are far away from one another ($\max S_b$) and data points from the same class are close to each other ($\min S_w$). This leads to the standard LDA optimization objective function as follows [25]:

$$\max_U J = \mathbf{tr}\left(\frac{U^T S_b U}{U^T S_w U}\right). \tag{9}$$

Since $\mathbf{tr}(A/B) = \mathbf{tr}(B^{-1}A) = \mathbf{tr}(AB^{-1})$, the solution can be obtained by applying the eigen-decomposition to matrix $S_w^{-1} S_b$ where S_w is assumed to be nonsingular.

Ambiguity Caused by Data Points with Multiple Labels in Classical LDA. Eqs. (5), (6), (7), (8), and (9) summarize the classical LDA algorithm, where the scatter matrices S_b, S_w, and S_t are well-defined in single-label multi-class classification. However, in multi-label classifications, the definitions are obscure. Fig. 1a illustrates an example of traditional single-label classification problems. The dataset has three different classes, with training data points denoted by blue $\circ$, red $+$, and magenta $\times$. Black thick lines denote the decision boundaries, and the background colors denote the decision regions of the respective classes. In single-label classification, each data point is uniquely assigned to one single class, thereby the data scatters are clear. However, in multi-label scenario, the decision regions overlap among one another and the decision boundaries are ambiguous. The inter and intra class scatters remain indistinct, because each data point could belong to multiple classes at the same time. As shown in Fig. 1b, for a typical multi-label classification problem, besides the training points only associated to one class, $\oplus$ denotes the training points associated to both class π_1 and class π_2, $\otimes$ denotes the data points associated to both class π_1 and class π_3, and data points represented by $\circledast$ have all three class labels. In this case, how much a data point with multiple labels should contribute to the data scatters is not defined. Thus the scatter matrices defined in Eqs. (5), (6), (7), and (8) can not be computed.

3.2 Multi-Label Linear Discriminant Analysis

Classical LDA deals with single-label multi-class classifications, where the partitions of data points are mutually exclusive. That is, $\pi_i \cap \pi_j = \varnothing$ if $i \neq j$. However this is no longer held in multi-label classifications. In this section, we present a multi-label LDA [26] for multi-label classification, which is a natural generalization of classical LDA.

Instead of defining the scatter matrices from data points perspective as in Eqs. (5), (6), (7), and (8) in classical LDA, we consider to formulate them by class-wise

$$\begin{cases} S_b = \sum_{k=1}^{K} S_b(k) \\ S_w = \sum_{k=1}^{K} S_w(k) \\ S_t = \sum_{k=1}^{K} S_t(k). \end{cases} \tag{10}$$

Through Eq. (10) the structural variances of the training data are represented more lucid and the construction of the scatter matrices is easier. Particularly, the ambiguity—how much a data point with multiple labels should contribute to the scatter matrices—p is avoided. Therefore, the *multi-label between-class scatter matrix* is defined as

$$S_b = \sum_{k=1}^{K} S_b^{(k)},\ S_b^{(k)} = \left(\sum_{i=1}^{n} Y_{ik}\right)(\mathbf{m_k} - \mathbf{m})(\mathbf{m_k} - \mathbf{m})^T, \tag{11}$$

the *multi-label within-class scatter matrix* S_w is defined as

$$S_w = \sum_{k=1}^{K} S_w^{(k)},\ S_w^{(k)} = \sum_{i=1}^{n} Y_{ik}(\mathbf{x_i} - \mathbf{m_k})(\mathbf{x_i} - \mathbf{m_k})^T, \tag{12}$$

and the *multi-label total scatter matrix* is defined as

$$S_t = \sum_{k=1}^{K} S_t^{(k)},\ S_t^{(k)} = \sum_{i=1}^{n} Y_{ik}(\mathbf{x_i} - \mathbf{m})(\mathbf{x_i} - \mathbf{m})^T, \tag{13}$$

where $\mathbf{m_k}$ is the mean of class k and $\mathbf{m}$ is the *multi-label global mean*, which are defined as follows:

$$\mathbf{m_k} = \frac{\sum_{i=1}^{n} Y_{ik}\mathbf{x_i}}{\sum_{i=1}^{n} Y_{ik}}, \quad \mathbf{m} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}\mathbf{x_i}}{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}}. \tag{14}$$

Now we write the multi-label scatter matrices in a more compact way in matrix form. First, let

$$\tilde{X} = X - \mathbf{m}\mathbf{e}^T, \tag{15}$$

where $\mathbf{e} = [1, \ldots, 1]^T$. Eq. (15) centers the input data for multi-label classification, which is different from data centering in single-label classification by classical LDA where $\tilde{X} = X(I - \mathbf{e}\mathbf{e}^T/n)$.

Let $W = \mathrm{diag}(w_1, \ldots, w_K)$, where $w_k = \sum_{i=1}^{n} Y_{ik}$ is the weight of class k in scatter matrices calculation, we have

$$S_b = \tilde{X}YW^{-1}Y^T\tilde{X}^T. \tag{16}$$

In single-label classification, $w_k = n_k$ is the number of data points in class k.

Let $L = \mathrm{diag}(l_1, \ldots, l_n)$, where $l_i = \sum_{k=1}^{K} Y_{ik}$ is the number of the labels attached to data point $\mathbf{x_i}$, we have

$$S_t = \tilde{X}L\tilde{X}^T. \tag{17}$$

In single-label classification, $L = I$, because each data point only belongs to one class.

Lemma 1. *When applied into single-label classification, the multi-label scatter matrices, S_b, S_w, and S_t, defined in Eqs. (11), (12), and (13), are reduced to their corresponding counterparts in classical LDA as defined in Eqs. (5), (6), (7), and (8).*

From the above definitions, Lemma 1 can be easily obtained. Most importantly, in classical LDA, $S_t = S_b + S_w$, which is still held in multi-label classifications.

Theorem 1. *For multi-label class-wise scatter matrices, $S_b^{(k)}$, $S_w^{(k)}$, and $S_t^{(k)}$ as defined in Eqs. (11), (12), and (13), the following relationship is held:*

$$S_t^{(k)} = S_b^{(k)} + S_w^{(k)}. \tag{18}$$

Therefore, $S_t = S_b + S_w$.

Proof. According to Eq. (14), we have $\sum_{i=1}^{n} Y_{ik}\mathbf{m_k} = \sum_i 1_{ik}^{nY}\mathbf{x}_i$. So, $\sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m_k}^T = \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{x}_i^T$ and $\sum_i = 1^n Y_{ik}\mathbf{m_k}\mathbf{m_k}^T = \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m_k}^T$. From Eqs. (11), (12), and (13), we have

$$\begin{aligned} S_t^{(k)} &= \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{x}_i^T \\ &= \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m_k}^T \end{aligned}$$

$$\begin{aligned} &S_b^{(k)} + S_w^{(k)} \\ &= \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m_k}^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m_k}^T \\ &\quad + \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m_k}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m_k}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{x}_i^T \\ &= \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m_k}^T \\ &\quad + \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m_k}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m_k}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{x}_i^T \\ &= \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m_k}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m_k}^T + \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T. \end{aligned}$$

Thus, $S_t^{(k)} = S_b^{(k)} + S_w^{(k)}$. Theorem 1 is proved. □

The optimization objective of multi-label LDA is hence defined in a similar way to Eq. (9) using the trace of ratio as following:

$$\max_U J_{\mathrm{MLDA}} = \mathbf{tr}\left(\frac{U^T S_b U}{U^T S_w U}\right). \tag{19}$$

Eq. (19) defines the proposed Multi-label Linear Discriminant Analysis algorithm when scatter matrices, S_b, S_w and S_t, are computed as in Eqs. (11), (12), and (13). In real applications, because the number of features of a dataset is often greater than the number of data points, S_w could be singular. Therefore in our implementation, we solve the eigenvalue problem $S_w^+ S_b\mathbf{u}_k = \lambda_k\mathbf{u}_k$, where S_w^+ is the pseudo-inverse of S_w. By taking the eigenvectors corresponding to the r largest eigenvalues, the transformation matrix U is obtained and the classification tasks can be performed on the projected data.

3.3 ℓ_1-Normalized MLDA

Our further analysis on MLDA in Section 3.2 shows that the data points with multiple labels are not only over-counted but also over-emphasized in scatter matrices calculations.

Over-Counting. For example, because data point **a** in Fig. 1b has two labels, π_1 and π_2, it is used in both $S_b^{(1)}$ and $S_b^{(2)}$ calculations. Because $S_b = S_b^{(1)} + S_b^{(2)} + S_b^{(3)}$, data point **a** is used twice in the between-class scatter matrix S_b. Similarly, data point **c** is used three times in both S_b and S_w. In general, in MLDA data points $\mathbf{x_i}$ with l_i labels is used l_i times in the scatter matrices, which are over-counted compared to the data points with single-label.

Over-Emphasizing. As shown in Fig. 1b, data points with multiple labels are usually far away from the class centroids. Because the contribution of a data point to the scatter matrix is proportional to its squared distance from the corresponding centroid, e.g., the contribution of data point $\mathbf{x_i}$ to $S_w^{(k)}$ is measured by $(\mathbf{x_i} - \mathbf{m_k})(\mathbf{x_i} - \mathbf{m_k})^T$. Compared to data points with single label, the influence of data points with multiple labels to the data scatters are over-emphasized.

While over-emphasizing is intrinsic in all LDA techniques and not easy to deal with, we correct the over-counting problem by ℓ_1-normalization in the following way. We define a new label matrix $\tilde{Y} \in \mathbb{R}^{n \times K}$ as

$$\tilde{Y}_{ik} = \begin{cases} 1/l_i & \text{if } \mathbf{x_i} \text{ belongs to the } k\text{th class,} \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

such that $\sum_{k=1}^{K} \tilde{Y}_{ik} = 1$. Therefore, the contribution of each data point to the scatter matrices is always weighted as 1, regardless it is associated with multiple labels or only single label.

With the definition of Eq. (20), $\tilde{Y}$ is the ℓ_1-normalized label matrix. By replacing Y by $\tilde{Y}$, the scatters matrices can be computed in the same forms as in Eqs. (11), (12), and (13). We call them as ℓ_1-normalized multi-label scatter matrices, hence ℓ_1-normalized MLDA (NMLDA) is defined in the same way as in Eq. (19) using the ℓ_1-normalized multi-label scatters matrices. In single-label classifications, l_i is always 1 and NMLDA is also reduced to classical LDA with the following Lemma:

Lemma 2. *When NMLDA is applied into single-label classification, the multi-label scatter matrices, $\tilde{S}_b$, $\tilde{S}_w$, and $\tilde{S}_t$, are reduced to their corresponding counterparts in classical LDA as defined in Eqs. (5), (6), (7), and (8).*

Similar to Theorem 1, we also have:

Theorem 2. *For multi-label class-wise scatter matrices in NMLDA, $\tilde{S}_t = \tilde{S}_b + \tilde{S}_w$ is still held.*

4 Incorporating Network Data

The recent availability of protein interaction networks for many model species opens another area to predict protein function using graph algorithms, and many computational approaches have been developed [29]. In order to achieve more accurate function predictions, we incorporate network data and use Laplacian embedding [28] to convert the graph data obtained from biological networks into vectors, $\mathbf{p_i} \in \mathbb{R}^r$, one for each protein. By concatenating them with projected feature vector $\mathbf{q_i}$ computed from MLDA or NMLDA algorithm, we obtain a hybrid feature vector $\mathbf{z_i}$ for each protein as following:

$$\mathbf{z_i} = \begin{bmatrix} \mathbf{q_i} \\ \alpha \mathbf{p_i} \end{bmatrix}, \tag{21}$$

where α is a tradeoff parameter and empirically selected as $\alpha = \sqrt{\frac{\sum_{i,j,i \neq j} \|\mathbf{q_i} - \mathbf{q_j}\|^2}{\sum_{i,j,i \neq j} \|\mathbf{p_i} - \mathbf{p_j}\|^2}}$, because KNN uses the euclidean distance in the hybrid feature space to make classification. In the rest of this section, we derive $\mathbf{p_i}$ and reveal its enrichment from graph-cut perspective of view.

In this work, we download protein-protein interaction (PPI) data from BioGRID (version 2.0.56) database [30] and focus on the *Saccharomyces cerevisiae* species. By removing the proteins connected by only one PPI, we have the same number of protein as obtained from sequence database, i.e., we totally have $\tilde{n} = n + m$ proteins with $n = 4,403$ annotated proteins and $m = 1,988$ unannotated proteins. There are 89,452 PPIs among these proteins.

Protein interaction network is routinely modeled as a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertices $\mathcal{V}$ represent proteins $\{\mathbf{x}_1, \ldots, \mathbf{x}_{\tilde{n}}\}$, and the edges $\mathcal{E}$ are weighted by an $\tilde{n} \times \tilde{n}$ similarity matrix W with W_{ij} indicating the similarity between $\mathbf{x_i}$ and $\mathbf{x_j}$. In the simplest case, W is the adjacency matrix of the protein-protein interaction graph where $W_{ij} = 1$ if protein $\mathbf{x_i}$ and $\mathbf{x_j}$ interact, and 0 otherwise. For the graph data with pairwise relationship W, Laplacian embedding preserves the same relationships and maximize the smoothness with respect to the intrinsic manifold of the dataset in the embedding space by minimizing the following objective [28]:

$$\min J_{\text{Lap}} = \min_{P} \mathbf{tr}\big(P^T (D - W) P\big), \tag{22}$$

where $P^T = [\mathbf{p}_1, \ldots, \mathbf{p}_n] \in \mathbb{R}^{r \times \tilde{n}}$ are the embeddings of the data points, and $D = \text{diag}(d_1, \ldots, d_{\tilde{n}})$, $d_i = \sum_j W_{ij}$. Thus, $L = D - W$ is the graph Laplacian [31]. The solution to this problem is well established in mathematics by solving the eigenvalue problem, $(D - W)\mathbf{v_k} = \lambda_k \mathbf{v_k}$, where $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues and $\mathbf{v_k}$ are the corresponding eigenvectors. Because $\mathbf{u}_1$ is a constant vector [31], we use the first r non-trivial eigenvectors to construct P, i.e., $P = [\mathbf{v}_2, \ldots, \mathbf{v}_{r+1}]$. Here, we select $r = K - 1$, same as that in MLDA, such that the construction of $\mathbf{z_i}$ in Eq. (21) is balanced.

The true power of Laplacian embedding lies in that it amounts to K-ways ratio-cut graph partition when K-means clustering is performed on P [32]. Namely, the PPI graph are partitioned into K exclusive parts according to its topology. With this enhancement, the performance of the classification conducted on the hybrid feature vectors $\mathbf{z_i}$ is further improved.

5 Reliability Rank of Putative Functions

One of the primary goal of computational approaches to predict protein function is to assist biologist to discover new functional roles of proteins for experimental verification. Therefore, instead of simply assigning a "yes" or "no" to a prediction as in many existing approaches, a

real-valued reliability rank is often of great use in post-processing of proteomic analysis. For example, biologist may use the reliability rank as testable hypothesis to conduct biological experiments on highly-reliable putative functions, so that the cost on utility of the expensive experimental equipments can be minimized.

Quantitatively evaluating the reliability of a prediction is usually not easy, because the underlying probability model and the actual training and testing data distributions are constantly changing for different biological functions. However, this problem can be intuitively resolved by borrowing the idea in the computation process of LDA, because the compactness a within-class scatter is a good measure of the optimization by LDA. The closer a data point is from the centroid of its predicted class, the more representative it is for this class. Specifically, given the centroid for the kth class as

$$\mathbf{m_k} = \frac{\sum_{i=1}^{n} Y_{ik}\mathbf{z_i}}{\sum_{i=1}^{n} Y_{ik}}, \tag{23}$$

the reliability rank for a prediction $\{Y_{ik}\}_{i=n+1}^{n+m}$ is computed as

$$r(Y_{ik}) = e^{\left(-\|\mathbf{z_i}-\mathbf{m_k}\|^2\right)}. \tag{24}$$

Apparently, the data points close to the centroid of its predicted class has high reliability rank, and the data points far away from the centroid has low reliability rank. In the perfect case when an unannotated protein happens to coincide with the centroid of a functional class, we have 100 percent confidence to assign the function to this protein.

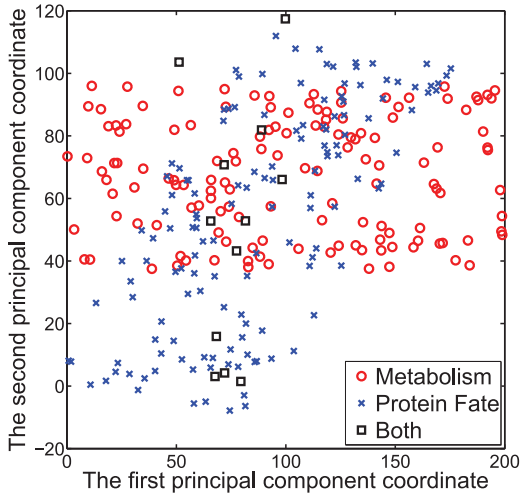
6 Results and Discussion

We evaluate the Multi-label Linear Discriminant Analysis in function prediction for yeast proteins. The sequence data for the proteins are obtained from GenBank [21] (Section 2.1), and the network data are downloaded from BioGRID [30] (Section 4). The dataset used in our evaluations contains 6,392 proteins, among which 4,403 proteins are annotated according to the FunCat annotation database [22] and 1,988 proteins remain unannotated (Section 2.2). We focus on the main functional categories defined in FunCat annotation scheme, and 17 functional classes are involved in functional annotation for yeast species.

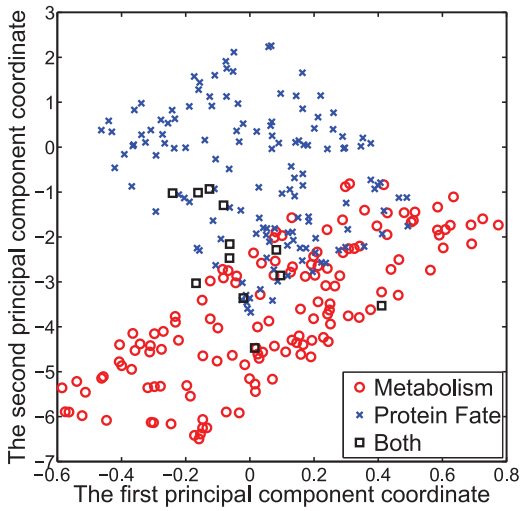
6.1 Improved Discriminability by MLDA

The main purpose of MLDA is for feature reduction, through which the data points in the projected space should be more separable. Therefore we first evaluate its projection effectiveness.

We use functional classes, "Metabolism" and "Protein Fate", for illustration, because these two functions have the biggest number of annotated proteins as can be seen in Table 1. We randomly pick up 150 proteins from each functional class, and among the selected proteins there are 12 proteins are annotated with the both functions. Therefore, we end up with 288 data points. We do not use all the 4,403 annotated proteins, because too many data points will fill up the visualization panel and mess up the illustration, though the same conclusion can be drawn. We first project these data points from their original data space ($p = 20^3$) onto the 2D plane by principal component analysis (PCA) (using the first two principal component coordinates and PCA is used only for visualization purpose) as shown in Fig. 2a. Clearly, the data points from the two classes are mingled together and it is difficult to find a decision boundary with high classification accuracy. We then run MLDA on the whole dataset with all the 17 functions, and project the same data points from the reduced data space ($r = 16$) onto the 2D plane as shown in Fig. 2b. Here we only use two classes in demonstration, because there are too many proteins are annotated with more than one functions and clear depiction for all the functional classes can only be illustrated in a higher-dimensional space but not on the 2D visualization plane. Obviously, data points in the two different classes are distinctly separated from each other as in Fig. 2b. All these observations demonstrate that MLDA is indeed an effective feature reduction algorithm, which not only significantly reduces the computational complexity (from 20^3 dimensions to 16 dimensions) of the classification task but also improves the discriminability



(a) Projection from the original space ($p = 20^3$) onto the 2D plane.



(b) Projection from the reduced subspace ($r = 17$) by MLDA onto the 2D plane.

Fig. 2. Projection of randomly selected data points from two functional classes on the 2D plane. The red circles denotes the proteins only annotated with function "Metabolism", the blue crosses denote the proteins only annotated with function "Cell Fate", and the black squares denote the proteins annotated with the both functions.

of the input data. Therefore, through MLDA the subsequent classification can be carried out on the projected data points more efficiently and effectively.

6.2 Comparison with Existing Computational Approaches for Protein Function Prediction

We evaluate the function prediction capability of the proposed MLDA approach by comparing it against two well known existing approaches: (1) functional similarity weight (FS) approach [33] and (2) fusion kernel (FK) approach [34], one baseline approach from biological perspective: (3) majority voting (MV) approach [35], and two baseline approaches from machine learning perspectives: (4) linear discriminant analysis [25] and (5) multi-label support vector machine (ML-SVM) [36]. Besides, we also compare our approach to three most recent approaches: (6) Laplacian Network Partitioning incorporating function category Correlations (LNPC) approach [37], (7) function-function correlated multi-label protein function prediction over interaction networks (FCML) approach [38], and (8) Maximization of Data-Knowledge Consistency (MDKC) approach [16].

6.2.1 Evaluation Methods and Metrics

We use standard five-fold cross validation (CV) method in our performance evaluations. The proteins are divided into 5 equal-size groups randomly. One group is assumed to be unannotated and the rest four groups are annotated. We run a prediction methods to predict the functions for the kept-out group of proteins. The predicted results are compared to the true functions of these proteins. This is repeated 5 times to keep each group as unannotated in turn, and final results are averaged.

As in many previous studies, we choose *precision* and *F1 score* as the prediction performance metrics. Let true positive (TP) be the number of proteins which we correctly predict to have a given function, false positive (FP) be the number of proteins which we incorrectly predict to have the function, and false negative (FN) be the number of proteins which we incorrectly predict to not have the function. The "precision" is defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, and the "recall" (also known as "sensitivity") is defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$. In addition, we also use the "F1 score" to evaluate precision and recall together, which is the harmonic mean of precision and recall: defined as $(2\times\mathrm{Precision}\times\mathrm{Recall})/(\mathrm{Precision}+\mathrm{Recall})$. F1 score is extensively used in the related works and other domains such as information retrieval. Typically, improving the precision of an algorithm decreases its recall and vice versa, therefore F1 score is a balanced performance metric. To measure the overall prediction performance, we use average precision and average F1 score over all 17 main functional categories to evaluate our algorithm.

6.2.2 Implementation Details

FS-weight approach predicts protein function using the protein interaction network and transfer functions to a protein from its directly and indirectly connected neighbors. We download the implementation codes from the authors' web site and conduct the experiments on the PPI graph as described earlier in Section 4. Because this approach does not explicitly supply a threshold for prediction, we use the one that gives the highest F1 score. FK approach use Semi-Definite Programming (SDP) to combine heterogeneous data sources for function prediction using Support Vector Machines (SVM). A separate kernel is generated from each data source using customized techniques. SDP is then used to obtain an optimal combination of the kernels for SVM learning. We build two kernels, one from protein sequences and the other from the PPI graph. Instead of computing the weights by ourselves, we use those computed by the original work for a fair comparison. Again, we use the threshold giving best F1 score to make function prediction.

MV approach assigns functions to a protein via its connecting neighbors, which, though simple, proves to be useful in the early ages because of the clear intuitions. In our implementation, we make prediction using the top 3 frequent functions appearing one protein's interacting partners.

LDA approach [25] is one of the most broadly used statistical learning method for dimension reduction, which also motivates our work. We learn one projection for each function and predict the functions for unannotated proteins using 1NN in the projected space. ML-SVM [36] extends the classical support vector machine to deal with multi-label data.

LNPC approach [37] considers protein function prediction as a binary voting and FCML approach [38] simulates protein function prediction as a electric flow, both of which utilize the topology of a protein-protein interaction network. MDKC approach [16] assign protein functions by minimizing the differences between data networks and knowledge networks, where the former refers to original experimental measurements or results while the latter refers to human-curated research findings recorded in well structured databases or documented in biomedical literatures.

We evaluate our proposed MLDA and NMLDA approaches in the following three ways. We first conduct classification using the projected sequence features by MLDA ($\mathbf{q_i}$), and then make prediction on the hybrid features ($\mathbf{z_i}$) constructed by projected sequence vectors ($\mathbf{q_i}$) by MLDA and embedded graph vectors ($\mathbf{p_i}$). Finally we verify the performance of NMLDA using the hybrid features constructed by its projected sequence features and the embedded graph features. The classification is carried out by 1NN, one function at a time. For each function, the classification is conducted as a binary classification task.

6.2.3 Results and Discussion

Because we formulate protein function prediction as a multi-label classification problem, we are more concerned with the overall performance over all the functional classes. Table 3 presents the overall function prediction performance comparison of the compared approaches measured by average precision and average F1 score. The results demonstrate that the proposed MLDA and NMLDA approaches clearly outperform the other approaches, especially when they work together with the embedded features obtained from network data. Moreover, the advantage of NMLDA approach compared to MLDA approach justifies the ℓ_1-normalization is necessary to alleviate the over-counting problem.

Besides evaluate our approach for protein functions defined by Funcat annotation scheme, we also compare our method against the counterparts when Gene Ontology is used for annotation, which is more comprehensive than the Funcat annotation scheme. We randomly pick up 30 terms from

TABLE 3
Average Precision and Average F1 Score by the Compared Approaches on the Main Functional Categories by FunCat Annotation Scheme

Approaches	Average Precision	Average F1 score
FS	32.84%	22.68%
FK	54.64%	39.01%
MV	30.12%	28.56%
LDA	50.11%	40.12%
ML-SVM	51.19%	41.22%
LNPC	49.20%	43.70%
FCML	54.83%	43.74%
MDKC	61.38%	42.17%
MLDA	60.30%	41.24%
MLDA + graph	63.68%	43.74%
NMLDA + graph	64.09%	43.97%

We evaluate our propopsed MLDA and NMLDA approaches in three ways: on projected sequence vectors by MLDA ($\mathbf{q_i}$) denoted as "MLDA", on hybrid features vectors constructed by projected sequence vectors ($\mathbf{q_i}$) by MLDA and embedded graph features ($\mathbf{p_i}$) denoted as "MLDA + graph", and on the hybrid features constructed by the projected sequence features by NMLDA and the embedded graph features denoted as "NMLDA + graph".

molecular function and biological process of GO and use the selected 60 terms to annotate the same set of proteins as above. The performance metrics are used and results are reported in Table 4, which once again demonstrate the effectiveness of our approach in protein function prediction.

In addition, Fig. 3 shows the class-wise prediction performance. The results show that, besides the overall performance, the proposed MLDA and NMLDA approach consistently outperform the other approaches in most of the individual functional classes, which again confirms the effectiveness of the proposed algorithms.

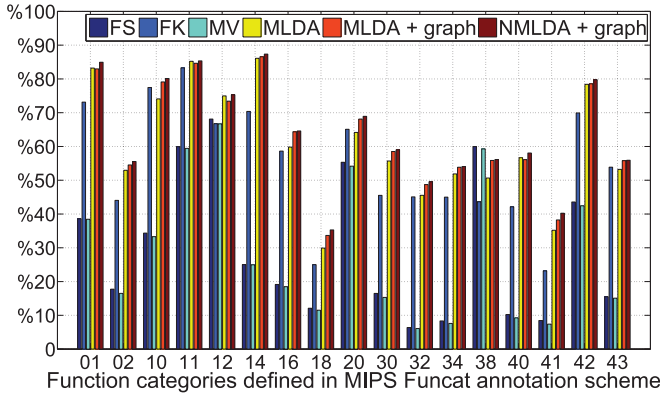
6.3 Putative Functions of Unannotated Proteins

Because one of the most important contribution of computational approaches to biological research is to discover and suggest putative protein functions for experimental verification, we apply NMLDA (plus embedded graph data) approach to infer functions for unannotated proteins. A list of putative

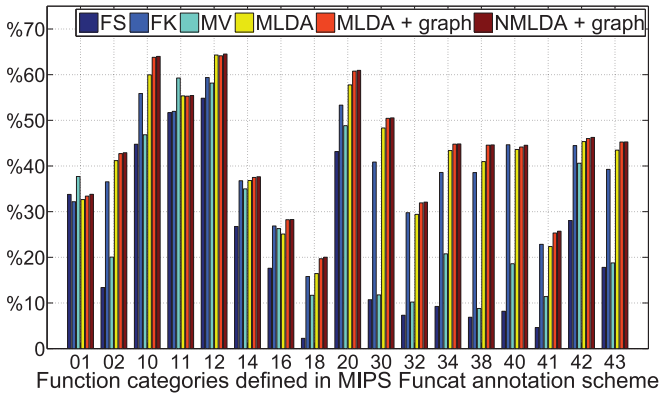
TABLE 4
Average Precision and Average F1 Score by the Compared Approaches on 60 Randomly Selected GO Terms

Approaches	Average Precision	Average F1 score
FS	35.11%	26.78%
FK	41.36%	332.3%
MV	34.66%	24.15%
LDA	43.33%	32.43%
ML-SVM	44.12%	33.15%
LNPC	40.15%	30.63%
FCML	46.12%	34.87%
MDKC	50.14%	38.42%
MLDA	51.54%	40.11%
MLDA + graph	55.61%	42.62%
NMLDA + graph	55.47%	43.07%

We evaluate our proposed MLDA and NMLDA approaches in three ways: on projected sequence vectors by MLDA ($\mathbf{q_i}$) denoted as "MLDA", on hybrid features vectors constructed by projected sequence vectors ($\mathbf{q_i}$) by MLDA and embedded graph features ($\mathbf{p_i}$) denoted as "MLDA + graph", and on the hybrid features constructed by the projected sequence features by NMLDA and the embedded graph features denoted as "NMLDA + graph".



(a) Precision.



(b) F1 score

Fig. 3. Performance of five-fold cross validation for the main functional categories in FunCat scheme by FS, FK, MV, and proposed MLDA approaches evaluated in three ways (same as in Table 3) on yeast species.

functions predicted by our algorithm with reliability rank value greater than 0.7 are provided in Table S1 (supplied as supplementary information due to space), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2016.2591529. The corresponding reliability rank values are also reported. For example, we predict protein "YHL023C" to have function "11" (Transcription) with reliability rank of 0.82 and function "32" (Cell Rescue, Defense and Virulence) with reliability rank of 0.41, which means our our algorithm suggests that protein "YHL023C" is more likely to be annotated with function "11" rather than function "32".

7 Conclusion and Future Works

In this paper, we addressed the issue of using classical LDA, a famous feature reduction method in statistical learning, in protein function prediction using protein sequences. The former is by nature designed for *single-label classification*, while the latter is an ideal incarnation of *multi-label classification*. Feature reduction is necessary in predicting protein function from sequence, because the original features extracted from sequence data is of high dimensionality and often contains irrelevant patterns, which makes the classification inefficient and ineffective. Thus we applied the Multi-label Linear Discriminant Analysis approach [26] to deal with multi-label classification problems and meanwhile preserve the powerful classification capability of classical LDA. We presented a class-wise scatter matrices computation scheme to avoid the

the ambiguities in scatter construction caused by the the data points with multiple labels. We further extended MLDA to NMLDA by employing ℓ_1-normalization to overcome the problem of over-counting data points with multiple labels in scatter matrices calculations, such that the contribution of a single data point to the scatter matrices is always weighted as 1. Using Laplacian embedding, we successfully incorporated the biological network data into our method in a natural and integral way, such that the prediction performance is improved by taking advantage of the information from multiple different data sources. Motivated by the computation process of LDA, we devised the reliability rank to assess the confidence of putative functions, which may greatly facilitate the the post-proteomic processes. Through extensive evaluations from different aspects, our proposed MLDA and NMLDA approaches have demonstrated promising results, which empirically confirms their usefulness.

We notice that the function prediction by integrating network data with sequence data in our evaluation exhibits enhanced performance, which hint us the integration of heterogenous biological experimental data may help to improve the predictive accuracy. Indeed, many computational approaches have already been proposed to make use of the information from multiple data sources through various biological or statistical mechanisms. The fusion kernel approach [34] used in our empirical evaluation is an good example of such methods. Therefore, in our future work, we will further develop our methods to leverage multiple types of biological experimental data to achieve more meaningful protein function predictions.

Acknowledgments

Heng Huang is the corresponding author. This work was partially supported by US NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, and NIH R01 AG049371.

References

[1] G. Pandey, V. Kumar, and M. Steinbach, "Computational approaches for protein function prediction: A survey," Dept. Comput. Eng., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. 06-028, 2006.

[2] W. Pearson and D. Lipman, "Improved tools for biological sequence comparison," *Proc. Nat. Academy Sci. United States America*, vol. 85, no. 8, 1988, Art. no. 2444.

[3] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990.

[4] S. Altschul, et al., "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, 1997, Art. no. 3389.

[5] J. Gerlt and P. Babbitt, "Can sequence determine function?" *Genome Biol.*, vol. 1, no. 5, 2000.

[6] J. Whisstock and A. Lesk, "Prediction of protein function from protein sequence and structure," *Quart. Rev. Biophysics*, vol. 36, no. 3, pp. 307–340, 2004.

[7] P. Bork and E. Koonin, "Protein sequence motifs," *Current Opinion Structural Biol.*, vol. 6, no. 3, pp. 366–376, 1996.

[8] F. Servant, et al., "ProDom: Automated clustering of homologous domains," *Briefings Bioinf.*, vol. 3, no. 3, 2002, Art. no. 246.

[9] X. Wang, D. Schroeder, D. Dobbs, and V. Honavar, "Automated data-driven discovery of motif-based protein function classifiers," *Inf. Sci.*, vol. 155, no. 1/2, pp. 1–18, 2003.

[10] K. Blekas, D. Fotiadis, and A. Likas, "Motif-based protein sequence classification using neural networks," *J. Comput. Biol.*, vol. 12, no. 1, pp. 64–82, 2005.

[11] A. Ben-Hur and D. Brutlag, "Sequence motifs: Highly predictive features of protein function," in *Proc. Found. Appl.*, 2006, pp. 625–645.

[12] L. Jensen, et al., "Prediction of human protein function from post-translational modifications and localization features," *J. Mol. Biol.*, vol. 319, no. 5, pp. 1257–1265, 2002.

[13] R. King, A. Karwath, A. Clare, and L. Dehaspe, "Accurate prediction of protein functional class from sequence in the Mycobacterium tuberculosis and Escherichia coli genomes using data mining," *Yeast*, vol. 17, no. 4, pp. 283–293, 2000.

[14] C. Cai, L. Han, Z. Ji, X. Chen, and Y. Chen, "SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Res.*, vol. 31, no. 13, 2003, Art. no. 3692.

[15] R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner, "Improving protein function prediction using the hierarchical structure of the Gene Ontology," in *Proc. IEEE Symp. Comput. Intell. Bioinf. Comput. Biol.*, 2005, pp. 1–10.

[16] H. Wang, H. Huang, and C. Ding, "Correlated protein function prediction via maximization of data-knowledge consistency," in *Research in Computational Molecular Biology*. Berlin, Germany: Springer, 2014, pp. 311–325.

[17] W. Haque, A. Aravind, and B. Reddy, "Pairwise sequence alignment algorithms: A survey," in *Proc. Conf. Inf. Sci. Tech. Appl.*, 2009, pp. 96–103.

[18] C. Notredame, "Recent evolutions of multiple sequence alignment algorithms," *PLoS Comput. Biol*, vol. 3, no. 8, 2007, Art. no. e123.

[19] D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," *Proc. Eur. Conf. Mach. Learn.*, 1998, pp. 4–18.

[20] C. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[21] D. Benson, I. Karsch-Mizrachi, and D. Lipman, "GenBank," *Nucleic Acids Res.*, vol. 34, pp. D16–D20, 2006.

[22] H. Mewes, et al., "MIPS: A database for genomes and protein sequences," *Nucleic Acids Res.*, vol. 27, no. 1, 1999, Art. no. 44.

[23] G. O. Consortium, "The Gene Ontology project in 2008, " *Nucleic Acids Res.*, vol. 36, no. suppl 1, pp. D440–D444, 2008.

[24] P. Radivojac, et al., "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, no. 3, pp. 221–227, 2013.

[25] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York, NY, USA: Academic, 1990.

[26] H. Wang, C. Ding, and H. Huang, "Multi-label linear discriminant analysis," in *Computer Vision*. Berlin, Germany: Springer, 2010, pp. 126–139.

[27] H. Wang, C. H. Ding, and H. Huang, "Multi-label classification: Inconsistency and class balanced k-nearest neighbor," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1264–1266.

[28] K. Hall, "An r-dimensional quadratic placement algorithm," *Manage. Sci.*, vol. 17, pp. 219–229, 1970.

[29] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Mol. Syst. Biol.*, vol. 3, no. 1, 2007, Art. no. 88.

[30] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: A general repository for interaction datasets," *Nucleic Acids Res.*, vol. 34, 2006, Art. no. D535.

[31] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.

[32] L. Hagen and A. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074–1085, Sep. 1992.

[33] H. Chua, W. Sung, and L. Wong, "Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions," *Bioinformatics*, vol. 22, no. 13, pp. 1623–1630, 2006.

[34] G. Lanckriet, M. Deng, N. Cristianini, M. Jordan, and W. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," in *Proc. Pacific Symp. Biocomputing*, 2004, vol. 9, pp. 300–311.

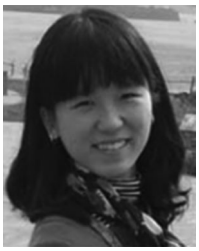
[35] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein-protein interactions in yeast," *Nat. Biotech.*, vol. 18, pp. 1257–1261, 2000.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011, Software Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[37] H. Wang, H. Huang, and C. Ding, "Protein function prediction via Laplacian network partitioning incorporating function category correlations," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2049–2055.

[38] H. Wang, H. Huang, and C. H. Ding, "Function-function correlated multi-label protein function prediction over interaction networks," *J. Comput. Biol.*, vol. 20, no. 4, pp. 322–343, 2013.

Hua Wang received the bachelor's degree from Tsinghua University, China, in 1999, the master's degree from Nanyang Technological University, Singapore, in 2003, and the PhD degree in computer science from the University of Texas at Arlington, in 2012. He is an assistant professor in the Department of Electrical Engineering and Computer Science, Colorado School of Mines. His research interests include machine learning and data mining, as well as their applications in bioinformatics, health informatics, medical image analysis, computer vision, and cheminformatics.



Lin Yan received the BSc degree in automation and the MSc degree in science in engineering from Shanghai Jiao Tong University, in 2010 and 2013, respectively. She is working toward the PhD degree in the Computer Science and Engineering Department, University of Texas at Arlington. Since July 2013, she has been working for the Department of Electronic Engineering, Shanghai Jiao Tong University, China.

Heng Huang received the BS and MS degrees from Shanghai Jiao Tong University, Shanghai, China, in 1997 and 2001, respectively, and the PhD degree in computer science from Dartmouth College, in 2006. He started working as an assistant professor in the Computer Science and Engineering Department, University of Texas at Arlington, in 2007, and became a tenured associate professor in the same department, in 2013. Since 2015, he has been a full professor in the same department. His research interests include machine learning, data mining, bioinformatics, neuroinformatics, and health informatics.

Chris Ding is a professor of computer science with the University of Texas at Arlington. His research include data mining, high performance computing, bioinformatics, etc. He studied theoretical physics with Columbia University, where the PhD work was mainly on building a parallel computer which appeared in front cover of science. He worked at Caltech, Jet Propulsion Lab, and Berkeley National Lab, before joining UTA. A main thread of his work is utilizing matrix models for solving machine learning problems, the equivalence of principal component analysis and K-means clustering, clustering properties of nonnegative matrix factorizations, and L21 matrix norm. He has given seminars at UC Berkeley, Stanford, CMU, Waterloo U, Alberta U, Google Research, IBM Research, Microsoft Research, etc. He has published about 200 papers that were cited 18,900 times.