Genome-Wide Protein Function Prediction through Multi-instance Multi-label Learning

Jian-Sheng Wu, Sheng-Jun Huang and Zhi-Hua Zhou, *IEEE Fellow*

Abstract—Automated annotation of protein function is challenging. As the number of sequenced genomes rapidly grows, the vast majority of proteins can only be annotated computationally. Nature often brings several domains together to form multi-domain and multi-functional proteins with a vast number of possibilities, and each domain may fulfill its own function independently or in a concerted manner with its neighbors. Thus, it is evident that the protein function prediction problem is naturally and inherently Multi-Instance Multi-Label (MIML) learning tasks. Based on the state-of-the-art MIML algorithm MIMLNN, we propose a novel ensemble MIML learning framework EnMIMLNN and design three algorithms for this task by combining the advantage of three kinds of Hausdorff distance metrics. Experiments on seven real-world organisms covering the biological three-domain system, i.e., archaea, bacteria, and eukaryote, show that the EnMIMLNN algorithms are superior to most state-of-the-art MIML and Multi-Label learning algorithms.

Index Terms—Protein Function Prediction; Genome Wide; Machine learning; Multi-instance Multi-label Learning; Hausdorff distances; Ensemble learning

1 Introduction

Understanding biological functions of proteins is a key challenge in the post-genomic era. Due to its inherent difficulty and expense, experimental annotation of protein functions cannot scale up to accommodate the vast amount of sequence data already available. As the number of sequenced genomes rapidly grows, the overwhelming majority of protein products can only be annotated computationally[1]. Therefore, computational methods that predict biological functions of a protein have emerged as an urgent problem at the forefront of computational and molecular biology. By constructing classifiers to predict the functions, machine learning techniques provide an effective approach for this task.

A protein domain among a given protein sequence or structure is a conserved part that can evolve, function, and exist independently to the rest of the protein chain. Molecular evolution uses domains as building blocks and these may be recombined in different arrangements to create proteins with different functions. Many proteins consist of several structural domains, and one domain may appear in a variety of different proteins. Multi-domain proteins are likely to have emerged from a selective pressure during evolution to create new functions. Various proteins have diverged from common ancestors by different combinations and associations of domains. Nature often brings several domains together to form multi-domain and multifunctional proteins with a vast number of possibilities[2]. The majority of genomic proteins, two-thirds in unicellular organisms and more than 80% in metazoa, are multi-domain proteins[3]. In a multi-domain protein, each domain may fulfill its own function independently, or in a concerted manner with its neighbors[4].

Recently, Zhou and Zhang (2007) [5] proposed the Multi-Instance Multi-Label learning (MIML) framework, where one object is represented by a bag of instances and the object is allowed to have multiple labels simultaneously. Labels of the training objects are known, however, labels of instances are unknown. We can represent each domain with an input instance and take each biological function as an output label. Thus, it is evident that the protein function prediction problem is naturally and inherently MIML learning tasks. During the past few years, a number of MIML algorithms have been developed. To name a few, the MIMLSVM and MIMLBoost approaches were proposed by degenerating MIML task to a simplified single-instance multi-label and multi-instance single-label learning, respectively[5, 6]; SISL-MIML formulates MIML as a combination of two optimizations of a quadratic programming and an integer programing[7]; the DBA approach formulates MIML as a probabilistic generative model [8]; the RankLossSIM approach optimizes the label ranking loss for bag and instance annotation[9]. The MIML learning framework have already been successfully applied in many real-world tasks. For example, Xu et al. proposed a ensemble Multi-Instance Multi-Label learning approach named En-MIMLSVM for Video Annotation Task[10]; In bioinformatics, Li et al. disclosed that the underlying nature of Drosophila gene expression pattern annotation problem matches well with the MIML learning framework and proposed a MIML algorithm named MIMLSVM+ for this task[11].

Although the underlying nature of protein function

- *J.-S. Wu is with the National Key Laboratory for Novel Software Technology, Nanjing University, 210023, and the School of Geography and Biological Information of Nanjing University of Posts and Telecommunications, 210046, Nanjing, China. E-mail: wujs@lamda.nju.edu.cn.*
- *S.-J. Huang is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: huangsj@lamda.nju.edu.cn.*
- *Z.-H. Zhou is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: zhouzh@lamda.nju.edu.cn.*

prediction problem matches well with the MIML learning framework, till now there is no attempt that has been made under this learning framework. During the last couple of years, many machine learning methods have been proposed for predicting protein function combing with multiple data sources including amino acid sequences, inferred evolutionary relationships and genomic context, protein-protein interaction networks, protein structure data, microarrays or a combination of multiple data types [1, 12]. These machine learning methods fall into two major groups. The first group is based on traditional supervised learning framework [13-15], while the second group relies on multi-label learning framework [16-21]. Traditional supervised learning framework is evidently a degenerated version of multi-instance learning framework as well as a degenerated version of multi-label learning framework, while traditional supervised learning framework, multi-instance learning framework and multi-label learning framework are all degenerated versions of MIML learning framework [5, 6].Such degenerated strategies may lose helpful information in the feature space, and further hurts the prediction performance[5, 6]. This is not strange because for a protein associated with multiple domains and multiple functions, if it is described by only a single instance, the information of describing relationships between domains and functions are mixed and thus difficult for learning; if we can transform the single-instance into a set of instances by representing each domain with an input instance, the mixed information might become explicit and clearer and thus less difficult for learning[5, 6].

Then,based on a state-of-the-art MIML algorithm MIMLNN[6], we propose an ensemble learning framework called EnMIMLNN to address the prediction task. In EnMIMLNN, we replace back propagation (BP) neural network with an innovative neural network style named RBF neural networks derived from the popular radial basis function (RBF) method[22] which are adapted to learn from MIML examples. Meanwhile, in EnMIMLNN we combines with three kinds of Hausdorff distance metrics[23], to measure the similarity of two proteins, resulting in a distance with following advantages: (1) it considers the contribution of all domains contained in the proteins; (2) it well refects the relationship between different domains in a specific protein; (3) it is robust and less sensitive to outliers[23]. Under the EnMIMLNN learning framework, three algorithms are proposed based on metric-based and classfier voting-based ensemble ways. Multiple experiments for genome-wide protein function prediction on seven real-world species covering the biological three-domain system[24], i.e., archaea, bacteria, and eukaryote, show that the EnMIMLNN algorithms are superior to the vast majority of state-of-the-art MIML and Multi-Label learning approaches in most cases.

The rest of the paper is organized as follows. Section 2 shows the relation between the prediction problem and the MIML learning framework, and presents the ensemble MIML learning framework. Section 3 reports on experimental configuration and results. Section 4 concludes the paper.

2 THE PROPOSED METHOD

2.1 The Formulation of the Protein Function Prediction Task

Domains can be thought as distinct functional and structural units of a protein, and nature often brings several domains together to form multi-domain and multi-functional proteins with a vast number of possibilities[2]. Figure 1 gives an illustration by the protein "type 1 Insulin-like growth factor receptor (PDB ID: 1IGR)" whose image sources from the Conserved Domain Database (http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml) of National Center for Biotechnology Information(NCBI). The image illustrates four conserved domains identified as structural units in the PDB-entry 1IGR, as segments highlighted with dashed oval. Protein function can be described in multiple ways, and we focus on the molecular function aspect provided by the Gene Ontology (GO) Consortium[25] in this paper. As shown in Figure 1, this protein is annotated with six GO molecular functions. During molecular evolution, domains may have been utilized as building blocks and recombined in different arrangements to modulate protein function. Each domain may fulfill its own function independently, or in a concerted manner with its neighbors[4]. Thus, automatic prediction of GO molecular function is challenging since it is not explicit which GO term is assigned to which domains in proteins.

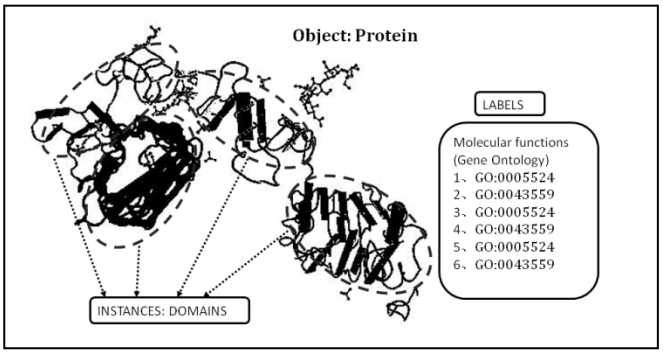


Fig. 1. Illustration of the protein function prediction task with exact matches for Multi-Instance Multi-Label (MIML) learning frameworks. The protein "type 1 I nsulin-like growth factor receptor (PDB ID: 1IGR)" is presented as the illustration where contains four domains (instances) highlighted with dashed oval and six GO Ontology molecular function terms (labels).An object (protein) has many alternative input instances (domains) and output labels (GO molecular function terms), and MIML considers the ambiguity in the input and output spaces simultaneously.

Formally, we denote by $\{(X_i, Y_i)\}$ $(i = 1, 2, \cdots, n)$ the training dataset of n examples, where X_i denotes the i-th protein in the training set, and Y_i denote the Gene Ontology terms assigned to X_i. Further, let D_{ij} denote the j-th domain of the protein X_i . As stated before, each domain can be represented with an input instance and each Gene Ontology term corresponds to an output label. From the machine learning view, X_i is a bag containing multiple instances, and Y_i is the label set of X_i. Thus, with the model trained on the training data, the prediction task is to predict a set of proper labels Y^* for a test bag X^*, which

can be equivilantly formalized as a multi-instance multi-label calssfication problem [5, 6] and thus can be solved by using MIML learning techniques. MIML tries to learn a function $f : 2^X \rightarrow 2^Y$ from a training set $\{(X_i, Y_i)\} = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\}$. It is noteworthy that since there is no explicit relationship between an instance D_{ij} and a label $y_{ij} \in Y_i$, this learning problem is more difficut than conventional supervised learning methods that learns concepts from objects represented by a single instance associated with a single label.

2.2 The EnMIMLNN Learning Framework

MIMLNN is a state-of-the-art Multi-Instance Multi-Label Learning algorithm[6] . As indicated in Figure 2, MIMLNN is trained with a two-layer architecture. In the first layer, the training examples for each class label are clustered by invoking *k*-Medoids and then the medoids of the clustered groups are retained. Then, neural networks are used for computing the basis functions $\Phi(.)$ between one example and the medoids[6]. After that, the model outputs are obtained by multiplying the basis functions with the weights between the first and second layers, and the class labels with output value larger than zero are selected as positive labels.

Formally, let $S = \{(X_i, Y_i) | 1 \le i \le n\}$ be the MIML training set and let $U_l = \{X_i | (X_i, Y_i) \in S, l \in Y_i\}$ denote the set of MIML examples that have the *l*-th label. Thus , S is the union set of Q sets of bags U_l $(1 \leqslant l \leqslant Q)$, where Q denote the number of label classes of S. Given two bags of instances $A = \{a_1, a_2, \ldots, a_{n_a}\}$ and $B = \{b_1, b_2, \ldots, b_{n_b}\}$, the Maximal Hausdorff distance[6] between A and B are used in MIMLNN [6] with the definition as:

$$H(A,B) = \max\{\max_{a\in A}\min_{b\in B}\|a-b\|, \max_{b\in B a\in A}\min_{a\in A}\|a-b\|\} \quad (1)$$

where $\|.\|$ measures the Euclidean distance between two instances.

At the first layer, combined with the metric $H(A,B)$, *k*-Medoids algorithm is employed to partition U_l into M_l disjoint groups of bags G_j^l $(1 \le l \le Q, 1 \le j \le M_l)$, and the number of retained medoids M_l for each label class is set to be fraction α of the number of MIML examples in U_l, i.e., $M_l = \alpha \times |U_l|$, where α is the fraction parameter. The total number of bags in the first layer equals $M = \sum_{l=1}^{Q} M_l$ .The medoids C_j^l of bags G_j^l are intuitively determined as

$$C_j^l = \{\arg\min_{A\in G_j^l} \sum_{B\in G_j^l} H(A,B)\} \quad (1 \le l \le Q, 1 \le j \le M_l) \quad (2)$$

Then, based on the cluster medoids, in this paper the activation of the *j*-th basis function on X_i is implemented by radial basis function (RBF) neural networks instead of back-propagation neural network used in Reference[6]. The defintion is as follows:

$$\Phi_j^l(X_i) = \exp(-\frac{H(X_i, C_j^l)^2}{2\delta^2}) \quad (1 \le i \le n, 1 \le l \le Q, 1 \le j \le M_l) \quad (3)$$

In addtion, the activaiton of $\Phi_0(X_i)$ is fixed at 1. The standard deviation δ is a parameter controlling the smoothness of the basis function $\Phi_j^l(.)$, and δ is some multiple of the average distance between each pair of medoids in the first layer:

$$\delta = \mu \times \left(\frac{\sum_{k=1}^{M-1}\sum_{q=k+1}^{M} H(C_k, C_q)}{M(M-1)/2} \right) \quad (4)$$

where μ is the parameter of scaling factor.

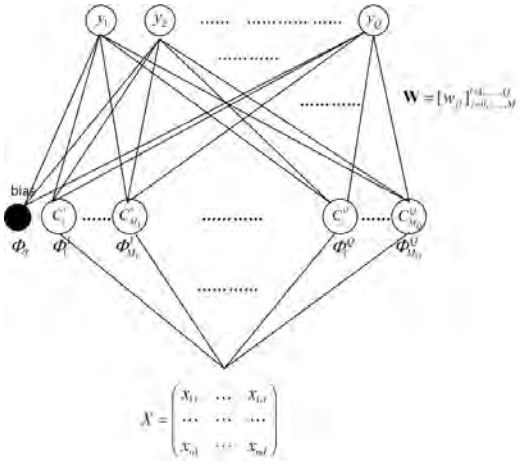


Fig. 2. Architecture of the MIMLNN algorithm.

After that, a weight matrix $W = [w_j^l]$ between the first and second layers of the network maps the basis functions to the outputs:

$$y_l(X_i) = \sum_{j=0}^{M_l} w_j^l \Phi_j^l(X_i) \quad (5)$$

The weight matrix is optimized by minimizing the following sum-of-squares error function:

$$E = \frac{1}{2}\sum_{i=1}^{n}\sum_{l=1}^{Q}\left(y_l(X_i) - t_l^i\right)^2 \quad (6)$$

where t_l^i is the ground-truth of X_i on the *l*-th class, which takes the value of +1 if $l \in Y_i$ and -1 otherwise.

Then, a test example X is fed into the trained model for prediction to get the individual outputs, and the class labels with output value larger than zero are selected as positive labels. Formally, the predicted label subset Y is defined as follows:

$$Y = \{l | \mathrm{y}_l(X) = \sum_{j=0}^{M} w_{jl}\Phi_j(X) > 0, l \in \mathcal{Y}\} \quad (7)$$

However, in the context of the multi-instance problems, an example is a bag that contains multiple instances and therefore does not correspond to a single point. Different criteria on evaluating distances between multi-instance objects intend to have a huge impact on the learning performance[23]. In MIMLNN[6], Maximal Hausdorff distanceis used to calculate distances. Besides, minimal and average Hausdorff distances are another two popular criteria for calculating distance between bags[23, 26]. Minimal Hausdorff distance indicates the minimal distance between all instances of one bag and those of another bag[23], while average Hausdorff distance averages the distances between each instance in one bag and its nearest instance in the other bag[26]. Minimal Hausdorff distance is less sensitive to outlying points than maximal Hausdorff distance[23], while average Hausdorff distance takes more geometric relationships between two bags of instances into consideration than those of maximal and minimal Hausdorff distances[26].Since different domains in the same protein could be very diverse, it may be difficult to acurately measure the distance between proteins using the simple maximal Hausdorff distance. It may be helpful to use multiple Hausdorff distance metrics to describe the distances between proteins and then combine the advantages of these metrics by ensemble learning methods[27]. In this paper, according to the characteristics of protein function prediction tasks, we propose an ensemble multi-instancemulti-label learning framework EnMIMLNN and design three algorithms using metric-based and classifier voting-based ways.

2.2.1 The Metric-based EnMIMLNN Algorthim

Given two bags of instances $A = \{a_1, a_2, \ldots, a_{n_a}\}$ and $B = \{b_1, b_2, \ldots, b_{n_b}\}$, the Hausdorff distances between *A* and *B* are defined as[23, 26]:

$$H^p(A,B) = \begin{cases} \dfrac{\sum_{a\in A}\min_{b\in B}\|a-b\| + \sum_{b\in B}\min_{a\in A}\|a-b\|}{|A|+|B|}, & p = avg; \\ \max\{\max_{a\in A}\min_{b\in B}\|a-b\|, \max_{b\in B a\in A}\min_{a\in A}\|a-b\|\}, & p = max; \\ \min_{a\in A}\min_{b\in B}\|a-b\|, & p = \min; \end{cases} \quad (8)$$

where $|\cdot|$ gives the cardinality of a set, $\|.\|$ measures the Euclidean distance between two instances and $p \in \eta = \{"avg","max","min"\}$.

These Hausdorff distances are known metrics, so each of them is required to satisfy the following conditions: (1) $H^p(A, B) \geq 0$; (2) $H^p(A, B) = 0$ if and only if $A = B$; (3) $H^p(A, B) = H^p(B, A)$; (4) $H^p(A, C) \leq H^p(A, B) + H^p(B, C)$, where $\{A, B, C\} \in S$. By averging the three kinds of Hausdorff distance, we can get a novel ensemble distance as :

$$\tilde{H}(A,B) = \frac{1}{3}\sum\nolimits_{p\in\eta} H^p(A,B) \quad (9)$$

It is obvious that this new distance is also a metric due to its satisfaction of the above four conditions. Table 1 presents the pseudo-code for the metric-based EnMIMLNN algorthims called EnMIMLNN{metric} where the novel ensemble distance $\tilde{H}(A,B)$ is employed to measure distance between bags.

TABLE 1
PSEUDO CODE OF ENMIMLNN{METRIC}

Y= EnMIMLNN{metric} (S,α,μ, X)

Inputs:

S: the MIML training set $\{(X_1,Y_1),\ldots,(X_n,Y_n)\}$

α: the fraction parameter

μ: the scaling factor

X: the test MIML example

Outputs:

Y: the predicted prediction GO terms for X ($Y \subseteq \mathcal{Y}$)

Process:

1 Ensemble the Hausdorff distance metrics by Eq.(9);

2 For $l \in \mathcal{Y}$ do

3 Derive $U_l = \{X_i | (X_i, Y_i) \in S, l \in Y_i\}$;

4 Cluster U_l into $M_l = \alpha \times |U_l|$ subgroups G_j^l ($1 \leqslant j \leqslant M_l$) based on *k*-Medoids with distance Metric $\tilde{H}(A,B)$ of Eq.(9),where α is the fraction parameter ;

5 Calculate the medoid C_j^l of each subgroup using Eq.(2);

6 Compute the kernel matrix Φ *and* weight matrix W ;

7 For a test bag *X*, its GO prediction terms can be obtained by: $Y = \{l | \mathrm{y}_l(X) = \sum w_{jl}\Phi_j(X) > 0, l \in \mathcal{Y}\}$;

2.2.2 The Classfier Voting-based EnMIMLNN Algorthim

We first independently train one MIMLNN based on each distance and combine the outputs of individual classifiers via simple averaging, and then the class labels with average output value larger than zero are selected as positive labels. In this paper, we have proposed two algorthims. The first algorthim is called EnMIMLNN{voting}^{3} where three Hausdorff distances $\{"avg","max","min"\}$ are adopted; another is called EnMIMLNN{voting}^{4} where the ensemble distance $\tilde{H}(A,B)$ is appended. The pseudo-code for EnMIMLNN{voting}^{3} and EnMIMLNN{voting}^{4} are described in Table 2 and Table 3, respectively.

TABLE 2
PSEUDO CODE OF ENMIMLNN{VOTING}^{3}

Y= EnMIMLNN{voting}^{3} (S,α,μ, X)

Inputs:

S: the MIML training set $\{(X_1,Y_1),...,(X_n,Y_n)\}$

α: the fraction parameter

μ: the scaling factor

X: the test MIML example

Outputs:

Y: the predicted prediction GO terms for X ($Y \subseteq \mathcal{Y}$)

Process:

1 For $p \in \eta$ = { Hausdorff distances {"avg","max","min"}} do

2 Train a MIMLNN and calculate the predicted labels $Y^p(X)$;

3 For a test bag X, its GO prediction terms can be obtained by: $Y = \{l|\text{sign}(\sum Y^p(X))>0, p \in \eta, l \in \mathcal{Y}\}$;

TABLE 3
PSEUDO CODE OF ENMIMLNN{VOTING}^{4}

Y= EnMIMLNN{voting}^{4} (S,α,μ, X)

Inputs:

S: the MIML training set $\{(X_1,Y_1),...,(X_n,Y_n)\}$

α: the fraction parameter

μ: the scaling factor

X: the test MIML example

Outputs:

Y: the predicted prediction GO terms for X ($Y \subseteq \mathcal{Y}$)

Process:

1 For $p \in \eta$ = { Hausdorff distances {"avg","max", "min", "ensemble"}} do

2 Train a MIMLNN and calculate the predicted output values by : $Y(X) = \{y_l(X) = \sum w_j^l \Phi_j^l(X)\}$

3 For a test bag X, its GO prediction terms can be obtained by: $Y = \{l|\text{sign}(\sum_p Y(X))>0, p \in \eta, l \in \mathcal{Y}\}$;

3 EXPERIMENTS AND RESULTS

3.1 Experimental Configuration

In this paper, complete proteome on seven real-world organisms covering the biological three-domain system[24] are considered including two bacteria genomes (*Geobacter sulfurreducens*, *Azotobacter vinelandii*), two archaea genomes(*Haloarcula marismortui, Pyrococcus furiosus*) and three eukaryote genomes(*Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophila melanogaster*). For each organism, complete proteome with manually annotated function has been downloaded from the Universal Protein Resource (UniProt) databank[28] (released by April, 2013) by querying the terms of {"organism name" AND "reviewed: yes" AND "keyword: Complete proteome"}.

Redundancy among the protein sequences of each organism is removed by clustering analysis using the blastclust executable program in the BLAST package[29] from NCBI with a threshold of 90% for sequence identity, and a non-redundant dataset is created by retaining only the longest sequence in each cluster for each organism[30]. Then, each non-redundant dataset is uploaded as a *txt* file into the Batch CD-Search servers[31] of NCBI for achieving conserved domains. For each domain, a frequency vector with 216-dimensions is used for its representation where each element denotes the frequency of a triad type[32]. Protein function can be described in multiple ways, and the most well-known and widely used one is Gene Ontology Consortium[25] which provides ontology in three aspects: molecular function, biological process and cellular location. In this study, we focus on the molecular function aspect. We obtain the GO molecular function terms with manual annotation for a protein from the downloaded UniProt format text file. Then, the same strategy as [33] is adopted for prepare label vectors for a protein based on a hierarchal directed acyclic graph (DAG) of GO molecular function, and the latest version (December 2006) of GO function ontology is used as the bases of the functional terms and their relations in this work.

In the MIML learning framework, each protein is represented as a bag of instances where each instance corresponds to a domain and is labeled with a group of GO molecular function terms (multi-labels). Detailed descriptions of datasets, i.e., complete proteome on seven real-world organisms, are summarized in Table 4. For example, there are 379 proteins (examples) with a total of 320 gene ontology terms (label classes) on molecular function in the *Geobacter sulfurreducens* dataset (Table 4). The average number of instances (domains) per bag (protein) is 3.20±1.21, and the average number of labels (GO terms) per example (protein) is 3.14±3.33 (Table 4).

We use three popular multi-label learning evaluation criteria, i.e., Hamming Loss (*HL*), Macro-F1 (*maF1*) and Micro-F1 (*miF1*)[34-36] in this paper. Hamming Loss evaluates how many times on average a bag label pair is incorrectly predicted. The smaller the value of hamming loss, the better the performance. Macro-F1 calculates F1 measure on individual class labels at first, and then averages over all class labels. Macro-F1 is more affected by the performance of the classes containing less examples . The larger the value of Macro-F1, the better the performance. Micro-F1 globally calculates the F1 measure on the predictions over all bags and all class labels. Micro-F1 is more affected by the performance of the classes containing more examples. The larger the value of Micro-F1, the better the performance. The definition of these criteria can be found in [36]. We repeat 10-fold cross validation for ten times and the mean± std. performances are recorded for the proposed and compared methods.

TABLE 4
CHARACTERISTICS OF THE DATA SETS

	Genome	examples	classes	Instances per bag (Mean± std.)	Labels per example (Mean± std.)
Bacteria	*Geobacter sulfurreducens*	379	320	3.20±1.21	3.14±3.33
	Azotobacter vinelandii	407	340	3.07±1.16	4.00±6.97
Archaea	*Haloarcula marismortui*	304	234	3.13±1.09	3.25±3.02
	Pyrococcus furiosus	425	321	3.10±1.09	4.48±6.33
Eukaryota	*Saccharomyces cerevisiae*	3509	1566	1.86±1.36	5.89±11.52
	Caenorhabditis elegans	2512	940	3.39±4.20	6.07±11.25
	Drosophila melanogaster	2605	1035	3.51±3.49	6.02±10.24

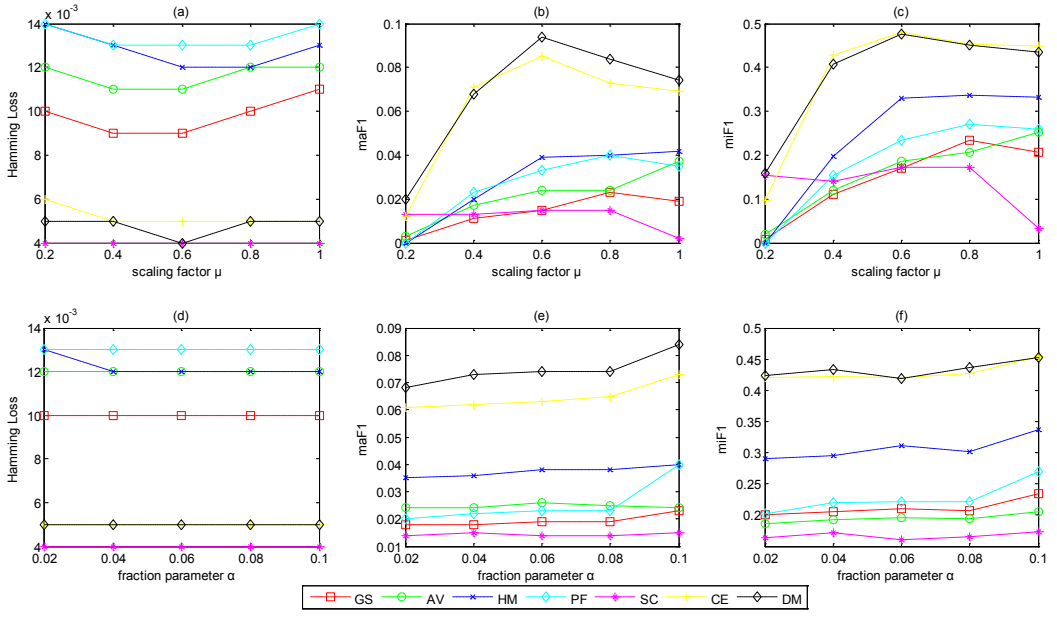


Fig. 3. The performance of EnMIMLNN{metric} on all seven datasets under different values of scaling factor μ when the fraction parameter α is fixed to 0.1 and different values of the fraction parameter α when the scaling factor μ is fixed to 0.8. The performance of EnMIMLNN{metric} reaches the perk in most cases by setting the scaling factor μ to 0.8 and the fraction parameter α to 0.1.*GS: Geobacter sulfurreducens; AV: Azotobacter vinelandii; HM: Haloarcula marismortui; PF: Pyrococcus furiosus; SC: Saccharomyces cerevisiae; CE: Caenorhabditis elegans; DM: Drosophila melanogaster.*

3.2 Parameter Configurations

The ensemble Multi-Instance Multi-Label learning learning framework EnMIMLNN involve two different parameters, i.e., the scaling factor μ and the fraction parameter α. Figure 3 illustrate how the EnMIMLNN{metric} algorithm performs on all seven datasets under different parameter configurations, where the performance is evaluated in terms of *HL*, *maF1* and *miF1*. In addition, the performance of EnMIMLNN{voting}[3] and EnMIMLNN{voting}[4] under different parameter configurations can be reached in the **Supplementary Fig.1 and Supplementary Fig.2, respectively.** Here, μ varies from 0.2 to 1.0 with an interval of 0.2 when α is fixed to 0.1, and α increases from 0.02 to 0.1 with an interval of 0.02 with the fixed μ equal to 0.8. It is shown that the performance of the EnMIMLNN algorithms reach the perk in most cases by setting the scaling factor μ to 0.8 and the fraction parameter α to 0.1 (Figure 3, **Supplementary Fig.1 and Supplementary Fig.2)** . Therefore, in this paper, the EnMIMLNN algorithms all are implemented by setting the scaling factor μ to 0.8 and the fraction parameter α to 0.1.

3.3 Performance of the EnMIMLNN Learning Framework

The EnMIMLNN learning framework combines three different Hausdorff distances, i.e. *average* , *maximal* and *minimal* , for denote the distance between two proteins. Table 5 illustrates the comparison results based on each kind of Hausdorff distance on all datasets. For each evaluation criterion, ↑ (↓) indicates the larger (smaller), the better the performance; the best results on each evaluation criterion are highlighted in boldface. As indicated in Table 5, the results show that the EnMIMLNN algorthims all are obviously better than the MIMLNN models with

TABLE 5
COMPARISON RESULTS (MEAN± STD.) OF ENMIMLNN MODELS BASED ON DIFFERENT HAUSDORFF DISTANCES ON SEVEN DATASETS.

Datasets		Methods	HL↓	maF1↑	miF1↑
Bacteria	*Geobacter sulfurreducens*	EnMIMLNN{metric}	0.010±0.001	**0.023±0.008**	**0.235±0.043**
		EnMIMLNN{voting}^{3}	**0.009±0.002**	0.013±0.006	0.167±0.051
		EnMIMLNN{voting}^{4}	0.010±0.001	0.016±0.007	0.193±0.031
		average	0.010±0.003	0.004±0.004	0.032±0.040
		maximal	0.010±0.002	0.000±0.000	0.000±0.000
		minimal	0.011±0.002	0.019±0.017	0.124±0.083
	Azotobacter vinelandii	EnMIMLNN{metric}	0.012±0.004	**0.024±0.012**	**0.206±0.091**
		EnMIMLNN{voting}^{3}	**0.011±0.004**	0.017±0.009	0.163±0.080
		EnMIMLNN{voting}^{4}	**0.011±0.003**	0.019±0.016	0.159±0.102
		average	**0.011±0.002**	0.008±0.006	0.056±0.044
		maximal	0.012±0.004	0.001±0.001	0.006±0.015
		minimal	0.014±0.003	0.019±0.009	0.116±0.056
Archaea	*Haloarcula marismortui*	EnMIMLNN{metric}	**0.012±0.002**	**0.040±0.019**	**0.337±0.080**
		EnMIMLNN{voting}^{3}	**0.012±0.002**	0.035±0.014	0.286±0.079
		EnMIMLNN{voting}^{4}	**0.012±0.002**	0.036±0.025	0.314±0.102
		average	0.013±0.003	0.006±0.007	0.042±0.053
		maximal	0.014±0.002	0.000±0.000	0.000±0.000
		minimal	0.014±0.004	0.042±0.018	0.244±0.088
	Pyrococcus furiosus	EnMIMLNN{metric}	**0.013±0.003**	**0.040±0.034**	**0.270±0.118**
		EnMIMLNN{voting}^{3}	**0.013±0.004**	0.021±0.011	0.207±0.074
		EnMIMLNN{voting}^{4}	**0.013±0.002**	0.026±0.012	0.229±0.083
		average	0.014±0.003	0.004±0.008	0.023±0.050
		maximal	0.014±0.004	0.000±0.000	0.000±0.000
		minimal	0.015±0.004	0.028±0.021	0.161±0.097
Eukaryota	*Saccharomyces cerevisiae*	EnMIMLNN{metric}	0.004±0.000	0.015±0.004	**0.173±0.034**
		EnMIMLNN{voting}^{3}	0.004±0.000	0.011±0.002	0.153±0.029
		EnMIMLNN{voting}^{4}	0.004±0.000	0.012±0.004	0.163±0.038
		average	0.004±0.000	0.001±0.001	0.031±0.010
		maximal	0.004±0.001	0.000±0.000	0.005±0.005
		minimal	0.004±0.000	**0.018±0.005**	0.159±0.037
	Caenorhabditis elegans	EnMIMLNN{metric}	**0.005±0.000**	0.073±0.012	**0.454±0.039**
		EnMIMLNN{voting}^{3}	**0.005±0.001**	0.073±0.015	0.450±0.052
		EnMIMLNN{voting}^{4}	**0.005±0.001**	0.070±0.019	0.446±0.062
		average	0.006±0.001	0.016±0.006	0.129±0.030
		maximal	0.006±0.001	0.000±0.000	0.004±0.003
		minimal	0.008±0.001	**0.087±0.014**	0.380±0.060
	Drosophila melano-gaster	EnMIMLNN{metric}	0.005±0.001	0.084±0.022	0.452±0.067
		EnMIMLNN{voting}^{3}	**0.004±0.001**	0.083±0.020	0.455±0.045
		EnMIMLNN{voting}^{4}	**0.004±0.001**	0.085±0.026	**0.460±0.050**
		average	0.005±0.000	0.030±0.007	0.208±0.024
		maximal	0.006±0.001	0.001±0.001	0.007±0.006
		minimal	0.008±0.003	**0.088±0.019**	0.363±0.070

↑ (↓) indicates the larger (smaller), the better the performance; the best results on each evaluation criterion are highlighted in boldface.

TABLE 6
COMPARISON RESULTS (MEAN± STD.) WITH SIX STATE-OF-THE-ART MIML METHODS ON SEVEN REAL-WORLD ORGANISMS.

Datasets		Methods	HL↓	maF1↑	miF1↑
Bacteria	*Geobacter sulfurreducens*	EnMIMLNN{metric}	**0.010±0.001**	0.023±0.008	0.235±0.043
		MIMLNN	**0.010±0.002**	0.004±0.001	0.103±0.056
		MIMLSVM+	0.011±0.002	0.008±0.004	0.152±0.036
		En-MIMLSVM	0.012±0.002	0.003±0.002	0.071±0.054
		MIMLBoost	0.011±0.002	0.009±0.003	0.163±0.039
		MIMLkNN	0.092±0.025	0.020±0.007	0.056±0.018
		DBA	0.017±0.003	**0.025±0.009**	0.146±0.034
	Azotobacter vinelandii	EnMIMLNN{metric}	**0.012±0.004**	0.024±0.012	**0.206±0.091**
		MIMLNN	**0.012±0.003**	0.005±0.003	0.069±0.037
		MIMLSVM+	0.013±0.002	0.008±0.004	0.115±0.024
		En-MIMLSVM	0.014±0.002	0.005±0.003	0.045±0.025
		MIMLBoost	0.013±0.004	0.009±0.004	0.121±0.052
		MIMLkNN	0.098±0.027	**0.029±0.015**	0.068±0.032
		DBA	0.020±0.008	0.024±0.012	0.144±0.030
Archaea	*Haloarcula marismortui*	EnMIMLNN{metric}	**0.012±0.002**	**0.040±0.019**	**0.337±0.080**
		MIMLNN	0.013±0.003	0.007±0.003	0.126±0.044
		MIMLSVM+	0.015±0.002	0.013±0.008	0.199±0.060
		En-MIMLSVM	0.017±0.002	0.009±0.007	0.099±0.047
		MIMLBoost	0.015±0.003	0.011±0.005	0.203±0.048
		MIMLkNN	0.080±0.032	0.039±0.015	0.124±0.064
		DBA	0.023±0.004	0.032±0.012	0.165±0.039
	Pyrococcus furiosus	EnMIMLNN{metric}	**0.013±0.003**	**0.040±0.034**	**0.270±0.118**
		MIMLNN	0.014±0.003	0.005±0.003	0.082±0.032
		MIMLSVM+	0.015±0.003	0.007±0.002	0.131±0.031
		En-MIMLSVM	0.017±0.004	0.002±0.001	0.037±0.018
		MIMLBoost	0.015±0.003	0.008±0.003	0.133±0.040
		MIMLkNN	0.085±0.046	0.022±0.009	0.080±0.043
		DBA	0.024±0.003	0.030±0.015	0.138±0.034
Eukaryota	*Saccharomyces cerevisiae*	EnMIMLNN{metric}	**0.004±0.000**	0.015±0.004	**0.173±0.034**
		MIMLNN	**0.004±0.000**	0.005±0.001	0.096±0.009
		MIMLSVM+	**0.004±0.000**	0.002±0.001	0.061±0.007
		En-MIMLSVM	N/A	N/A	N/A
		MIMLBoost	N/A	N/A	N/A
		MIMLkNN	0.026±0.006	0.008±0.003	0.025±0.006
		DBA	0.007±0.001	**0.016±0.002**	0.121±0.008
	Caenorhabditis elegans	EnMIMLNN{metric}	**0.005±0.000**	**0.073±0.012**	**0.454±0.039**
		MIMLNN	0.006±0.001	0.023±0.004	0.288±0.044
		MIMLSVM+	0.006±0.001	0.030±0.010	0.226±0.052
		En-MIMLSVM	N/A	N/A	N/A
		MIMLBoost	N/A	N/A	N/A
		MIMLkNN	0.033±0.006	0.035±0.006	0.088±0.013
		DBA	0.011±0.001	0.052±0.004	0.160±0.009
	Drosophila melanogaster	EnMIMLNN{metric}	**0.005±0.001**	**0.084±0.022**	**0.452±0.067**
		MIMLNN	**0.005±0.001**	0.025±0.005	0.280±0.025
		MIMLSVM+	0.006±0.001	0.025±0.007	0.188±0.032
		En-MIMLSVM	N/A	N/A	N/A
		MIMLBoost	N/A	N/A	N/A
		MIMLkNN	0.032±0.009	0.028±0.008	0.080±0.031
		DBA	0.009±0.001	0.055±0.008	0.201±0.015

↑ (↓) indicates the larger (smaller), the better the performance; the best results on each evaluation criterion are highlighted in boldface. N/A indicates that no result was obtained in 100 hours.

each single metric. Furthermore, the metric-based ensemble method EnMIMLNN{metric} are better than classfier voting-based algorithms EnMIMLNN{voting}[3] and EnMIMLNN{voting}[4] in most cases, and appending of the ensemble Hausdorff distance metric $\tilde{H}(A,B)$ contribute to the performance improvment in the classfier voting-based models in most cases.

3.4 Performance Comparison

In this paper, we compare the EnMIMLNN algorithms with seven state-of-the-art MIML algorithms, i.e., MIMLNN[6], MIMLSVM+[11], En-MIMLSVM[10], MIMLBOOST[5], MIMLkNN[37], DBA[8] and RankLossSIM[9]. Here, only the EnMIMLNN{metric} is shown here as the representation of EnMIMLNN learning framework. The codes of compared MIML algorithms are shared by their authors, and these algorithms are set to the best parameters reported in the papers . Specifically, for MIMLNN, the number of clusters is set to 40% of the training bags, and the regularization parameter used to compute matrix inverse is set to 1[6]; for En-MIMLSVM, Gaussian kernel with width 0.2 is used and the number of clusters is set to 20% of the training bags[10]; for MIMLBOOST, the number of boosting rounds s set to 25[5]; for MIMLkNN, the number of nearest neighbors and the number of citers are set to 10 and 20, respectively[37]; for DBA, the number of iterations and the convergence threshold are set to 100 and 10^(-4), respectively[8]. Table 6 summarize the experimental results of each compared algorithm on seven real-world organisms. For each evaluation criterion,“↓” indicates “the smaller the better” while “↑” indicates “the bigger the bet ter”. Furthermore, the best results on each evaluation criterion are highlighted in boldface. N/A indicates that no result is obtained in 100 hours. The experimental results of RankLossSIM are not shown in Table 6 because no result is obtained in 100 hours even on the smallest dataset *Haloarcula marismortui*. It is shown that the EnMIMLNN algorthims performs quite well in terms of all criteria in most cases (Table 5 and Table 6). Specifically, paired *t*-tests at 95% significance level indicates that the EnMIMLNN{metric} algorthim achieves significantly better perfor mance than compared methods in most cases, as shown by the overwhelming ●'s in Table 6.

Since the protein function prediction was degenerated as a multi-label learning framework by some previous work [16-21] , we compared the ensemble Multi-Instance Multi-Label learning learning framework EnMIMLNN with the state-of-the-art multi-label leaning methods, including ML-*k*NN[38], BSVM [39], BPMLL[40] and RankSVM[41]. By the way, only the EnMIMLNN{metric} is shown here as the representation of EnMIMLNN learning framework. The codes of these algorithms are shared by their authors. For BSVM, the SVM is implemented by LIBSVM[42] package with radial basis function whose parameter "-*g*" selected from $\{2^{-8},2^{-6},.....,2^{6},2^{8}\}$ and parameters "-*c*" searched from $\{2^{-4},2^{-2},.....,2^{6},2^{8}\}$ by ten-fold cross validation. The optimal values for parameters "-*g*" and "-*c*" are 2^{-4} and 2^{4} ,respectively. The compared ML-*k*NN , BPMLL and RankSVM algorithms are set to the best parameters reported in the papers[38, 40, 41]. For ML-*k*NN, the number of nearest neighbors and the smoothing parameter are set to 10 and 1, respectively [38]; for BPMLL, the number of hidden neurons is set to 20% of the input dimensionality, and the training epochs of 100 is used[40]; for RankSVM, polynomial kernels with degree 8 are chosen and the cost parameter *C* is set to be 1[41]. Table 7 summarize the experimental results of each compared algorithm onseven real-world organisms. For each evaluation criterion, ↑ (↓) indicates that the larger (smaller) the value, the better the performance. ●(○) indicates that EnMIMLNN{metric} is significantly better(worse) than the corresponding method based on paired *t*-tests at 95% significance level. Furthermore, the best results on each evaluation criterion are highlighted in boldface. It is shown that the EnMIMLNN algrithms performs quite well in terms of all criteria in most cases (Table 5 , Table 7).

3.5 Comparison with another version

In addition, the EnMIMLNN algrithms replace the back-propagation neural networks in MIMLNN with the radius function based neural networks. We have implemented another version of each EnMIMLNN algorithm to use back-propagation neural networks, and performed experiments to examine the difference between using radial basis function*(rbf)* and back-propagation*(bp)* neural networks. Table 8 summarize the performance on seven real-world organisms. For each evaluation criterion,“↓” indicates “the smaller the better” while “↑” indicates “the bigger the better”. It is shown from Table 8 that each EnMIMLNN algorithm performs better than its *bp* version in terms of all criteria in most cases. Specifically, paired *t*-tests at 95% significance level also indicates that each EnMIMLNN algorithm achieves significantly better performance than its *bp* version in most cases, as shown by the overwhelming ●'s in Table 8.

3.6 Efficiency Comparison

We record the average run time on seven real-world or ganisms in Table 9. The experiments are implemented on Windows 7 platform (x64) with 4*2.6G CPU processor and 8GB memory. It is shown that the run time of the EnMIMLNN algorithms all are ranking in the top amongst all compared methods. Moreover, the EnMIMLNN algorithms generally achieve significantly better performance than the compared methods.

4 Conclusion

In this paper, we disclose that the protein function prediction problem is naturally and inherently Multi-Instance Multi-Label learning tasks. Then, based on a state-of-the-art MIML algorithm MIMLNN[6],we propose a novel ensemble Multi-Instance Multi-Label learning framework called EnMIMLNN to deal with the prediction task by combining the advanatage of different distance metrics and by replacing back-propagation neural network in MIMLNN with radial basis function neural networks. Under this learning framework, we realize three algorithms based on two ensemble ways, i.e., metric-based and classifier-based ways. Experiments on seven real-world organisms covering the biological three-domain system[24], *i.e.*, archaea, bacteria, and eukaryote, show that the EnMIMLNN algorithms are

TABLE 7
COMPARISON RESULTS (MEAN± STD.) WITH FOUR STATE-OF-THE-ART MULTI-LABEL LEARNING METHODS.

Datasets		Methods	HL↓	maF1↑	miF1↑
Bacteria	*Geobacter sulfurreducens*	EnMIMLNN{metric}	**0.010±0.001**	0.023±0.008	**0.235±0.043**
		MLkNN	**0.010±0.002**	0.003±0.002 ●	0.065±0.054 ●
		BSVM	**0.010±0.002**	**0.136±0.046** ○	0.151±0.052 ●
		BPMLL	0.133±0.045 ●	0.011±0.004	0.036±0.016 ●
		RankSVM	0.011±0.002	0.001±0.000 ●	0.126±0.053 ●
	Azotobacter vinelandii	EnMIMLNN{metric}	**0.012±0.004**	0.024±0.012	**0.206±0.091**
		MLkNN	**0.012±0.003**	0.002±0.001 ●	0.033±0.024 ●
		BSVM	0.013±0.003	**0.121±0.045** ○	0.130±0.052 ●
		BPMLL	0.096±0.042 ●	0.013±0.006 ●	0.042±0.014 ●
		RankSVM	0.013±0.002	0.001±0.000 ●	0.085±0.026 ●
Archaea	*Haloarcula marismortui*	EnMIMLNN{metric}	**0.012±0.002**	0.040±0.019	**0.337±0.080**
		MLkNN	0.014±0.003	0.007±0.004 ●	0.114±0.049 ●
		BSVM	0.014±0.003	**0.178±0.069** ○	0.221±0.060 ●
		BPMLL	0.112±0.043 ●	0.020±0.010 ●	0.061±0.016 ●
		RankSVM	0.016±0.001 ●	0.002±0.000 ●	0.116±0.035 ●
	Pyrococcus furiosus	EnMIMLNN{metric}	**0.013±0.003**	0.040±0.034	**0.270±0.118**
		MLkNN	0.014±0.004	0.002±0.001 ●	0.046±0.024 ●
		BSVM	0.015±0.002	**0.133±0.026** ○	0.138±0.037 ●
		BPMLL	0.067±0.034 ●	0.018±0.008 ●	0.093±0.041 •
		RankSVM	0.017±0.002 ●	0.002±0.001 ●	0.123±0.036 ●
Eukaryota	*Saccharomyces cerevisiae*	EnMIMLNN{metric}	**0.004±0.000**	0.015±0.004	**0.173±0.034**
		MLkNN	**0.004±0.000**	0.000±0.000 ●	0.005±0.003 ●
		BSVM	**0.004±0.000**	**0.054±0.009** ○	0.064±0.018 ●
		BPMLL	N/A	N/A	N/A
		RankSVM	N/A	N/A	N/A
	Caenorhabditis elegans	EnMIMLNN{metric}	**0.005±0.000**	0.073±0.012	**0.454±0.039**
		MLkNN	0.006±0.001	0.004±0.002 ●	0.024±0.006 ●
		BSVM	0.006±0.001	**0.166±0.025** ○	0.218±0.045 ●
		BPMLL	N/A	N/A	N/A
		RankSVM	N/A	N/A	N/A
	Drosophila melanogaster	EnMIMLNN{metric}	**0.005±0.001**	0.084±0.022	**0.452±0.067**
		MLkNN	0.006±0.001	0.009±0.004 ●	0.074±0.016 ●
		BSVM	0.006±0.001	**0.210±0.030** ○	0.264±0.048 ●
		BPMLL	N/A	N/A	N/A
		RankSVM	N/A	N/A	N/A

↑ (↓) indicates the larger (smaller), the better the performance; ●(○) indicates the compared model is significantly worse (better) than EnMIMLNN algorithm based on paired t-tests at 95% significance level; the best results on each evaluation criterion are highlighted in boldface. N/A indicates that no result was obtained in 100 hours.

TABLE 8
COMPARISON RESULTS (MEAN± STD.) WITH THEIR CORRESPONDING VARIANTS.

Datasets		Methods	HL↓	maF1↑	miF1↑
Bacteria	*Geobacter sulfurreducens*	EnMIMLNN{voting}^{3}	0.009±0.002	0.013±0.006	0.167±0.051
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.010±0.002	0.004±0.002●	0.096±0.027●
		EnMIMLNN{voting}^{4}	0.010±0.001	0.016±0.007	0.193±0.031
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.009±0.002	0.007±0.003●	0.115±0.040●
		EnMIMLNN{metric}	0.010±0.001	0.023±0.008	0.235±0.043
		EnMIMLNN{metric}$^{\text{bp}}$	0.009±0.002	0.013±0.007●	0.175±0.047●
	Azotobacter vinelandii	EnMIMLNN{voting}^{3}	0.011±0.004	0.017±0.009	0.163±0.080
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.012±0.003	0.005±0.003●	0.065±0.038●
		EnMIMLNN{voting}^{4}	0.011±0.003	0.019±0.016	0.159±0.102
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.011±0.003	0.014±0.007●	0.134±0.045●
		EnMIMLNN{metric}	0.012±0.004	0.024±0.012	0.206±0.091
		EnMIMLNN{metric}$^{\text{bp}}$	0.011±0.003	0.022±0.011	0.188±0.060●
Archaea	*Haloarcula marismortui*	EnMIMLNN{voting}^{3}	0.012±0.002	0.035±0.014	0.286±0.079
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.014±0.002	0.006±0.001●	0.122±0.055●
		EnMIMLNN{voting}^{4}	0.012±0.002	0.036±0.025	0.314±0.102
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.013±0.002	0.017±0.010●	0.198±0.054●
		EnMIMLNN{metric}	0.012±0.002	0.040±0.019	0.337±0.080
		EnMIMLNN{metric}$^{\text{bp}}$	0.013±0.002	0.019±0.010●	0.222±0.095●
	Pyrococcus furiosus	EnMIMLNN{voting}^{3}	0.013±0.004	0.021±0.011	0.207±0.074
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.014±0.002	0.005±0.002●	0.076±0.044●
		EnMIMLNN{voting}^{4}	0.013±0.002	0.026±0.012	0.229±0.083
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.013±0.002	0.010±0.006●	0.107±0.045●
		EnMIMLNN{metric}	0.013±0.003	0.040±0.034	0.270±0.118
		EnMIMLNN{metric}$^{\text{bp}}$	0.013±0.003	0.029±0.026●	0.209±0.109●
Eukaryota	*Saccharomyces cerevisiae*	EnMIMLNN{voting}^{3}	0.004±0.000	0.011±0.002	0.153±0.029
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.004±0.000	0.003±0.001●	0.073±0.016●
		EnMIMLNN{voting}^{4}	0.004±0.000	0.012±0.004	0.163±0.038
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.004±0.000	0.009±0.002●	0.138±0.022●
		EnMIMLNN{metric}	0.004±0.000	0.015±0.004	0.173±0.034
		EnMIMLNN{metric}$^{\text{bp}}$	0.004±0.001	0.011±0.002●	0.150±0.024●
	Caenorhabditis elegans	EnMIMLNN{voting}^{3}	0.005±0.001	0.073±0.015	0.450±0.052
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.008±0.003●	0.011±0.002●	0.167±0.034●
		EnMIMLNN{voting}^{4}	0.005±0.001	0.070±0.019	0.446±0.062
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.005±0.001	0.061±0.011●	0.443±0.051
		EnMIMLNN{metric}	0.005±0.000	0.073±0.012	0.454±0.039
		EnMIMLNN{metric}$^{\text{bp}}$	0.005±0.000	0.066±0.010●	0.453±0.031
	Drosophila melanogaster	EnMIMLNN{voting}^{3}	0.004±0.001	0.083±0.020	0.455±0.045
		EnMIMLNN{voting}$^{3\text{-bp}}$	0.009±0.002●	0.010±0.003●	0.156±0.033●
		EnMIMLNN{voting}^{4}	0.004±0.001	0.085±0.026	0.460±0.050
		EnMIMLNN{voting}$^{4\text{-bp}}$	0.005±0.001	0.066±0.021●	0.444±0.025●
		EnMIMLNN{metric}	0.005±0.001	0.084±0.022	0.452±0.067
		EnMIMLNN{metric}$^{\text{bp}}$	0.005±0.001	0.056±0.011●	0.404±0.044●

↑ (↓) indicates the larger (smaller), the better the performance; ● indicates the corresponding variant is significantly worse than its initial algorithm based on paired t-tests at 95% significance level.

TABLE 9
RUNTIME COMPARISON (IN SECONDS).

Datasets		*Geobacter sulfurreducens*	*Azotobacter vinelandii*	*Haloarcula marismortui*	*Pyrococcus furiosus*	*Saccharomyces cerevisiae*	*Caenorhabditis elegans*	*Drosophila melanogaster*
MIML methods	EnMIMLNN{metric}	121	139	93	171	5283	4532	4558
	EnMIMLNN{voting}^{3}	201	234	134	208	7930	7581	8794
	EnMIMLNN{voting}^{4}	276	383	242	514	19305	14765	18345
	MIMLNN	81	87	50	104	3197	4563	5969
	MIMLSVM+	139	248	95	160	306122	43689	52307
	En-MIMLSVM	4581	5305	3585	8991	N/A	N/A	N/A
	MIMLBoost	167448	347494	95997	208884	N/A	N/A	N/A
	MIMLkNN	209	282	145	230	14421	10257	10340
	DBA	1455	1505	822	1683	86772	79932	103434
	EnMIMLNN{voting}$^{3\text{-bp}}$	167	195	165	209	9264	13652	15281
	EnMIMLNN{voting}$^{4\text{-bp}}$	213	239	157	289	71336	88049	87171
	EnMIMLNN{metric}^{bp}	102	115	68	123	71800	86404	83777
ML methods	MLkNN	22	24	14	24	1007	515	541
	BSVM	105	144	65	150	14333	5816	5702
	BPMLL	31938	36478	22394	36056	N/A	N/A	N/A
	RankSVM	4323	7077	1613	9426	N/A	N/A	N/A

N/A indicates that no result was obtained in 100 hours.

superior to the state-of-the-art MIML and Multi-Label learning algorithms in most cases.

In this paper, we focuse more on the formulation problem of protein function prediction tasks and model construction, and less on the design of descriptive features for representing protein patterns. However, it is also vital for protein function prediction to effectively exploit multiple types of biological data because combining such heterogeneous data can bring a more complete picture about protein function[15]. Thus, one of the future efforts will focus on how to effectively integrate such heterogeneous data into the MIML learning framework for improving the performance of protein function prediction.

ACKNOWLEDGMENT

The authors want to thank the associate editor and anonymous reviewers for helpful comments and suggestions. The authors also wish to thank Min-Ling Zhang, Yu-Feng Li and Shu-Jun Yang for their helpful discussions. This research was supported by the National Key Basic Research Program of China (2010CB327903), China Postdoctoral Science Foundation (20110490129, 2013T60523) and the National Science Foundation of China (61203289). Z.-H. Zhou is the corresponding author of this paper.

REFERENCES

[1] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, G. Pandey, J. M. Yunes, A. S. Talwalkar, S. Repo, M. L. Souza, D. Piovesan, R. Casadio, Z. Wang, J. Cheng, H. Fang, J. Gough, P. Koskinen, P. Toronen, J. Nokso-Koivisto, L. Holm, D. Cozzetto, D. W. Buchan, K. Bryson, D. T. Jones, B. Limaye, H. Inamdar, A. Datta, S. K. Manjari, R. Joshi, M. Chitale, D. Kihara, A. M. Lisewski, S. Erdin, E. Venner, O. Lichtarge, R. Rentzsch, H. Yang, A. E. Romero, P. Bhat, A. Paccanaro, T. Hamp, R. Kassner, S. Seemayer, E. Vicedo, C. Schaefer, D. Achten, F. Auer, A. Boehm, T. Braun, M. Hecht, M. Heron, P. Honigschmid, T. A. Hopf, S. Kaufmann, M. Kiening, D. Krompass, C. Landerer, Y. Mahlich, M. Roos, J. Bjorne, T. Salakoski, A. Wong, H. Shatkay, F. Gatzmann, I. Sommer, M. N. Wass, M. J. Sternberg, N. Skunca, F. Supek, M. Bosnjak, P. Panov, S. Dzeroski, T. Smuc, Y. A. Kourmpetis, A. D. van Dijk, C. J. ter Braak, Y. Zhou, Q. Gong, X. Dong, W. Tian, M. Falda, P. Fontana, E. Lavezzo, B. Di Camillo, S. Toppo, L. Lan, N. Djuric, Y. Guo, S. Vucetic, A. Bairoch, M. Linial, P. C. Babbitt, S. E. Brenner, C. Orengo, B. Rost, S. D. Mooney and I. Friedberg, "A large-scale evaluation of computational protein function prediction," *Nat Methods,* vol. 10, pp. 221-7, Mar 2013.

[2] C. Chothia, "Proteins. One thousand families for the molecular biologist," *Nature,* vol. 357, pp. 543-4, Jun 18 1992.

[3] G. Apic, J. Gough, and S. A. Teichmann, "Domain combinations in archaeal, eubacterial and eukaryotic proteomes," *J Mol Biol,* vol. 310, pp. 311-25, Jul 6 2001.

[4] C. J. Tsai and R. Nussinov, "Hydrophobic folding units derived from dissimilar monomer structures and their interactions," *Protein science,* vol. 6, pp. 24-42, 1997.

[5] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Advances in Neural Information Processing Systems19*, 2007, pp. 1609-1616.

[6] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artificial Intelligence,* vol. 176, pp. 2291-2320, 2012.

[7] N. Nguyen, "A new svm approach to multi-instance multi-label learning," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, 2010, pp. 384-392.

[8] S.-H. Yang, H. Zha, and B.-G. Hu, "Dirichlet-bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora," in *Advances in neural information processing systems*, 2009, pp. 2143-2150.

[9] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 534-542.

[10] X.-S. Xu, X. Xue, and Z.-H. Zhou, "Ensemble multi-instance multi-label learning approach for video annotation task," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 1153-1156.

[11] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, "Drosophila gene expression pattern annotation through multi-instance multi-label learning," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on,* vol. 9, pp. 98-112, 2012.

[12] S. Erdin, A. M. Lisewski, and O. Lichtarge, "Protein function prediction: towards integration of similarity metrics," *Curr Opin Struct Biol,* vol. 21, pp. 180-8, Apr 2011.

[13] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae)," *Proc Natl Acad Sci U S A,* vol. 100, pp. 8348-53, Jul 8 2003.

[14] M. N. Wass, G. Barton, and M. J. Sternberg, "CombFunc: predicting protein function using heterogeneous data sources," *Nucleic Acids Res,* vol. 40, pp. W466-70, Jul 2012.

[15] L. Lan, N. Djuric, Y. Guo, and S. Vucetic, "MS-kNN: protein function prediction by integrating multiple data sources," *BMC Bioinformatics,* vol. 14 Suppl 3, p. S8, 2013.

[16] W. T. Clark and P. Radivojac, "Analysis of protein function and its prediction from amino acid sequence," *Proteins,* vol. 79, pp. 2086-96, Jul 2011.

[17] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, and Z. Yu, "Transductive multi-label ensemble classification for protein function prediction," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1077-1085.

[18] J. Q. Jiang and L. J. McQuay, "Predicting protein function by multi-label correlated semi-supervised learning," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 9, pp. 1059-69, Jul-Aug 2012.

[19] X.-F. Zhang and D.-Q. Dai, "A framework for incorporating functional interrelationships into protein function prediction algorithms," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* vol. 9, pp. 740-753, 2012.

[20] S. Mostafavi and Q. Morris, "Fast integration of heterogeneous data sources for predicting gene function with limited annotation," *Bioinformatics,* vol. 26, pp. 1759-65, Jul 15 2010.

[21] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Zhang, "Protein function prediction by integrating multiple kernels," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1869-1875.

[22] C. M. Bishop, *Neural networks for pattern recognition*: Oxford university press, 1995.

[23] J. Wang and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 1119-1125.

[24] C. R. Woese, O. Kandler, and M. L. Wheelis, "Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya," *Proc Natl Acad Sci U S A,* vol. 87, pp. 4576-9, Jun 1990.

[25] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nat Genet,* vol. 25, pp. 25-9, May 2000.

[26] M.-L. Zhang and Z.-H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Applied Intelligence,* vol. 31, pp. 47-68, 2009.

[27] Z.-H. Zhou, " Ensemble Methods: Foundations and Algorithms," *Boca Raton, FL: Chapman & Hall/CRC,* 2012.

[28] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L. S. Yeh, "UniProt: the Universal Protein knowledgebase," *Nucleic Acids Res,* vol. 32, pp. D115-9, Jan 1 2004.

[29] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol,* vol. 215, pp. 403-10, Oct 5 1990.

[30] J. Wu, H. Liu, X. Duan, Y. Ding, H. Wu, Y. Bai, and X. Sun, "Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature," *Bioinformatics,* vol. 25, pp. 30-5, Jan 1 2009.

[31] A. Marchler-Bauer, S. Lu, J. B. Anderson, F. Chitsaz, M. K. Derbyshire, C. DeWeese-Scott, J. H. Fong, L. Y. Geer, R. C. Geer, and N. R. Gonzales, "CDD: a C onserved Domain Database for the functional annotation of proteins," *Nucleic acids research,* vol. 39, pp. D225-D229, 2011.

[32] J. Wu, D. Hu, X. Xu, Y. Ding, S. Yan, and X. Sun, "A novel method for quantitatively predicting non-covalent interactions from protein and nucleic acid sequence," *J Mol Graph Model,* vol. 31, pp. 28-34, Nov 2011.

[33] O. S. Sarac, V. Atalay, and R. Cetin-Atalay, "GOPred: GO molecular function prediction by combined classifiers," *PLoS One,* vol. 5, p. e12382, 2010.

[34] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, pp. 195-200.

[35] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, pp. 659-661.

[36] S.-J. Yang, Y. Jiang, and Z.-H. Zhou, "Multi-instance multi-label learning with weak label," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1862-1868.

[37] M.-L. Zhang, "A k-nearest neighbor based multi-instance multi-label learning algorithm," in *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, 2010, pp.

207-212.

[38] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition,* vol. 40, pp. 2038-2048, 2007.

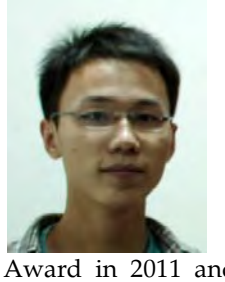
[39] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition,* vol. 37, pp. 1757-1771, 2004.

[40] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 18, pp. 1338-1351, 2006.

[41] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2001, pp. 681-687.

[42] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

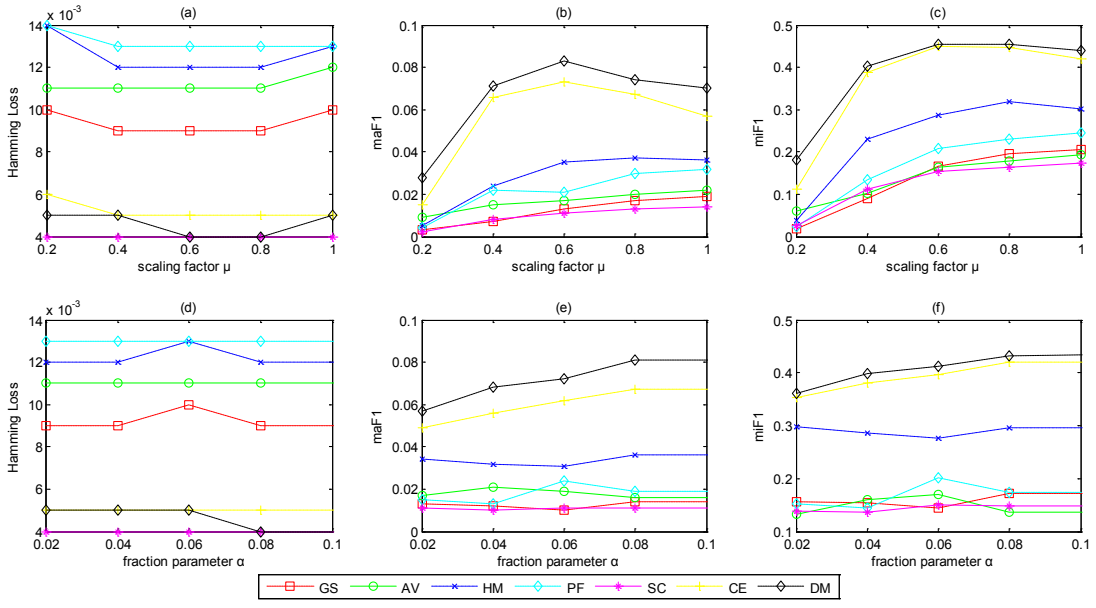
Jian-Sheng Wu received the BS, MS and PhD degrees in bioengineering, ecology, biomedical engineering from nanchang University, east china normal university, southeast university, China, in 2000, 2004, and 2009, respectively. He joined the School of Geography and Biological Information of Nanjing University of Posts and Telecommunications, China, in 2009. Currently, He is also a postdoctoral fellow of the National Key Laboratory for Novel Software Technology in Nanjing University and a member of LAMDA group. His research interests are mainly in machine learning and bioinformatics. In these areas he has published more than 10 papers in leading journals and conference proceedings.



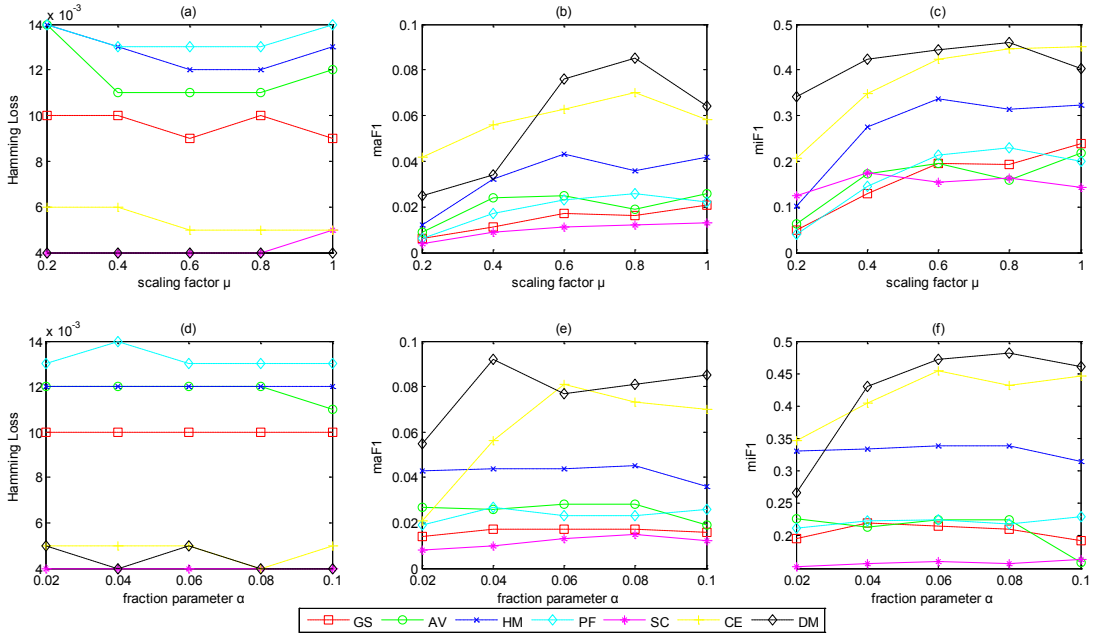
Sheng-Jun Huang is a PhD student in the Department of Computer Science & Technology of Nanjing University. He received the BSc degree from Nanjing University, China, in 2008. His main research interests include machine learning and data mining. He won the Microsoft Fellowship Award in 2011 and the best poster award at the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) in 2012.

Zhi-Hua Zhou (S'00-M'01-SM'06-F'13) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as an assistant professor in 2001, and is currently professor and Director of the LAMDA group. His research interests are mainly in artificial intelligence, machine learning, data mining, pattern recognition and multimedia information retrieval. In these areas he has published more than 100 papers in leading international journals or conference proceedings, and holds 12 patents. He has won various awards/honors including the IEEE CIS Outstanding Early Career Award, the National Science & Technology Award for Young Scholars of China, the Fok Ying Tung Young Professorship Award, the Microsoft Young Professorship Award and nine international journals/conferences paper or competition awards. He is an Executive Editor-in-Chief of the Frontiers of Computer Science, Associate Editor-in-Chief of the Chinese Science Bulletin, Associate Editor or editorial boards member of the ACM Transactions on Intelligent Systems and Technology, IEEE Transactions on Neural Networks and Learning Systems, etc. He served as Associate Editor for IEEE Transactions on Knowledge and Data Engineering (2008-2012) and Knowledge and Information Systems (2003-2008). He is the founder and Steering Committee Chair of ACML, and Steering Committee member of PAKDD and PRICAI. He serves/ed as General Chair/Co-chair of ACML12, ADMA12, PCM13, PAKDD14, Program Chair/Co-Chair of PAKDD07, PRICAI08, ACML09, SDM13, etc., Workshop Chair/Co-Chair of KDD12 and ICDM14, Tutorial Chair/Co-Chair of KDD13 and CIKM14, and Program Vice Chair or Area Chair of various conferences such as ICML, IJCAI, AAAI, ICPR, etc. He is the Chair of the Machine Learning Technical Committee of the Chinese Association of Artificial Intelligence, Chair of the Artificial Intelligence& Pattern Recognition Technical Committee of the China Computer Federation, Vice Chair of the Data Mining Technical Committee of IEEE Computational Intelligence Society and the Chair of the IEEE Computer Society Nanjing Chapter. He is an IEEE Fellow, IAPR Fellow, IET/IEE Fellow and ACM Distinguished Scientist.

Supplementary Files



Supplementary Fig. 1. The performance of EnMIMLNN{voting}^{3} on all seven datasets under different values of scaling factor μ when the fraction parameter α is fixed to 0.1 and different values of the fraction parameter α when the scaling factor μ is fixed to 0.8. The performance of EnMIMLNN{voting}^{3} reaches the perk in most cases by setting the scaling factor μ to 0.8 and the fraction parameter α to 0.1.*GS: Geobacter sulfurreducens; AV: Azotobacter vinelandii; HM: Haloarcula marismortui; PF: Pyrococcus furiosus; SC: Saccharomyces cerevisiae; CE: Caenorhabditis elegans; DM: Drosophila melanogaster.*



Supplementary Fig. 2. The performance of EnMIMLNN{voting}^{4} on all seven datasets under different values of scaling factor μ when the fraction parameter α is fixed to 0.1 and different values of the fraction parameter α when the scaling factor μ is fixed to 0.8. The performance of EnMIMLNN{voting}^{4} reaches the perk in most cases by setting the scaling factor μ to 0.8 and the fraction parameter α to 0.1.*GS: Geobacter sulfurreducens; AV: Azotobacter vinelandii; HM: Haloarcula marismortui; PF: Pyrococcus furiosus; SC: Saccharomyces cerevisiae; CE: Caenorhabditis elegans; DM: Drosophila melanogaster.*