This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

hMuLab: a Biomedical Hybrid MUlti-LABel Classifier Based on Multiple Linear Regression

Pu Wang, Ruiquan Ge, Xuan Xiao, Manli Zhou, and Fengfeng Zhou, *Senior Member* of *IEEE*

Abstract— Many biomedical classification problems are multi-label by nature, *e.g.* a gene involved in a variety of functions and a patient with multiple diseases. The majority of existing classification algorithms assumes each sample with only one class label, and the multi-label classification problem remains to be a challenge for biomedical researchers. This study proposes a novel multi-label learning algorithm, hMuLab, by integrating both feature-based and neighbor-based similarity scores. The multiple linear regression modeling techniques make hMuLab capable to produce multiple label assignments for a query sample. The comparison results over six commonly-used multi-label performance measurements suggest that hMuLab performs accurately and stably for the biomedical datasets, and may serve as a complement to the existing literature.

Index Terms—hMuLab, Multi-label learning, Multiple linear regression, Hybrid method, Feature score, Neighbor score

1 Introduction

COMPARED with the single-label classification problem, multi-label classification is more often to be observed in the real world, and represents a major challenge for machine learning. A single label classification problem assumes that each sample has one and only one class label [1, 2], which is the basis for the majority of existing machine learning algorithms [3-6]. But many real world classification requests have multiple labeling for a sample [7-11], e.g. a picture may be identified as both "sunny" and "barbeque". The biomedical datasets may also label a sample with multiple keywords, *e.g.* a protein with more than one functional annotations [12] or subcellular localizations [13]. These multi-label classification problems are difficult to handle by the existing algorithms, due to that they usually have more than one optimization goals.

Formally, let's give the symbol definition of multi-label learning. Let $X=R^d$ denote the sample space, and $L=\{l_1, l_2, ..., l_c\}$ be the finite label set with c possible class labels. A given sample x ($x \in X$) is a d-dimensional feature vector, and its class label assignment $y \subset L$ is the label subset associated with x. The class label assignment of x may also be represented as a c-dimensional binary vector $\boldsymbol{y}=[b_1, b_2, ..., b_c]$, where $b_i=1$ if the sample x has the label l_i, and $b_i=-1$ if not. A multi-label learning

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB13040400), Shenzhen Peacock Plan (KQCX20130628112914301 and KQCX20130628112914291), the China 863 program (SS2015AA020109-4), Shenzhen Research Grants (JCYJ20130401170306884) and Key Laboratory of Human-Machine-Intelligence Synergic Systems, Chinese Academy of Sciences. This work was also supported by the grants from the National Natural Science Foundation of China (No.31560316, No.31260273, No.61261027, No.61462047 and No.61402209), the Department of Education of JiangXi Province (GJJ13641 and GJJ13636). Computing resources were partly provided by the Dawning supercomputing clusters at SIAT CAS. Constructive comments from the editor and the anonymous reviewers are appreciated.

- P. Wang, R. Ge and M. Zhou are with Shenzhen Institutes of Advanced Technology, and Key Lab for Health Informatics, Chinese Academy of Sciences; and Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, Guangdong 518055, P.R. China.
- P. Wang and X. Xiao are with Computer Department, Jingdezhen Ceramic Institute, Jingdezhen, Jiangxi 333403, PR China.
- F. Zhou is with College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, P.R. China; Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China; and the Shenzhen Institutes of Advanced Technology, and Key Lab for Health Informatics, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, P.R. China. Correspondence may be addressed to F. Zhou at ffzhou@jlu.edu.cn or FengfengZhou@gmail.com .

algorithm tries to find a function h: $X \rightarrow 2^c$ based on the training dataset $D=\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid i=1, 2, ..., n\}$, so that a performance measurement is optimized. The performance measurements of a multi-label learning algorithm are described in the next section. The multi-label learning algorithm usually calculates a real value of how confident a given sample x has a class label l_j as $f(x, l_j)$. The assignment of a class label to a given sample x is determined by the threshold segmentation strategy [14], *i.e.* $h(x)=\{l_j \mid f(x, l_j) \geq t$, where $l_j \in L\}$, and $t \in R$ is a real value as the threshold. A multi-label learning algorithm tries to maximize the difference $f(x, l')$-$f(x, l'')$, where $l' \in y$ and $l'' \notin y$[15].

From above definition, we can see that the multi-label learning is quite different from the single-label learning, no matter the binary or multi-class classification. In recent years, many multi-label learning algorithms have been developed for practical applications. Initially they are mainly the text mining problems [16, 17]. Subsequently many new methods are proposed for image and video annotation [18-20], music categorization [21] and so on. There are also quite a few biomedical multi-label learning problems, such as functional genomes [22, 23], protein subcellular location [11, 13], protein function prediction [24, 25], and medical diagnosis [26-28].

This study proposes a novel multi-label classification algorithm by utilizing both the feature-based information and the sample-based neighbor information. The existing algorithms determine the label assignments based on either feature vectors or the neighboring information, and we hypothesize that both information may complement each other into a better multi-label classifier. The extensive comparisons demonstrate that our algorithm hMuLab outperforms the existing algorithms in most cases, and performs similarly well in all the other cases.

The rest of this study is organized as follows: related work is discussed in section 2; section 3 elaborates the principles and procedure of the proposed algorithm; the experimental datasets, effects of parameters, comparison with the existing methods and result discussion are listed in section 4; at last, section 5 concludes this study.

2 Related Work

2.1 Performance Measurements

Due to its intrinsic difference to the single-label classification problem, the following set of performance measurements are defined to demonstrate how well a multi-label classification algorithm performs, *i.e. Hamming Loss*, *Subset Accuracy*, *One Error*, *Coverage*, *Ranking Loss*, and *Average Precision*, as similar to the existing studies [15, 17, 29, 30]. A better performance is represented by a larger value for the measurements *Subset Accuracy* and *Average Precision*. And all the other four measure-

ments have smaller values for better performances.

Let $S=\{(x_i, y_i) \mid i=1, 2, ..., m\}$ be the testing dataset with m samples, where $y_i \subset Y$ is the class label sets associated with x_i. The measurement *Hamming Loss* is defined as $1/m \sum_{1\le i\le m} \left|h(\boldsymbol{x_i}) \Delta \boldsymbol{y}_i\right| / c$, where Δ is the symmetric difference between two sets, and $|\bullet|$ gives the size of a set. The *Hamming Loss* calculates the percentage of mismatched label pairs.

The measurement *Subset Accuracy* is defined as $1/m \sum_{1\le i\le m} I(h(\boldsymbol{x_i}) = \boldsymbol{y}_i)$, where $I(true)=1$ and $I(false)=0$. The *Subset Accuracy* measures the percentage of samples whose predicted label set is identical with the real label set. This strict measurement evaluates how sound and comprehensive a multi-label classification algorithm performs.

The top-ranked prediction class label of a given sample will usually be regarded as a top candidate for further investigations, and *One Error* is calculated to measure the percentage of samples whose top-ranking labels are incorrect as $1/m \sum_{1\le i\le m} I([\arg\max_{y\in \mathcal{Y}} f(\boldsymbol{x_i}, y)] \notin \boldsymbol{y}_i)$.

The measurement Coverage $1/m \sum_{1\le i\le m} \max_{y\in \boldsymbol{y}_i} rank_f(\boldsymbol{x_i}, y) - 1$ is often calculated to evaluate the average rank to detect all the known class labels of the samples, where $rank_f(x_i, y)$ givens the rank of class label y in sample x_i's predictions, where y is a known label of x_i.

The predicted rankings of a given sample's class labels are supposed to be higher than those of the class labels not associated with this sample. The measurement *Ranking Loss* is calculated to evaluate the fraction of label pairs with incorrect orders, and the overall *Ranking Loss* is averaged over all the samples as $1/m \sum_{1\le i\le m} \left|\{(y', y'') \mid f(\boldsymbol{x_i}, y') \le f(\boldsymbol{x_i}, y''), (y', y'') \in \boldsymbol{y}_i \times \bar{\boldsymbol{y}}_i\}\right| / \left|\boldsymbol{y}_i\right| \left|\bar{\boldsymbol{y}}_i\right|$, where y_i is the subset of class labels associated with sample x_i, and $\bar{\boldsymbol{y}}_i$ is the complementary subset of y_i.

The Average Precision measures the averaged probability that class labels ranking higher than a sample's associated class label are also associated with this sample, and is defined as $1/m \sum_{1\le i\le m} 1/\left|\boldsymbol{y}_i\right| \sum_{y'\in \boldsymbol{y}_i} \left|\{y'' \mid rank_f(\boldsymbol{x_i}, y'') \le f(\boldsymbol{x_i}, y'), y'' \in \boldsymbol{y}_i\}\right| / rank_f(\boldsymbol{x_i}, y')$.

The two measurements *Subset Accuracy* and *Average Precision* are larger for a better multi-label classification algorithm, and *vice versa* for the other four measurements.

2.2 Existing Algorithms

There are two groups of multi-label classification algorithms, *i.e.* reducing and modeling. The *reducing algorithms* formulate

the multi-label classification problem into single-label ones, and utilize the existing single-label classification algorithms to solve these reduced problems. An intuitive reducing strategy is to split all the samples into the positive and negative classes based on whether a sample has the given label, and to build a binary classification model for each of the class labels. The representative algorithm is the Binary Relevance (BR) classifier [15, 20, 31]. The Label Powerset (LP) algorithm revises the BR strategy by converting all the label combinations as new class labels, and considers the original multi-label classification problem as a multi-class single-label problem [30]. Although outperforming the BR classifier, LP doesn't recognize new label combinations in the test dataset. Another challenge for LP is that the model may be overfit, if there are limited samples for a label combination. The RAkEL algorithm randomly splits the initial label set into smaller subsets, and utilizes LP to generate classifiers for each label subset [32]. RAkEL summarizes the final prediction from the sub-classifiers, and the experimental data suggests that RAkEL performs more stably and accurately than the other algorithms. Ensemble strategy is also proposed to generate a series of single-label classifiers, so that these classifiers utilize the prediction results of the preceding ones [33]. Different feature subsets are also suggested to improve the individual sub-classification problems reduced from the multi-label one [34, 35].

The multi-label *modeling algorithms* determine the label assignments without transforming the problem into single-label sub-problems. Instead, they revise the output calculation formula of the existing single-label classification algorithms. AdaBoost is revised into AdaBoost.MH and AdaBoost.MR to solve the multi-label learning problem [36]. AdaBoost.MH minimizes the multi-label classification performance measurement Hamming loss, whereas AdaBoost.MR tries to top-rank the correct labels. The K nearest neighbor algorithm is also updated to assign a class label subset to a query sample by maximizing a posteriori [37]. Rank-SVM is adapted from the Support Vector Machine algorithm to minimize the ranking error [38]. For BPMLL, A new error function is introduced into the classic BP neural network model, so that the optimization objective is to reduce the ranking error of the relative and irrelative labels [39].

3 The Proposed Algorithm hMuLab

A novel multi-label classification algorithm is proposed by integrating the feature score and neighbor score, described in the following sections. Both measurements are based on multiple regressions. Feature score evaluates whether a query sample belongs to a class label based on its features, while neighbor score determines this assignment based on the class labels of this query sample's neighbors. We believe that the two measurements complement each other, and they may work

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

together to generate a better multi-label classifier. The experimental data suggests the necessity of considering both measurements to determine the query sample's class labels.

3.1 Feature score

The measurement *Feature Score*$_1(x, l_j)$ is calculated to evaluate whether the sample x has the class label l_j based on a regression model. Suppose X_1 be the sample matrix, with the i^{th} row being the augmented feature vector of sample x_i, denoted as $[1, x_{i1}, x_{i2}, ..., x_{id}]^T$. Let Y be the class label matrix, and its i^{th} row gives the label vector y_i of sample x_i. The multi-output regression model is defined as,

$$\underset{n\times c}{\mathrm{Y}} = \underset{n\times(d+1)}{\mathrm{X}_1} \times \underset{(d+1)\times c}{\Theta_1} + \underset{n\times c}{\mathrm{U}}, \tag{1}$$

where Θ_1 is the regression coefficient matrix, and U is the residual matrix. The regression coefficients may be calculated by minimizing the residual sum of squares as,

$$\begin{aligned} \min J(\Theta_1) &= \frac{1}{2}\left\|\mathrm{Y}-\mathrm{X}_1\Theta_1\right\|_F^2 + \frac{\lambda}{2}\left\|\Theta_1\right\|_F^2 \\ &= tr\left\{\frac{1}{2}\left(\mathrm{Y}-\mathrm{X}_1\Theta_1\right)^{\mathrm{T}}\left(\mathrm{Y}-\mathrm{X}_1\Theta_1\right)+\frac{\lambda}{2}\Theta_1{}^{\mathrm{T}}\Theta_1\right\}, \end{aligned} \tag{2}$$

where T is the transpose operation, $\left\|\bullet\right\|_F$ indicates the Frobenius norm, and *tr* is used to get the trace of a matrix. On the right-hand side of the equation, the first term is the square of errors, while the second term is the regularization term, which is used to reduce the parameter value and avoid overfitting. The non-negative λ is the tradeoff of the two terms.

The regression coefficient matrix Θ_1 can be determined by setting $\nabla_{\Theta_1} J(\Theta_1) = 0$, and we can get the optimal estimated parameters as,

$$\hat{\Theta}_1 = (\mathrm{X}_1{}^{\mathrm{T}}\mathrm{X}_1 + \lambda\mathrm{I})^{-1}\mathrm{X}_1{}^{\mathrm{T}}\mathrm{Y}, \tag{3}$$

where I is the identity matrix. Because the problem (2) is convex, according to the optimization theory, the stationary point is just the global minimum point. It should be noted that sometimes $\mathrm{X}_1{}^{\mathrm{T}}\mathrm{X}_1$ is not invertible, however if λ is large enough, it will make $(\mathrm{X}_1{}^{\mathrm{T}}\mathrm{X}_1 + \lambda\mathrm{I})$ invertible. Then for any unknown sample x which is also an augmented feature vector, the output for each label may be calculated simply as below,

$$f_1(\boldsymbol{x}, l_j) = \boldsymbol{x}^{\mathrm{T}}\hat{\Theta}_1^j,\ 1 \le j \le c, \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

where Θ_1^j is the j^{th} column of matrix Θ_1. Thus in the frame of minimizing the residual sum of squares, if a sample has label l_j, the output of this label will tend to be 1. By contrast, if it doesn't have the label, then the output will tend to -1. A given class label is assigned to this sample, if its prediction is no smaller than the threshold value 0.

3.2 Neighbor score

A *Neighbor Score*$_2(x, l_j)$ is calculated to evaluate how significantly the neighbors of a given sample x tend to have the class label l_j. K nearest neighbor (KNN) algorithm [40] is one of the most widely used supervised learning algorithm, for its simplicity and intuitiveness. A revised KNN algorithm is proposed to give larger weights to closer neighbors based on the user-defined distance [41]. Let the K nearest neighbors and their class labels of a query sample x be (x_k^*, y_k^*) from the training dataset D, where $1 \le k \le K$. The distances of the sample x to its K nearest neighbors are d_k, where $1 \le k \le K$. The distances are ordered, so that $d_1 \le d_2 \le \ldots \le d_K$. The weight of the sample x's k^{th} nearest neighbor is defined as,

$$w_k = \begin{cases} \dfrac{d_K - d_k}{d_K - d_1}, & d_K \neq d_1 \\ 1, & d_K = d_1 \end{cases}. \tag{5}$$

So the weight $w_k \in [0, 1]$, and a closer neighbor has a larger weight. The first order Minkowski distance is utilized here to measure the distance between two samples. Then a score of each class label for the sample x is calculated as

$$s_j = \frac{\sum_{l_j \in \mathbf{y}_k^*} w_k}{\sum_{1 \le k \le K} w_k}, \quad 1 \le j \le c \tag{6}$$

The definition of this score suggests that a class label l_j tends to have a larger score, if more neighbors of the query sample x have this label. A maximal score 1 may be achieved, if all the x's neighbors have the class label. And none of its neighbors have a class label will give this label a minimum score of 0.

The associations between the labels are considered in determining the class label assignment of a given query sample x. After the scores of the class labels are calculated, a simple way is to assign the class labels independently to the sample x using a threshold for each label. We hypothesize that an integrated assignment of class labels to the sample x by evaluating the scores of all the class labels will perform better, *i.e.* the output of each label was determined by not only its own label score, but also those of the others. Our experimental data demonstrate that the label correlation incorporated in the input of the following regression model performs satisfyingly.

A linear regression model is established to determine the assignment of a given class label to the sample x, based on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

scores of all the class labels. Suppose X_2 be the sample matrix with the i^{th} row corresponding to the sample x_i, which is composed of the scores by distance-weighted KNN and has been augmented so that $\boldsymbol{x}_i = [1 \;\; s_{i1} \;\; s_{i2} \;\; \cdots \;\; s_{ic}]^{\mathrm{T}}$. If Θ_2 is the coefficient matrix, we have the similar regression model just as described in the Feature Score method,

$$\min J(\Theta_2) = \frac{1}{2}\left\|\mathrm{Y} - \mathrm{X}_2\Theta_2\right\|_F^2 + \frac{\lambda}{2}\left\|\Theta_2\right\|_F^2. \tag{7}$$

The optimal Θ_2 is

$$\hat{\Theta}_2 = (\mathrm{X}_2{}^{\mathrm{T}}\mathrm{X}_2 + \lambda \mathrm{I})^{-1}\mathrm{X}_2{}^{\mathrm{T}}\mathrm{Y}, \tag{8}$$

For any unknown sample $\boldsymbol{x}$ which is also the augmented score vector, the output for each label can be calculated by

$$f_2(\boldsymbol{x}, l_j) = \boldsymbol{x}^{\mathrm{T}}\hat{\Theta}_2^j, 1 \le j \le c \tag{9}$$

where Θ_2^j is the j^{th} column of matrix Θ_2.

3.3 Hybrid MUlti-LABel classifier based on Multiple Regression (hMuLab)

The confidence of a class label being assigned to the query sample is quantified as a weighted sum of feature score f_1 and neighbor score f_2 as

$$f(x, l_j)=\alpha f_1(x, l_j)+(1-\alpha)f_2(x, l_j), \tag{10}$$

where $1 \le j \le c$, and $0 \le \alpha \le 1$. The prediction of all the class labels for the query sample is defined as

$$h(x)=\{l_j \mid f(x, l_j) \ge 0, 1 \le j \le c\}. \tag{11}$$

The pseudo-code of the algorithm hMuLab is illustrated in Fig.1, and the default value for the regularization parameter λ is 1.

[h, f] = hMuLab (D, λ, K, α, x)
Inputs:
D: Training dataset (X, Y)
λ: regularization parameter λ in (2)
K: number of neighbors in (6)
α: Hybrid parameter in (10)
x: query sample x
Outputs:
h: predicted label set
f: confidence of each class label
Procedure:
(a) Train:
1. Use (3) to calculate coefficient matrix Θ_1.
2. Create the score vector of training samples according to (5~6), then get coefficient matrix Θ_2 according to (8).

(b) Predict:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

3. Calculate Feature Score according to (4).
4. Create the score vector of the query sample according to (5~6), and get Neighbor Score according to (9).
5. Obtain the final confidence of each class label according to (10), and the predicted label set according to (11).

Fig. 1. Pseudo-code of hMuLab.

4 Experiments and Discussion

4.1 Datasets

This study chooses three biomedical multi-label datasets to evaluate how the proposed algorithm performs. Multi-label learning problem arises in many areas, including but not limited to images, text and music. A number of community web sites host commonly used benchmark datasets, *e.g.* Mulan (http://mulan.sourceforge.net/datasets.html) and Keel (http://sci2s.ugr.es/keel/multilabel.php). There are only three biomedical datasets from the above databases, as summarized in Table 1. Detailed information of each dataset is described in details in the following paragraphs.

TABLE 1
Biomedical multi-label datasets and their statistics. The columns *Example* and *Feature* are the numbers of samples and features, respectively. In the column *Feature*, *n* indicates numeric features, and *b* indicates binary features. Column *Label* gives the number of class labels for the dataset. The measurement *Label Cardinality*[30] in the column *LC* gives the average number of labels per sample.

Name	Example	Feature	Label	LC
Yeast	2417	103*n*	14	4.2371
Genbase	662	1185*b*	27	1.2523
Medical	978	1449*b*	45	1.2444

The dataset *Yeast* [38] consists of 2,417 yeast genes, and each gene has 103 features, including both the microarray-based expression levels and phylogenetic profiling values. Each gene is annotated to belong to one or more of the 14 function classes.

Genbase [42] describes the functional annotations of 662 proteins. Each protein has 1,185 binary values to describe the existences of 1,185 motifs in this protein. A protein may belong to one or more of the 27 functional classes.

The third dataset Medical [43] was derived from the Medical Natural Language Processing Challenge 2007 in the Computational Medicine Center of the University of Waikato, New Zealand. It consists of 978 clinical records in the text format. Each text record was formatted into 1,449 binary features to describe the existences of 1,449 key words. Each clinical record is diagnosed with one or more of the 45 candidate disease codes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

The two parameters K and • are independently optimized in the following sections, due to the high computation requirement of the grid-based joint-optimization of the two parameters.

4.2 Parameter *K* for the neighbor score

Just like the classical KNN rule, the number of nearest neighbors K is the major factor influencing the performance of the neighbor-driven method. Taking the *Yeast* dataset for example, 20 repeats of ten-fold cross-validations (10-CV) are conducted for evaluating the performance of neighbor score, *i.e.* α=0. Fig.2 shows the mean values of different evaluation measurements with K=1, 5, 10, ..., 40. As we can see, at the beginning, all the six performance measurements are improved significantly along with an increasing number of nearest neighbors. The measurement *Subset Accuracy* gets worse after K=15, and the other five measurements receive little improvements after K=20. Similar patterns are observed for the dataset Medical, as shown in the Supplementary Fig.S1. The algorithm hMuLab achieves the best performances with K=5 on the dataset GenBase, and gets slightly worse performance measurements with K=15. So the following sections will use K=15 as the default value, and the experimental data in the following sections demonstrates satisfying performances compared with the other algorithms.

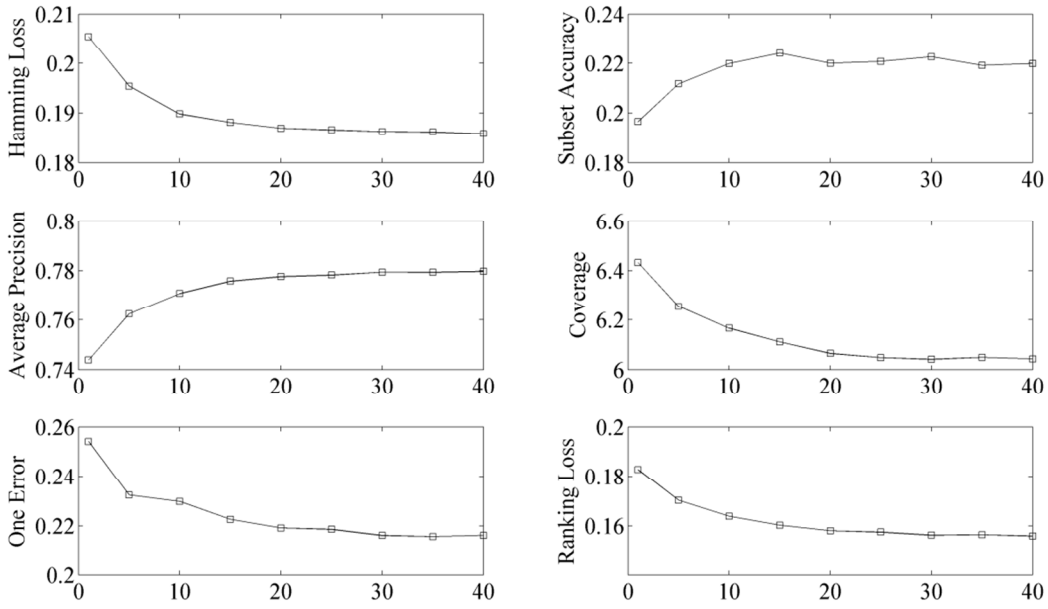

Fig. 2. Six measurements of hMuLab averaged over the 20 runs on the dataset *Yeast* with different number of neighbors K. The horizontal axis is the number of neighbors, and the vertical axis represents the average measurement value by 10-CV.

4.3 Parameter α for hMuLab

hMuLab uses the parameter α to adjust the weights of the two scores based on features and neighbors. If α=1, the hybrid algorithm assigns the labels to a sample based only on the feature score. While α=0 makes hMuLab use only the neighbor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

score. So an investigation is conducted to evaluate how hMuLab performs with different α. 20 repeats of 10-fold cross validations are conducted for each of the values α=0, 0.25, 0.50, 0.75, 1. The six performance measurements averaged over the 20 runs are illustrated in Fig. 3.

Either one of the feature score and neighbor score almost always performs worse than their collaborating model, *i.e.* $\alpha \neq 0$ and $\alpha \neq 1$. Except for the measurement *Subset Accuracy*, all the other five measurements achieve the best values at α=0.25 or α=0.50, and α=0.50 is always ranked among top two. The neighbor score performs better than the feature score for the measurement *Subset Accuracy*, considering its decreasing value along with the increased α. The best performances are achieved at α=0.75 or α=0.50 for the other two datasets, as shown in the Supplementary Fig.S2. So the rest sections of this study uses α=0.50 as its default value.

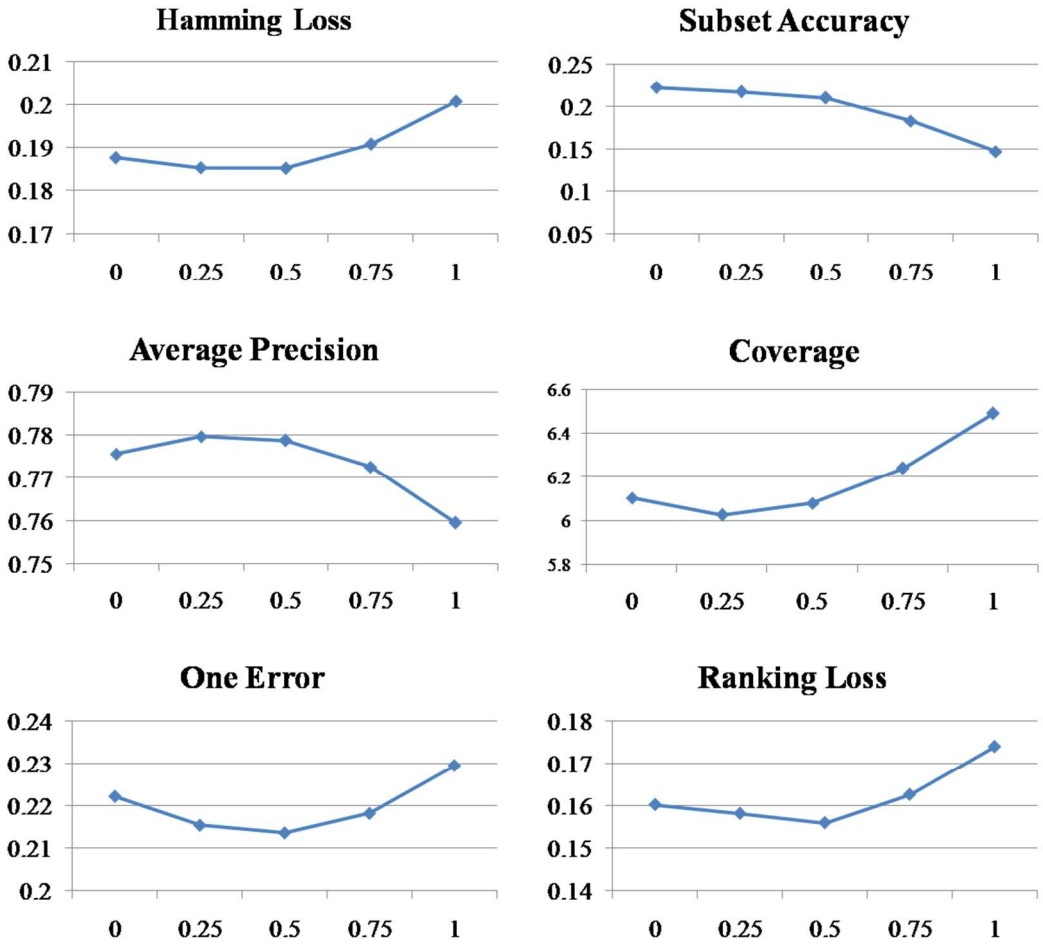

Fig. 3. Six measurements of hMuLab averaged over the 20 runs on the dataset *Yeast* with different values of α. The horizontal axis is the value of α, and the vertical axis represents the average measurement value by 10FCV.

It should be noted that the parameter tuning is always not trivial. To maximize the performance of hMuLab, the best practice is to tune the parameters in a grid way, for example optimizing λ in $\{0, 10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, K in $\{1, 5, 10, \ldots, 40\}$, α

in {0, 0.25, 0.5, 0.75, 1}, and find the best combination. However in the following experiments, there are many cross-validations on different datasets, if all the parameters were estimated by grid searching in each round of training and testing, it would be a heavy burden for hMuLab and the others. So we will choose default parameters for simplification, i.e. λ=1, K=15, α=0.5, which is a good choice in most cases by practice.

4.4 Comparison with other methods

A comprehensive investigation is conducted to compare hMuLab with the four existing state-of-the-art algorithms, *i.e.* MLKNN [37], RAkEL [32], ClassifierChain [33] and IBLR [31]. All the four algorithms are implemented in the library Mulan[44] for multi-label learning. J48 is chosen as the base learner of RAkEL and ClassifierChain, and default parameter values of these algorithms are used. 20 repeats of 10-fold cross validations are conducted on the three benchmark datasets, and the averaged results of six performance measurements are listed in Tables 2-7, respectively. For the comparison's purpose, the ranks of the compared six algorithms are listed in the tables for each of the six performance measurements, as did in [32].

In summary, among the six performance measurements on the three biomedical datasets, hMuLab is only outperformed by the algorithms RAkEL and ClassifierChain in the measurements *Hamming Loss* and *Subset Accuracy* on the dataset *Medical*, as shown in Tables 2 and 3. A better algorithm has a smaller *Hamming Loss*, and RAkEL and ClassifierChain achieve 4.72% and 3.77% improvements versus hMuLab, while hMuLab outperforms the next-ranked algorithm MLKNN in 31.61%. The measurement *Subset Accuracy* has a larger value for a better multi-label classifier, and hMuLab is ranked 3. The two better algorithms RAkEL and ClassifierChain outperform hMuLab of 3.03% and 4.72% in *Subset Accuracy*, respectively. And the rank-4 algorithm MLKNN has a *Subset Accuracy* 23.61% worse than hMuLab. hMuLab ranks top in all the other cases.

Quite a number of performance measurements make a complicated comparison of the five algorithms, and a performance triplet of (better/tie/worse) is calculated to summarize the comparison results of two algorithms *AlgA* and *AlgB* over all the three investigated datasets. For a given performance measurement, a paired *t*-test is carried out to evaluate whether 20 runs of *AlgA* performs better or worse than *AlgB* with statistical significance, using the cutoff $Pvalue \leq 0.05$. The two cases are defined as the notations "better" and "worse", respectively. A notation "tie" is to describe the paired comparison with $Pvalue > 0.05$. Table 8 gives the details of these performance comparisons. The data shows that hMuLab outperforms the algorithms MLKNN and IBLR in all the six performance measurements over the three biomedical datasets. As discussed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

above and in Table 8, hMuLab works slightly worse than RAkEL and ClassifierChain in the measurements *Hamming Loss* and *Subset Accuracy* on the dataset Medical. And hMuLab achieves the measurement Coverage with no statistically significant differences compared with RAkEL and ClassiferChain on the dataset Genbase. As detailed above, hMuLab performs the best in all the other cases.

So in summary, hMuLab performs accurately and stably over the biomedical multi-label classification problems. This is mainly due to the integrated modeling of two complementary scoring methods. In the feature score method, the linear decision function was calculated based on the global information in the training data. Whereas the neighbor score method utilized the non-linear summarization of the local label information, which incorporated the label correlations into the model. As a result, the hybrid method hMuLab demonstrates an effective way to integrate different information sources for the multi-label classification problem, and may be a complementary solution to the current algorithms.

TABLE 2
Comparative results in terms of Hamming Loss ↓ (lower is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.1854(1)	0.1928(2)	0.2280(4)	0.2688(5)	0.1933(3)
Genbase	0.0009(1)	0.0046(4)	0.0011(2)	0.0011(2)	0.0020(3)
Medical	0.0106(3)	0.0155(4)	0.0101(1)	0.0102(2)	0.0196(5)
Ave.Rank	1.6667	3.3333	2.3333	3	3.6667

TABLE 3
Comparative results in terms of Subset Accuracy ↑ (higher is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.2094(1)	0.1865(3)	0.1108(5)	0.1439(4)	0.2024(2)
Genbase	0.9782(1)	0.9121(4)	0.9717(2)	0.9717(2)	0.9579(3)
Medical	0.6439(3)	0.4919(4)	0.6634(2)	0.6743(1)	0.4694(5)
Ave.Rank	1.6667	3.6667	3	2.3333	3.3333

TABLE 4
Comparative results in terms of Average Precision ↑ (higher is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.7788(1)	0.7669(3)	0.7151(4)	0.6241(5)	0.7687(2)
Genbase	0.9964(1)	0.9880(5)	0.9905(3)	0.9926(2)	0.9903(4)
Medical	0.8996(1)	0.8081(4)	0.8359(3)	0.8375(2)	0.7561(5)
Ave.Rank	1	4	3.3333	3	3.6667

TABLE 5
Comparative results in terms of Coverage ↓ (lower is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	6.0805(1)	6.2223(3)	7.5208(4)	8.9455(5)	6.2035(2)
Genbase	0.3562(1)	0.5610(5)	0.3671(2)	0.3687(3)	0.4142(4)
Medical	1.2356(1)	2.6658(2)	4.2104(4)	4.6132(5)	3.8626(3)
Ave.Rank	1	3.3333	3.3333	4.3333	3

TABLE 6
Comparative results in terms of One Error ↓ (lower is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.2136(1)	0.2283(3)	0.2865(4)	0.3603(5)	0.2258(2)
Genbase	0.0030(1)	0.0107(5)	0.0091(4)	0.0034(2)	0.0072(3)
Medical	0.1426(1)	0.2508(4)	0.1764(3)	0.1758(2)	0.3151(5)
Ave.Rank	1	4	3.6667	3	3.3333

TABLE 7
Comparative results in terms of Ranking Loss ↓ (lower is better). Numbers in the parentheses are the ranks of the algorithms for each dataset.

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.1559(1)	0.1648(3)	0.2167(4)	0.3298(5)	0.1642(2)
Genbase	0.0020(1)	0.0062(5)	0.0031(3)	0.0030(2)	0.0036(4)
Medical	0.0170(1)	0.0405(2)	0.0740(4)	0.0782(5)	0.0655(3)
Ave.Rank	1	3.3333	3.6667	4	3

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2603507, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 8
The triplet better/tie/worse results for hMuLab against the other algorithms. The statistical significance is evaluated using the paired *t*-test with $Pvalue \leq 0.05$.

Measurement	hMuLab versus			
	MLKNN	**RAkEL**	**ClassifierChain**	**IBLR**
Hamming Loss	3/0/0	2/0/1	2/0/1	3/0/0
Subset Accuracy	3/0/0	2/0/1	2/0/1	3/0/0
Average Precision	3/0/0	3/0/0	3/0/0	3/0/0
Coverage	3/0/0	2/1/0	2/1/0	3/0/0
One Error	3/0/0	3/0/0	3/0/0	3/0/0
Ranking Loss	3/0/0	3/0/0	3/0/0	3/0/0
In total	**18/0/0**	**15/1/2**	**15/1/2**	**18/0/0**

5 Conclusion

In this work, we propose a novel multi-label modeling classifier hMuLab, which is a hybrid model by integrating the feature information and neighbor label information simultaneously, and then a multi-output regression model with regularization is employed for multi-label prediction. The experimental data shows that the two measurements feature score and neighbor score complement each other nicely. Experiments on the three multi-label biomedical datasets indicate that this method is simple in computation while excellent in performance, and it outperforms the existing multi-label learning algorithms in most cases. It should be noted that the three investigated biomedical datasets are representative for their specific patterns. For example, the dataset Yeast has float features and high label cardinality, while the dataset GenBase and Medical have binary features and low label cardinality. What's more, the latter two have the problem of sparse feature representation and high dimension. So hMuLab may perform reasonably in many other multi-label biomedical learning problems. hMuLab is a general method for the multi-label learning problem, and may work reasonably on the multi-label datasets from the other areas. hMuLab has two major time-consuming steps, *i.e.* the matrix-based coefficient calculation and the neighbor searching. We plan to speed up the matrix calculation by an GPU implementation and the neighbor searching by kd-tree based vector quantization.

REFERENCES

[1] M. L. Zhang, and Z. H. Zhou, "A Review on Multi-Label Learning Algorithms," *Ieee Transactions on Knowledge and Data Engineering,* vol. 26, no. 8, pp. 1819-1837, Aug, 2014.

[2] E. Gibaja, and S. Ventura, "Multi-label learning: a review of the state of the art and ongoing research," *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery,* vol. 4, no. 6, pp. 411-444, Nov-Dec, 2014.

[3] M. Zhou, Y. Luo, G. Sun, G. Mai, and F. Zhou, "Constraint programming based biomarker optimization," *Biomed Res Int,* vol. 2015, pp. 910515, 2015.

[4] P. Guo, Y. Luo, G. Mai, M. Zhang, G. Wang, M. Zhao, L. Gao, F. Li, and F. Zhou, "Gene expression profile based classification models of psoriasis," *Genomics,* vol. 103, no. 1, pp. 48-55, Jan, 2014.

[5] D. J. Yu, Y. Li, J. Hu, X. Yang, J. Y. Yang, and H. B. Shen, "Disulfide Connectivity Prediction Based on Modelled Protein 3D Structural Information and Random Forest Regression," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 12, no. 3, pp. 611-21, May-Jun, 2015.

[6] B. Liao, Y. Jiang, W. Liang, W. Zhu, L. Cai, and Z. Cao, "Gene Selection Using Locality Sensitive Laplacian Score," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 11, no. 6, pp. 1146-56, Nov-Dec, 2014.

[7] A. Clare, and R. King, "Knowledge Discovery in Multi-label Phenotype Data," *Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science L. De Raedt and A. Siebes, eds., pp. 42-53: Springer Berlin Heidelberg, 2001.

[8] S. Wan, M. W. Mak, and S. Y. Kung, "R3P-Loc: A compact multi-label predictor using ridge regression and random projection for protein subcellular localization," *J Theor Biol,* vol. 360, pp. 34-45, Nov 7, 2014.

[9] F. Sun, J. Tang, H. Li, G. J. Qi, and T. S. Huang, "Multi-label image categorization with sparse factor representation," *IEEE Trans Image Process,* vol. 23, no. 3, pp. 1028-37, Mar, 2014.

[10] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Yu, "Protein Function Prediction using Multi-label Ensemble Classification," *IEEE/ACM Trans Comput Biol Bioinform*, Sep 13, 2013.

[11] Y. Y. Xu, F. Yang, Y. Zhang, and H. B. Shen, "An image-based multi-label human protein subcellular localization predictor (iLocator) reveals protein mislocalizations in cancer tissues," *Bioinformatics,* vol. 29, no. 16, pp. 2032-40, Aug 15, 2013.

[12] H. L. Zou, "A Multi-label Classifier for Prediction Membrane Protein Functional Types in Animal," *Journal of Membrane Biology,* vol. 247, no. 11, pp. 1141-1148, Nov, 2014.

[13] S. B. Wan, M. W. Mak, and S. Y. Kung, "R3P-Loc: A compact multi-label predictor using ridge regression and random projection for protein subcellular localization," *Journal of Theoretical Biology,* vol. 360, pp. 34-45, Nov 7, 2014.

[14] X. P. Shen, M. Boutell, J. B. Luo, and C. Brown, "Multi-label machine learning and its application to semantic scene classification," *Storage and Retrieval Methods and Applications for Multimedia 2004,* vol. 5307, pp. 188-199, 2004.

[15] Z. Min-Ling, and Z. Zhi-Hua, "A Review on Multi-Label Learning Algorithms," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 26, no. 8, pp. 1819-1837, 2014.

[16] A. McAllum, "Multi-Label Text Classification with a Mixture Model Trained by {EM}."

[17] R. E. Schapire, and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning,* vol. 39, no. 2-3, pp. 135-168, May, 2000.

[18] L. Zhang, Y. Zhao, and Z. F. Zhu, "Extracting shared subspace incrementally for multi-label image classification," *Visual Computer,* vol. 30, no. 12, pp. 1359-1371, Dec, 2014.

[19] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *Ieee Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 3, pp. 394-410, Mar, 2007.

[20] M. R. Boutell, J. B. Luo, X. P. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition,* vol. 37, no. 9, pp. 1757-1771, Sep, 2004.

[21] A. Wieczorkowska, P. Synak, and Z. Ra•, "Multi-Label Classification of Emotions in Music," *Intelligent Information Processing and Web Mining*, Advances in Soft Computing M. Kłopotek, S. Wierzchoń and K. Trojanowski, eds., pp. 307-315: Springer Berlin Heidelberg, 2006.

[22] E. A. Tanaka, S. R. Nozawa, A. A. Macedo, and J. A. Baranauskas, "A multi-label approach using binary relevance and decision trees applied to functional genomics," *Journal of Biomedical Informatics,* vol. 54, pp. 85-95, Apr, 2015.

[23] P. Vateekul, M. Kubat, and K. Sarinnapakorn, "Hierarchical multi-label classification with SVMs: A case study in gene function prediction," *Intelligent Data Analysis,* vol. 18, no. 4, pp. 717-738, 2014.

[24] X. Xiao, P. Wang, W. Z. Lin, J. H. Jia, and K. C. Chou, "iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types," *Analytical Biochemistry,* vol. 436, no. 2, pp. 168-177, May 15, 2013.

[25] H. Wang, H. Huang, and C. Ding, "Function-Function Correlated Multi-label Protein Function Prediction over Interaction Networks," *Journal of Computational Biology,* vol. 20, no. 4, pp. 322-343, Apr, 2013.

[26] Y. Xu, L. P. Jiao, S. Y. Wang, J. S. Wei, Y. B. Fan, M. D. Lai, and E. I. C. Chang, "Multi-Label Classification for Colon Cancer Using Histopathological Images," *Microscopy Research and Technique,* vol. 76, no. 12, pp. 1266-1277, Dec, 2013.

[27] H. Shao, G. Z. Li, G. P. Liu, and Y. Q. Wang, "Symptom selection for multi-label data of inquiry diagnosis in traditional

Chinese medicine," *Science China-Information Sciences,* vol. 56, no. 5, May, 2013.

[28] R. W. Zhao, G. Z. Li, J. M. Liu, and X. Wang, "Clinical Multi-label Free Text Classification by Exploiting Disease Label Relation," *2013 Ieee International Conference on Bioinformatics and Biomedicine (Bibm)*, 2013.

[29] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random k-Labelsets for Multilabel Classification," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 23, no. 7, pp. 1079-1089, 2011.

[30] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, eds., pp. 667-685: Springer US, 2010.

[31] W. W. Cheng, and E. Hullermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Machine Learning,* vol. 76, no. 2-3, pp. 211-225, Sep, 2009.

[32] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-Labelsets for Multilabel Classification," *Ieee Transactions on Knowledge and Data Engineering,* vol. 23, no. 7, pp. 1079-1089, Jul, 2011.

[33] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning,* vol. 85, no. 3, pp. 333-359, Dec, 2011.

[34] M. L. Zhang, and L. Wu, "LIFT: Multi-Label Learning with Label-Specific Features," *Ieee Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, no. 1, pp. 107-120, Jan, 2015.

[35] J. J. Zhang, M. Fang, and X. Li, "Multi-label learning with discriminative features for each label," *Neurocomputing,* vol. 154, pp. 305-316, Apr 22, 2015.

[36] R. E. Schapire, and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning,* vol. 37, no. 3, pp. 297-336, Dec, 1999.

[37] M. L. Zhang, and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label leaming," *Pattern Recognition,* vol. 40, no. 7, pp. 2038-2048, Jul, 2007.

[38] A. Elisseeff, and J. Weston, "A kernel method for multi-labelled classification," *Advances in Neural Information Processing Systems 14, Vols 1 and 2,* vol. 14, pp. 681-687, 2002.

[39] Z. Min-Ling, and Z. Zhi-Hua, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 18, no. 10, pp. 1338-1351, 2006.

[40] T. M. Cover, and P. E. Hart, "Nearest Neighbor Pattern Classification," *Ieee Transactions on Information Theory,* vol. 13, no. 1, pp. 21-+, 1967.

[41] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. SMC-6, no. 4, pp. 325-327, 1976.

[42] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms," *Advances in Informatics, Proceedings,* vol. 3746, pp. 448-456, 2005.

[43] J. P. Pestian, C. Brew, Pawe, #322, Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, #322, odzis, #322, and a. Duch, "A shared task involving multi-label classification of clinical free text," in Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, 2007, pp. 97-104.

[44] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "MULAN: A Java Library for Multi-Label Learning," *Journal of Machine Learning Research,* vol. 12, pp. 2411-2414, Jul, 2011.

Pu Wang received the bachelor and master degrees from the Jingdezhen Ceramic Institute, China, in 2006 and 2009, respectively. Now he is a lecturer in Jingdezhen Ceramic Institute and also a PhD student in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His current research interests include Machine Learning and Bioinformatics.

Ruiquan Ge received the MS degree in computer science from Nanjing University of Posts and Telecommunications in 2009. He is currently working toward the PhD degree in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences; Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences. His research interests include bioinformatics and data mining.

Xuan Xiao received his PhD from Donghua University, China, in 2006. Currently, he is working as full professor in Jingdezhen Ceramic Institute, China. His research includes Bioinformatics, Semiotics and Pattern recognition.

Manli Zhou received the BS and MS degrees in computer science from Northeast Normal University, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, in 2012 and 2015, respectively. Her research interests include bioinformatics and data mining.

Fengfeng Zhou (M'2012-SM'2013) received the BS and PhD degrees of computer science from the University of Science and Technology of China in 2000 and 2005, respectively. He was awarded the *Hundred Talent* program from the Chinese Academy of Sciences, and the *Tang Aoqing* professorship from the Jilin University.

He is a full professor of health informatics with the College of Computer Science and Technology, Jilin University, Changchun, P.R. China. His laboratory focuses on the data fusion and multivariate biomarker selection algorithms for the heterogeneous health big data, including bio-OMICs, biomedical imaging, physiological signal, biochemical screening and electronic health record, etc.