August 13-17, 2017
Halifax, NS, Canada

Association for
Computing Machinery

Advancing Computing as a Science & Profession

KDD'17

Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

Sponsored by:

ACM SIGKDD & ACM SIGMOD

Supported by:

DiDi, Amazon, Microsoft, Alibaba Group, Facebook, SAS, CapitalOne, Criteo Research, Google, E-AI, Siemens, LinkedIn, Huawei, Booking.com, Honeywell, American Express, Oracle, Western Digital, Open Innovation KNIME, Institut Tutte, Uber, Two Sigma, Cambridge University Press, MIT Press, Springer, & Syngenta

Functional Annotation of Human Protein Coding Isoforms via Non-convex Multi-Instance Learning

Tingjin Luo
tingjinluo@gmail.com
College of Science& Department of Computational Medicine and Bioinformatics, National University of Defense Technology
Changsha, Hunan 410073, China

Weizhong Zhang
zhangweizhongzju@gmail.com
College of Computer Science & Department of Computational Medicine and Bioinformatics & Tencent AI Lab
Zhejiang University
Hangzhou, Zhejiang 310058, China

Shang Qiu
qiush@umich.edu
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109, USA

Yang Yang
yangyangfuture@gmail.com
State Key Lab of Software Development Environment, School of Computer Science and Engineering
Beihang University
Beijing 100191, China

Dongyun Yi
dongyun.yi@gmail.com
College of Science & State Key Laboratory of High Performance Computing, National University of Defense Technology
Changsha, Hunan 410073, China

Guangtao Wang
gtwang@umich.edu
Department of Computational Medicine and Bioinformatics
University of Michigan
Ann Arbor, MI 48109, USA

Jieping Ye
jpye@umich.edu
Department of Electrical Engineering and Computer Science & Department of Computational Medicine and Bioinformatics
University of Michigan
Ann Arbor, MI 48109, USA

Jie Wang
jwangumi@umich.edu
Department of Computational Medicine and Bioinformatics
University of Michigan
Ann Arbor, MI 48109, USA

ABSTRACT

Functional annotation of human genes is fundamentally important for understanding the molecular basis of various genetic diseases. A major challenge in determining the functions of human genes lies in the functional diversity of proteins, that is, a gene can perform different functions as it may consist of multiple protein coding isoforms (PCIs). Therefore, differentiating functions of PCIs can significantly deepen our understanding of the functions of genes. However, due to the lack of isoform-level gold-standards (ground-truth annotation), many existing functional annotation approaches are developed at gene-level. In this paper, we propose a novel approach to differentiate the functions of PCIs by integrating sparse simplex projection—that is, a nonconvex sparsity-inducing regularizer—with the framework of multi-instance learning (MIL). Specifically, we label the genes that are annotated to the function under consideration as *positive bags* and the genes without the function as *negative bags*. Then, by sparse projections onto simplex, we learn a mapping that embeds the original bag space to a discriminative feature space. Our framework is flexible to incorporate various smooth and non-smooth loss functions such as logistic loss and hinge loss. To solve the resulting highly nontrivial non-convex and non-smooth optimization problem, we further develop an efficient block coordinate descent algorithm. Extensive experiments on human genome data demonstrate that the proposed approaches significantly outperform the state-of-the-art methods in terms of functional annotation accuracy of human PCIs and efficiency.

CCS CONCEPTS

•Information systems →Data mining; •Computing methodologies →Optimization algorithms; Instance-based learning; *Information extraction; Semi-supervised learning settings; Bagging;* **•Applied computing →Computational genomics; Bioinformatics;**

KEYWORDS

Non-Convex Problem; Key Instance Detection; Human PCIs; Multiple Instance Learning; Alternative Splicing

This work is partly supported by research grants from NSF China (No. 61473302, 61503396), NIH (R01 LM010730, U54 EB020403) and NSF (IIS- 0953662, III-1539991, III-1539722). Jieping Ye and Jie Wang are both corresponding authors.

KDD '17, August 13–17, 2017, Halifax, NS, Canada

978-1-4503-4887-4/17/08…$15.00
DOI: 10.1145/3097983.3097984

1 INTRODUCTION

Functional annotation of human protein coding isoforms (PCIs) is a central task in bioinformatics and plays a critical role in understanding the biological significance and underlying mechanisms of genes. Recent studies [6, 25] have shown that a gene can perform various functions as it may consist of multiple PCIs. According to the most recent GENCODE human annotation (version 19) [9, 11, 12, 19], a total of 57,820 genes consist of 196,520 PCIs. Moreover, PCIs can not only increase the protein functional diversity of mammalian genomes, but also is closely related to various human inherited diseases [9, 34, 40], such as colorectal cancer and spinal muscular atrophy. Therefore, differentiating the functions of PCIs [6, 15, 37] will accelerate our understanding of protein and gene functions.

However, due to the lack of isoform-level gold-standards (ground-truth), many existing functional annotation methods are developed at gene-level based on typical supervised learning algorithms, such as support vector machine (SVM) [20], logistic regression [21], Bayesian network [17] and Adaboost [32]. A major challenge in determining the functions of genes is the functional diversity of proteins, that is, a gene can perform different functions as it may consist of multiple PCIs. Without the ground-truth of isoforms, it is very difficult to build a suitable classification model and annotate the functions of PCIs by these supervised methods directly. Nevertheless, with development of recent bio-technologies, a large amount of gene expression data is obtained by deep sequencing of RNA and provides informative source for identifying the functions of PCIs. Thus the wide availability of RNA-seq data greatly increases our ability to differentiate the functions of isoforms. Furthermore, a suitable machine learning model can greatly improve the functional prediction performance of PCIs.

The main challenge for functional annotation task is how to use gene-level label information and a large number of gene expression features to predict isoform-level patterns. Lack of isoform-level gold standards prevents the functional annotation at isoform-level. Recently, the multiple instance learning (MIL) [15, 26, 31, 33] approaches have been adopted to tackle this kind of problem. R. Eksi et al. [15] developed a method named multiple instance support vector machine (miSVM) to differentiate functions of alternatively spliced isoforms on the mouse RNA-seq data. Recently, Panwar et al. [31] adopted miSVM to annotate the funtions of human isoforms. However, the optimal solution of miSVM was sensitive to the initial labels of isoforms inherited from positive genes and a threshold that represents a degree of strictness for assigning labels.

Recent studies [2, 7, 15] indicate that functional annotation of PCIs can be viewed as a new scenario of MIL. Although the rapid growth of RNA-seq data opens the door for functional annotation of human PCIs, the major challenges for differentiating functions of human PCIs still remain:

- Capturing the differential functions of human PCIs directly by existing experimental approaches is difficult.
- The annotated isoforms are unavailable. Annotations of different functions are commonly performed at the gene level in widely used databases such as Gene Ontology [4, 9] and KEGG [6, 22, 23] instead of isoform level. In addition, as each gene may contain more than one isoform, traditional supervised learning algorithms are inadequate to determine if a isoform is related to a specific function assigned at the gene level.
- The functional annotation task of human PCIs is unconventional. In fact, this task consists of two types of predictions. When given a positive gene with a function, the first task aims to determine which of its isoforms inherit this function. Another prediction task is to assign the functions of isoforms even for genes which are unknown to these functions and simultaneously predict the functions of genes. Nevertheless, few of MIL methods are suitable for these two predictions simultaneously.

To address these challenges, we propose a novel approach to differentiate PCIs' functions by integrating sparse simplex projection—that is, a nonconvex sparsity-inducing regularizer—with the framework of MIL. Specifically, a gene carries out a specific function by its key isoforms. To obtain a more discriminative feature representation of positive genes, we detect the key isoforms from them and introduce an isoform weight vector for each positive gene to measure the contribution of its isoforms. Based on the assumption that each positive gene consists of at least one of positive isoforms to carry out the function, we impose a nonconvex sparsity-inducing regularizer, which incorporates l_0-norm, l_1-norm and non-negative constraints on each isoform weight vector into our MIL framework. It enables our model better approximate the problem of key isoform detection. Finally, we learn these isoform weight vectors by sparse projections onto simplex and obtain the new feature representations of the positive genes. Our unified framework is flexible to incorporate various smooth and non-smooth loss functions such as logistic loss and hinge loss. Furthermore, under our framework, we propose a novel method named weighted logistic regression-based MIL method (WLRM). With the increase of gene expression features' dimensionality, elastic net regularization is employed in our methods to alleviate the over-fitting problem. To solve our formulated non-convex and non-smooth optimization problem, we further develop an efficient accelerated block coordinate decent (BCD) algorithm. Extensive experiments on human genome data show that our methods significantly outperform the state-of-the-art methods in terms of functional annotation accuracy of human PCIs and efficiency.

The rest of this paper is organized as follows. Section 2 reviews the background of MIL, and Section 3 presents our proposed MIL framework and efficient block coordinate descent algorithm with backtracking line search. Experiments on real human genome data are presented in Section 4, and the paper concludes with a summary in Section 5.

Notations: Matrices and vectors are written as boldface uppercase letters and italic boldface lowercase letters, respectively. For a matrix $\mathbf{M} = [m_{ij}]$, its ith row and jth column are denoted by $\boldsymbol{m}^i$ and $\boldsymbol{m}_j$, respectively. The l_p-norm of a vector $\boldsymbol{v} \in \mathbb{R}^n$ is defined as $\|\boldsymbol{v}\|_p = \left(\sum_i |v_i|^p\right)^{1/p}, p > 0$ and l_0-norm is the cardinality of nonzero elements in $\boldsymbol{v}$. The notations used in this paper are summarized in Table 1.

Table 1: Notations.

Notations	Description
d	Dimensionality of the original data
N	Number of genes
N_1	Number of positive genes
B_N	Index set of all negative bags or genes
B_P	Index set of all positive bags or genes
n_i	Number of instances in ith bag or gene
$\mathbf{1}(\cdot)$	Indicator function onto set C
$\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$	Feature matrix of ith gene
$\boldsymbol{x}_j^i \in \mathbb{R}^d$	The jth feature vector of ith gene
$\boldsymbol{u}_i \in \mathbb{R}^{n_i}$	Isoform weight vector of ith gene
$\boldsymbol{w} \in \mathbb{R}^{d+1}$	Coefficients of the model

2 RELATED WORK

MIL was first introduced in [13] for drug activity prediction. Since then, many MIL methods have been proposed in the literature. Maron et.al. [29] proposed the diverse density (DD) method based on the elliptic target concept in feature space closely related to the peak density of positive instances. Zhang et al. [1] proposed a refinement of DD, named expectation maximization diverse density (EMDD) to learn the witness instances and perform multiple instance regression simultaneously by the EM method. Under the standard MI assumption, MIL could be viewed as a semi-supervised learning problem with the additional constraint that positive bags must contain at least one positive instance. In order to deal with large scale MIL problems, Wei et al. [38, 39] proposed MIL based on the Fisher Vector representation (miFV) and MIL based on the vector of locally aggregated descriptors representation (miVLAD) to convert the bag representation of an object to a simpler one, i.e., a vector representation. Thus, miFV and miVLAD only concern the classification of bags and cannot differentiate the function of instance. Besides, Andrews et al. [3] proposed multiple instance support vector machines (miSVM and MISVM) by encoding the positive constraints in the objective function of SVM. The aim of miSVM is to maximize the pattern margin of instances, while MISVM aims to maximize the margins at bag-level. It is possible for miSVM to predict the functions of isoforms based on the idea of selecting the witness instances. Recently, R. Eski et al. [15] applied miSVM and MISVM to annotate functions of mouse isoforms. Their experiments showed that miSVM performs better than MISVM. However, miSVM is sensitive to the initial labels of these isoforms extracted from positive genes.

The objective of these MIL methods is to predict the labels of bags, but not for instances. For a specific function, we have a set of positive genes annotated based on Gene Ontology (GO) and another set of negative genes that are unrelated to this function. Each gene consists of multiple isoforms. Differentiating functions of the genes can be tackled by traditional MIL methods. Nevertheless, the target of functional annotation of human PCIs consists of three tasks: key isoform detection, functional prediction of genes and PCIs. Due to the lack of the ground-truth of isoforms, it is very difficult to determine the functions of isoforms at given a multiple instance setting. In other words, functional annotation of human PCIs is very different from traditional MIL problems [33] and can be viewed as a new type of MIL. Therefore, it is necessary to develop a novel approach that can select key isoforms and differentiate the functions of PCIs simultaneously.

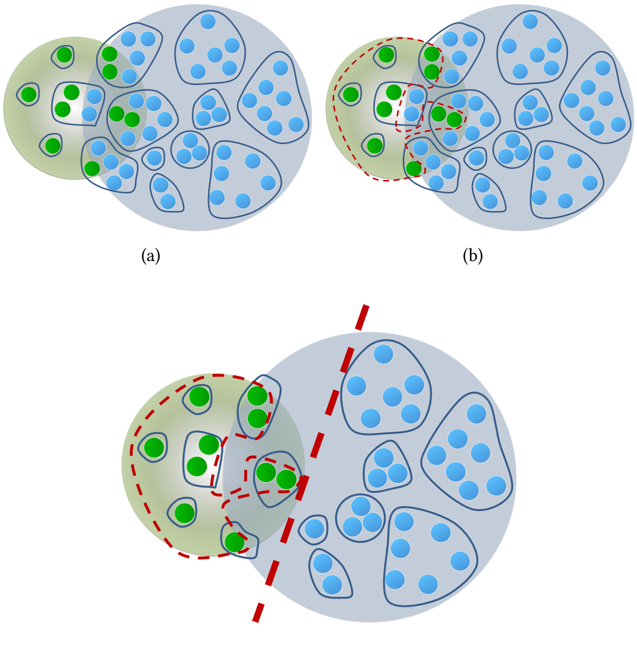


Figure 1: Illustration of key isoforms of positive genes for a certain function. (a) The green circles are the true positive isoforms and the remaining blue circles are negative isoforms. Each blue ring including at least one green circle represents the positive gene and others are negative genes. (b) The target of key isoform detection is to find all of the green circles in positive genes. (c) The classification hyperplane is learned by selected key isoforms.

3 A NOVEL MIL FRAMEWORK VIA NON-CONVEX PROGRAMING

We are given a set of genes $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n\}$, and their corresponding labels $\boldsymbol{y} = \{y_1, y_2, ..., y_n\} \in \{-1, +1\}^n$ for a specific biological process. The ith gene includes n_i isoforms whose feature vectors are $\mathbf{X}_i = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_i}] \in \mathbb{R}^{d \times n_i}$. In our paper, a bag refers to a gene, which contains multiple PCIs. An instance refers to an individual isoform and a positive bag refers to a positive gene related to the specific function. We aim to detect key isoforms from positive genes and predict the functions of isoforms jointly by using the available gene label information.

3.1 Motivation and Formulation

Most existing MIL methods [8, 15, 26, 27, 29] assume that each instance in a bag plays an equal role when considering the similarity between two bags. However, for a specific biological process, only a few of positive genes carry out this function. For each positive gene, only its key isoforms are closely related to this function. In other words, the importance of isoforms for genes is not equal, especially for positive genes. As shown in Fig.1, only a few of key isoforms carry out this function and they are critical to functional annotation of genes. Meanwhile, as demonstrated in [3] and [28], key isoform detection is able to discriminate the functions of isoforms and thus help to improve the performance in real applications. This motivates us to develop a novel approach that is able to differentiate the functions of PCIs by integrating a nonconvex sparsity-inducing regularizer within the framework of MIL.

Without loss of generality, denote y and $\hat{y}$ as the true label and the predicted label for data point $\boldsymbol{x}$. Then the loss function is defined as $l(y, \hat{y})$. Similar to the typical supervised learning methods, we adopt the linear model to predict the functions of isoforms, that is, $\hat{y} = \boldsymbol{w}^T \boldsymbol{x} + b$. However, although the labels of all genes are known, the ground-truth of each isoform in positive genes remains unavailable, which renders the loss $l(y, \hat{y})$ difficult to compute. For the i-th positive gene, a positive gene carries out a specific function by its key isoforms. Thus we introduce an isoform weight vector $\boldsymbol{u}_i \in \mathbb{R}^{n_i}$ to measure the contribution of n_i isoforms to the function of this gene. If an isoform is negative, its weight will be zero. Note that the isoform weight vector does not only detect the key isoforms, but also eliminate the effect of negative isoforms in positive genes. Thus the isoform weight vector can enhance the discriminative power of our model. With the estimated isoform weight vector, we represent the positive gene by its selected key isoforms, that is, the new feature representation of the i-th positive gene is $\mathrm{X}_i \boldsymbol{u}_i$. The loss of the i-th positive gene is $l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i + b)$. For isoforms of negative genes, their labels can inherit from the genes directly, and the loss of i-th negative gene is $\sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i + b)$. Finally, the loss of our model is formulated as

$$\sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i + b) + \sum_{i \in B_N} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i + b), \tag{1}$$

where $\boldsymbol{w}$ is the coefficients of the model, and B_P and B_N are the index vectors of positive genes and negative genes, respectively. For simplicity, the bias b can be absorbed into $\boldsymbol{w}$ when the constant value 1 is added as an additional dimension for each isoform $\boldsymbol{x}_i$. Thus the problem in Eq. (1) is rewritten as

$$\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i) + \sum_{i \in B_N} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i). \tag{2}$$

In practice, the RNA-seq data set is typically imbalanced, since the number of negative genes is much more than the number of positive genes. Similar to [20], we employ a weight parameter ρ/n_i for each negative gene to alleviate the imbalanced problem. The problem (2) is reformulated as

$$\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i) + \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i), \tag{3}$$

Based on the assumption [15] that each positive gene contains at least one key isoform to carry out the function and the remaining ones are negative isoforms, the cardinality (l_0-norm) constraint is a natural way to constrain the number of selected key isoforms. By definition, each element of the isoform weight vector represents the relationship between isoform and the function. It requires all elements of isoform weight vector to be non-negative. To some extent, the isoform weight vector can be viewed as a mapping which embeds the original bag space to a discriminative feature space. The new feature representation of a positive gene is the convex combination of all selected key isoforms. It is natural to constrain the summation of all elements to be equal to 1, that is, l_1-norm constraint. Therefore, the nonconvex sparsity-inducing regularizer, which incorporates l_0-norm, l_1-norm and non-negative constraints into the isoform weight vector, is employed in our formulation to better approximate the problem of key isoform detection. The formulation in Eq. (3) becomes

$$\begin{aligned}
&\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i) + \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i) \\
&s.t.\ \forall i \in \{1, \ldots, N_1\}, \|\boldsymbol{u}_i\|_0 \leq r, \|\boldsymbol{u}_i\|_1 = 1, \boldsymbol{u}_i \geq 0.
\end{aligned} \tag{4}$$

Similar to SVM, when the dimensionality of expression features is large than the number of isoforms, Eq. (4) is also prone to over-fitting. A standard technique to alleviate over-fitting is regularization. Zou et al. [42] proposed the elastic net penalty which is a flexible regularization by mixing l_1-norm and l_2-norm regularization. Prior works have shown that the model with the elastic net penalty often outperforms the model with l_1-norm or l_2-norm regularization only. Thus our unified model incorporates the elastic net regularization to sparsify the coefficients $\boldsymbol{w}$. Finally, our unified framework can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i) + \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i) \\
&\quad + \lambda_1 \|\boldsymbol{w}\|^2 + \lambda_2 \|\boldsymbol{w}\|_1 \\
&s.t.\ \forall i \in \{1, \ldots, N_1\}, \|\boldsymbol{u}_i\|_0 \leq r, \|\boldsymbol{u}_i\|_1 = 1, \boldsymbol{u}_i \geq 0.
\end{aligned} \tag{5}$$

Note that our unified framework is flexible to incorporate various smooth and non-smooth loss functions. In this paper, we propose the weighted logistic regression-based MIL method (WLRM) by using logistic loss under this framework. Thus the basic loss function of WLRM can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} \log\left(1 + \exp(-y_i \langle \boldsymbol{w}, \mathrm{X}_i \boldsymbol{u}_i \rangle)\right) + \\
&\qquad \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} \log\left[1 + \exp(-y_i \langle \boldsymbol{w}, \boldsymbol{x}_j^i \rangle)\right].
\end{aligned} \tag{6}$$

in which by integrating the noncovex sparsity-inducing regularizers, we can finally obtain our WLRM model as follows

$$\begin{aligned}
&\min_{\boldsymbol{w}, \boldsymbol{u}_i} \sum_{i \in B_P} \log\left(1 + \exp(-y_i \langle \boldsymbol{w}, \mathrm{X}_i \boldsymbol{u}_i \rangle)\right) + \lambda_1 \|\boldsymbol{w}\|^2 + \\
&\sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} \log\left[1 + \exp(-y_i \langle \boldsymbol{w}, \boldsymbol{x}_j^i \rangle)\right] + \lambda_2 \|\boldsymbol{w}\|_1 \\
&s.t.\ \forall i \in \{1, \ldots, N_1\}, \|\boldsymbol{u}_i\|_0 \leq r, \|\boldsymbol{u}_i\|_1 = 1, \boldsymbol{u}_i \geq 0.
\end{aligned} \tag{7}$$

It is worth noting that, although traditional logistic regression (LR) has been applied in many real applications, LR is not effective for functional annotation of PCIs, because there are no sufficient label information of isoforms, especially labels of true positive isoforms. Meanwhile, to solve our non-convex and non-smooth formulation, we further develop an efficient accelerated block coordinate decent (BCD) algorithm.

3.2 Optimization Methods

In this section, we develop an efficient algorithm based on the block coordinate descent (BCD) to solve this unified framework. Denote $\boldsymbol{x} = (\boldsymbol{w}, \boldsymbol{u}_1, ..., \boldsymbol{u}_{n_1})$ and $C_i = \{\boldsymbol{u}_i | \boldsymbol{u}_i \in \mathbb{R}^{n_i}, \|\boldsymbol{u}_i\|_1 = 1, \|\boldsymbol{u}_i\|_0 \leq r, \boldsymbol{u}_i \geq 0\}$, and the loss function is defined as

$$f(\boldsymbol{x}) = \sum_{i \in B_P} l(y_i, \boldsymbol{w}^T \mathrm{X}_i \boldsymbol{u}_i) + \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} l(y_i, \boldsymbol{w}^T \boldsymbol{x}_j^i).$$

The problem (5) can be summarized as the following framework

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{n_s} r_i(\boldsymbol{x}_i), \tag{8}$$

where $n_s = N_1 + 1$ and the function $r_1(\boldsymbol{x}_1) = \lambda_1 \|\boldsymbol{w}\|^2 + \lambda_2 \|\boldsymbol{w}\|_1$ and $r_i(\boldsymbol{x}_i) = \mathbf{1}_{C_i}(\boldsymbol{u}_i)$ for $i = 2, ..., n_s$ is an indicator function of C_i. By allowing $\mathbf{1}_{C_i}$ to take the ∞-value, $\mathbf{1}_{C_i}$ can incorporate the constraints $\boldsymbol{u}_i \in C_i$ since enforcing the constraints is equivalent to minimize the indicator function of C_i. According to its definition, $\mathbf{1}_{C_i}$ is proper and closed.

We can observe that (1) our unified framework may include the smooth and non-smooth loss function; (2) $\boldsymbol{w}$ and $\boldsymbol{u}_i$ for $i = 1, 2, ..., N_1$ are coupled in the loss function and (3) the nonconvex sparsity-inducing regularizer is imposed to constrain each $\boldsymbol{u}_i$. Thus, each block of $\boldsymbol{x}$ may be non-convex and non-smooth and the problem (8) is a highly nontrivial nonconvex and non-smooth optimization problem. It is very difficult to solve the problem directly by the general gradient descent methods. Motivated by the idea [41], we develop an efficient block coordinate descent (BCD) algorithm to solve the problem (5). Specifically, we minimize F cyclically over each block of variables $\boldsymbol{x}_i$ by BCD method of Gauss-Seidel type, while fixing the remaining blocks at their last updated values. Let $\boldsymbol{x}_i^{k+1}$ denote the value of $\boldsymbol{x}_i$ after its k-th update and let $f_i^k(\boldsymbol{x}_i) \triangleq f(\boldsymbol{x}_1^k, ..., \boldsymbol{x}_{i-1}^k, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}^k, ..., \boldsymbol{x}_{n_s}^k)$ for all i and k. At each iteration, we adopt a prox-linear surrogate function to approximate the upper bound of $F(\boldsymbol{x}^{k+1})$, and then each block of variables $\boldsymbol{x}_i$ can be updated as follows:

$$\begin{aligned} \boldsymbol{x}_i^{k+1} \in \arg\min_{\boldsymbol{x}_i} & f_i^k(\hat{\boldsymbol{x}}_i^{k+1}) + \langle \hat{g}_i^k, \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{k+1} \rangle \\ & + \frac{1}{2\alpha_k} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{k+1}\|^2 + r_i(\boldsymbol{x}_i), \end{aligned} \tag{9}$$

Since $f_i^k(\hat{\boldsymbol{x}}_i^{k+1})$ is constant with respect to $\boldsymbol{x}_i$, the problem (9) is equivalent to the following problem

$$\boldsymbol{x}_i^{k+1} \in \arg\min_{\boldsymbol{x}_i} \langle \hat{g}_i^k, \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{k+1} \rangle + \frac{1}{2\alpha_k} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{k+1}\|^2 + r_i(\boldsymbol{x}_i), \tag{10}$$

where $\alpha_k > 0$ is a step-size, $\hat{g}_i^k = \nabla f_i^k(\hat{\boldsymbol{x}}_i^k)$ and $\hat{\boldsymbol{x}}_i^{k+1}$ is the extrapolation

$$\hat{\boldsymbol{x}}_i^{k+1} = \boldsymbol{x}_i^k + \gamma_k (\boldsymbol{x}_i^k - \boldsymbol{x}_i^{k-1}),$$

where $\gamma_k \geq 0$ is an extrapolation weight. While we can simply set $\gamma_k = 0$, an appropriate $\gamma_k > 0$ can speed up the convergence of algorithm. Similar to the Nesterov's accelerated gradient descent [30], the extrapolation weight is given by $\gamma_k = \frac{t_k - 1}{t_{k+1}}$ with

$$t_0 = 0, t_k = \left(\sqrt{4t_{k-1}^2 + 1} + 1\right)/2.$$

Algorithm 1 Accelerated Block Coordinate Descent for Problem(5)

```
Initialize $\boldsymbol{w}^1 = \boldsymbol{w}^0, \boldsymbol{u}_i^1 = \boldsymbol{u}_i^0, \beta = 0.85, t_0 = 0$ and $k = 1$.
repeat
    1. Compute $t_k = \left(1 + \sqrt{4t_{k-1}^2 + 1}\right)/2$ and $\gamma_k = \frac{t_k - 1}{t_{k+1}}$;
    2. Find the smallest non-negative integer $i_k$ by backtracking line search with $\bar{\alpha}_k(\boldsymbol{w}) = \beta^{i_k} \alpha_k(\boldsymbol{w})$;
    3. Update $\boldsymbol{w}^{k+1}$ by $\boldsymbol{w}^{k+1} = \hat{\boldsymbol{w}}^{k+1} - \alpha_k \nabla q(\hat{\boldsymbol{w}}^{k+1})$.
    for $i = 1, 2, ..., N_1$ do
        4. Find the smallest non-negative integer $i_k$ by backtracking line search with $\bar{\alpha}_k(\boldsymbol{u}_i) = \beta^{i_k} \alpha_k(\boldsymbol{u}_i)$;
        5. Update each $\boldsymbol{u}_i^{k+1}$ by $\boldsymbol{u}_i^{k+1}(\mathcal{S}^*) = P_{C_i}(\tilde{\boldsymbol{u}}_i^{k+1}(\mathcal{S}^*))$.
    end for
    6. $k = k + 1$.
until Stopping criterion is satisfied.
```

When the Lipschitz constant L_k of $\hat{g}_i^k$ about $\boldsymbol{x}_i$ is known, we can set the step-size $\alpha_k = \frac{\beta}{L_k}$ with any $0 < \beta \leq 1$. However, L_k is often unknown or difficult to bound in practice, we will choose a proper step size α_k by backtracking line search method under the criterion:

$$f(\boldsymbol{x}^k) \leq f(\hat{\boldsymbol{x}}^{k-1}) + \langle \nabla f_i^k(\hat{\boldsymbol{x}}^{k-1}), \boldsymbol{x}_i^k - \hat{\boldsymbol{x}}_i^{k-1} \rangle + \frac{\beta}{2\alpha_k} \|\boldsymbol{x}_i^k - \hat{\boldsymbol{x}}_i^{k-1}\|^2.$$

Finally, the optimization details of solving the problem (5) by accelerated block coordinate descent under backtracking line search are listed in Algorithm 1.

3.3 Optimize model coefficients $\boldsymbol{w}$

When $\boldsymbol{u}_i$ is fixed, the problem (8) can be reformulated as:

$$q(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda_1 \|\boldsymbol{w}\|^2 + \lambda_2 \|\boldsymbol{w}\|_1. \tag{11}$$

Motivated by [5, 42], when fixed all of positive isoform weight vectors $[\boldsymbol{u}_1, ..., \boldsymbol{u}_{n_1}]$, the problem in (11) is a convex minimization problem with elastic net penalty. Therefore, we can obtain the optimal solution of the problem (11) by gradient-based methods. Denote $s(t) = \frac{1}{2}(1 + sign(1 - t))$ and let

$$\begin{aligned} \nabla q_{lr}(\boldsymbol{w}) = & \sum_{i \in B_N} \frac{\rho}{n_i} \sum_{j=1}^{n_i} \frac{-y_i \boldsymbol{x}_j^i}{1 + \exp(y_i \langle \boldsymbol{w}, \boldsymbol{x}_j^i \rangle)} + 2\lambda_1 \boldsymbol{w} \\ & + \sum_{i \in B_P} \frac{-y_i \mathrm{X}_i \boldsymbol{u}_i}{1 + \exp(y_i \langle \boldsymbol{w}, \mathrm{X}_i \boldsymbol{u}_i \rangle)} + \lambda_2 sign(\boldsymbol{w}), \end{aligned} \tag{12}$$

be the gradient of WLRM w.r.t. $\boldsymbol{w}$. According to Eq. (9), we derive a quadratic model to update $\boldsymbol{w}$ at each iteration:

$$\boldsymbol{w}^{k+1} \in \arg\min_{\boldsymbol{w}} \left\langle \nabla q_{lr}(\hat{\boldsymbol{w}}^{k+1}), \boldsymbol{w} - \hat{\boldsymbol{w}}^{k+1} \right\rangle + \frac{\|\boldsymbol{w} - \hat{\boldsymbol{w}}^{k+1}\|^2}{2\alpha_k}, \tag{13}$$

where $\hat{\boldsymbol{w}}^{k+1}$ is

$$\hat{\boldsymbol{w}}^{k+1} = \boldsymbol{w}^k + \gamma_k (\boldsymbol{w}^k - \boldsymbol{w}^{k-1}).$$

Moreover, the problem (13) has the closed form solution:

$$\boldsymbol{w}^{k+1} = \hat{\boldsymbol{w}}^{k+1} - \alpha_k \nabla q(\hat{\boldsymbol{w}}^{k+1}). \tag{14}$$

3.4 Optimize isoform weight vector $\boldsymbol{u}$

When $\boldsymbol{w}$ is fixed, the problem (8) becomes

$$\min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{N_1}} \sum_{i=1}^{N_1} g_i(\boldsymbol{u}_i) + \mathbf{1}_{C_i}(\boldsymbol{u}_i), \tag{15}$$

where $g_i(\boldsymbol{u}_i) = \log(1+\exp(y_i\langle \boldsymbol{w}, \mathrm{X}_i \boldsymbol{u}_i\rangle))$ for WLRM. The problem (15) is decoupled between different i, so we can solve the following problem for each i individually:

$$\min_{\boldsymbol{u}_i} g_i(\boldsymbol{u}_i) + \mathbf{1}_{C_i}(\boldsymbol{u}_i).$$

Denote

$$\nabla g_i^{lr}(\boldsymbol{u}_i) = \frac{\rho}{n_i}\frac{-y_i \mathrm{X}_i^T \boldsymbol{w}}{1+\exp(y_i\langle \boldsymbol{w}, \mathrm{X}_i \boldsymbol{u}_i\rangle)}, \tag{16}$$

as the partial derivatives of $g_i(\boldsymbol{u}_i)$ of WLRM with respect to $\boldsymbol{u}_i$. According to Eq. (9), we derive the following approximation model to update $\boldsymbol{u}_i$ at each iteration.

$$\begin{aligned} \boldsymbol{u}_i^{k+1} \in \arg\min_{\boldsymbol{u}_i} \frac{1}{2\alpha_k}\|\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i^{k+1}\|^2 + \mathbf{1}_{C_i}(\boldsymbol{u}_i) \\ + \langle \nabla g_i^{lr}(\hat{\boldsymbol{u}}_i^{k+1}), \boldsymbol{u}_i - \hat{\boldsymbol{u}}_i^{k+1}\rangle, \end{aligned} \tag{17}$$

where $\hat{\boldsymbol{u}}_i^{k+1} = \boldsymbol{u}_i^k + \gamma_k(\boldsymbol{u}_i^k - \boldsymbol{u}_i^{k-1})$.

The formulation in (17) is a convex optimization problem with a nonconvex sparsity-inducing regularizer. Motivated by the idea in [14, 24], the problem (17) can be solved by the projected gradient descent algorithm. Finally, according to the greedy selector and simplex projector (GSSP) algorithm, the optimal solution of the problem (17) is the sparse Euclidean projections of $\tilde{\boldsymbol{u}}_i^{k+1}$ onto the positive simplex C_i, that is,

$$\boldsymbol{u}_i^{k+1}(\mathcal{S}^*) = P_{C_i}(\tilde{\boldsymbol{u}}_i^{k+1}(\mathcal{S}^*)), \boldsymbol{u}_i^{k+1}(\overline{\mathcal{S}^*}) = 0, \tag{18}$$

where $P_{C_i}(\cdot)$ is a projection operator with respect to a collection of simpler sets C_i, $\tilde{\boldsymbol{u}}_i^{k+1} = \hat{\boldsymbol{u}}_i^{k+1} - \alpha_k \nabla g_i(\hat{\boldsymbol{u}}_i^{k+1})$, $\mathcal{S}^*$ is the index set which keeps the r-largest positive entries of $\tilde{\boldsymbol{u}}_i^{k+1}$ and $\overline{\mathcal{S}^*}$ is the complement of the set $\mathcal{S}^*$.

3.5 Key Instance Detection on Synthetic Data

We provide a toy example to illustrate the effectiveness of detecting key instances from positive bags in this section. The toy data set consists of two classes of instances, which are generated from two different Gaussian distributions (as shown in Fig. 2(a)). We randomly generate five positive bags and ten negative bags from two Gaussians. Fig. (2) (b) shows that, for each positive bag, we randomly sample a positive instance (the red point) from original positive class and certain negative instances from negtive class. Then we apply WLRM to detect the positive instances and learn the optimal classification model. In this experiment, we set $r = 1$. The results are displayed in Fig. (2)(c). The results verify that our WLRM is effective to detect the positive instances from bags and able to learn the optimal classification model by selected instances.

4 EXPERIMENTS

In this section, compared with the state-of-the-art methods miSVM, miFV and miVLAD, we present the experimental results on human RNA-seq data to demonstrate the effectiveness of our proposed WLRM. Meanwhile, we analyze the performance of WLRM in aspects of the parameter determination, convergence behavior and time complexity.

4.1 Experiment Settings

The dataset we used in our experiments is generated from a total of 573 human RNA-seq runs of ENCODE project [9]. We perform quality control on the original data.

(1) We used the human genome (build GRCh37.75) from Ensembl to align the short-reads of each RNA-seq dataset by TopHat (v.2.0.11) [35]. We removed the samples with less than 50% mapping reads coverage and 248 runs (of total 127 samples) remains. We then averaged expression values for each sample separately.

(2) We calculated the relative abundance of the transcript as Fragment Per Kilobase of exon per Million fragments (FPKM) by Cufflinks [36]. Then, we computed average expression values of of a total of 63,783 genes with 214,292 isoforms for each sample separately.

(3) As the extracted FPKM values for short transcripts (eg. tRNAs) were very high, we removed genes in which the average length of isoforms was less than 100 nucleotides. To ensure sufficient non-zero values for the subsequent machine learning step, we used only genes where more than half samples have larger than 1 FPKM values. Then, we used only genes marked as protein_coding (known, novel and putative) biotype in the Ensembl.

After these data preprocessing steps, we obtain a data set consisting of 11,946 genes with 59297 isoforms. We perform a $\log_2$-transformation of the FPKM values.

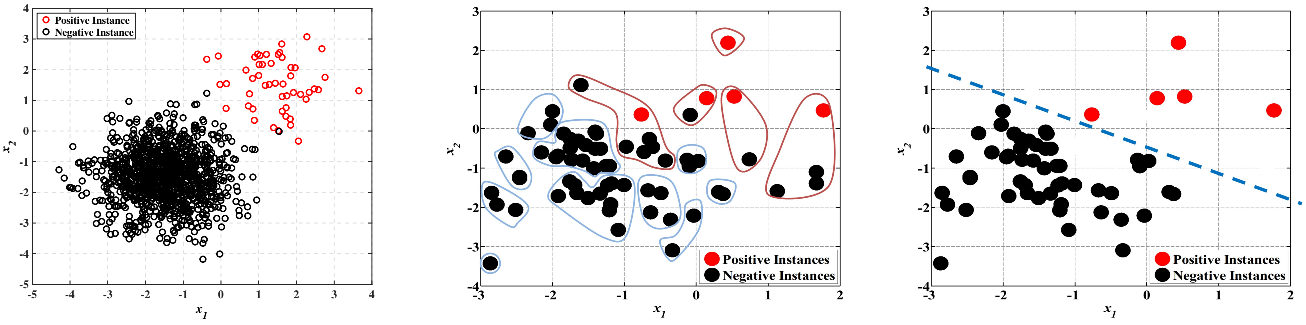
GO term based functional annotation [6, 9, 10, 36] has been well-studied in the past few years. Briefly speaking, a GO term (data-version:releases/2014-05-27) corresponds to a certain biological process and a set of "responsible" genes. Thus, for a given GO term, we label the genes that are annotated to this GO term as positive and the remaining genes as negative. The number of positive genes for these GO terms can be different, ranging from 20 to 300. In this paper, we focus on a list of 94 benchmark GO terms [1]. We follow the settings in [15] to divided all GO terms into five different groups according to the number of associated genes: (A) 20-27, (B) 28-40, (C) 41-64, (D) 65-114 and (E) 115-300.

For each GO term, we employ five fold cross validation [18] to estimate the model parameters and evaluate the performance of the proposed WLRM against miSVM[3, 31], miFV[38] and miVLAD[39] in terms of commonly used measures: classification accuracy (ACC), area under the ROC curve (AUC) [16], sensitivity, and specificity.

4.2 Functional Annotation of Human PCIs

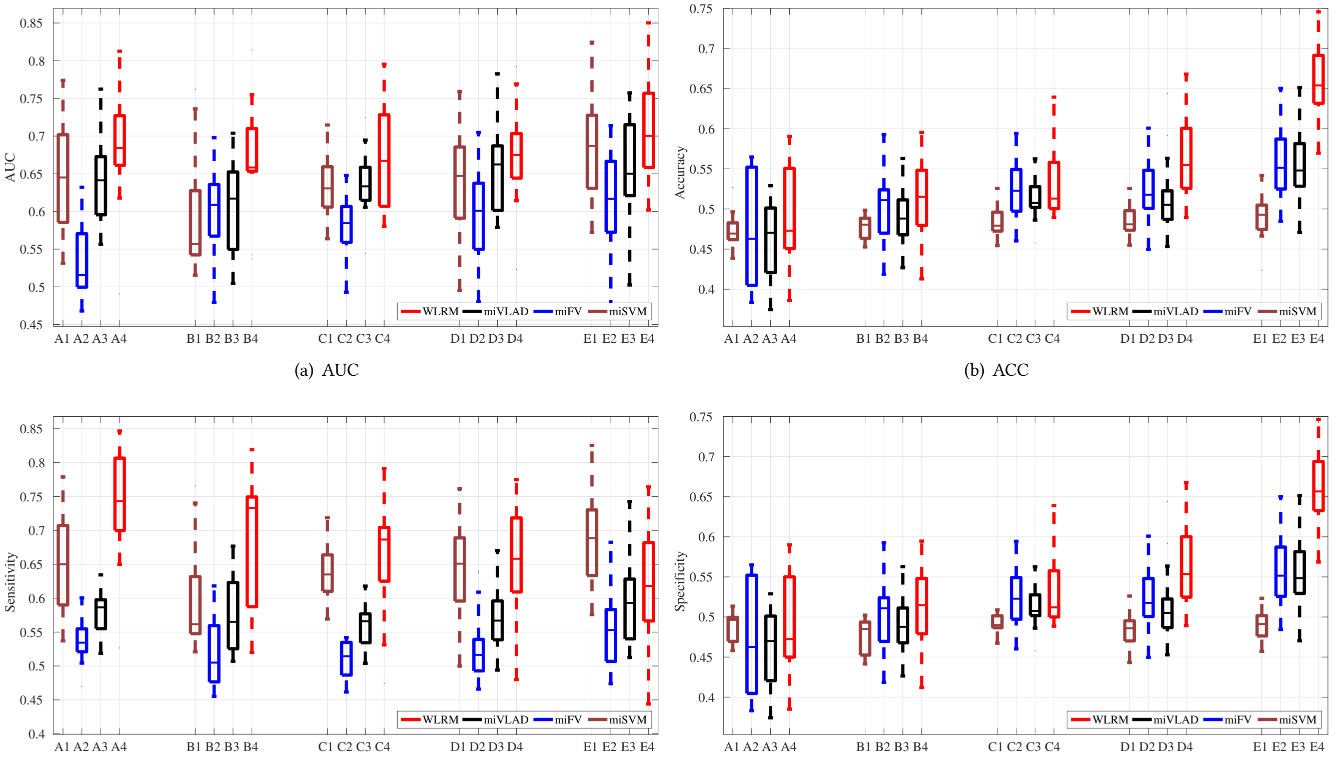
In this experiment, we apply WLRM into human RNA-seq dataset to annotate the functions of PCIs. We compare our proposed WLRM with the state-of-art algorithm miSVM, miFV and miVLAD in this experiment. Due to the lack of ground-truth of annotated isoforms, we can only evaluate the performance of all compared methods at gene level by cross validation. Thus we calculate the prediction results by the label information of genes from Gene Ontology. Recall that, we employ 5-fold cross validation to estimate the optimal parameter values for all compared methods. We choose $r = 2$

[1] http://geneontology.org/page/go-slim-and-subset-guide



(a) (b) (c)

Figure 2: Example results of the proposed method on a two dimensional synthetic data. (a) The original distribution of the synthetic positive and negative instances. (b) Training Bags with true instance-level labels sampled from original distributed data. (c) The learned classification model and the selected instances by our method.



(a) AUC

(b) ACC

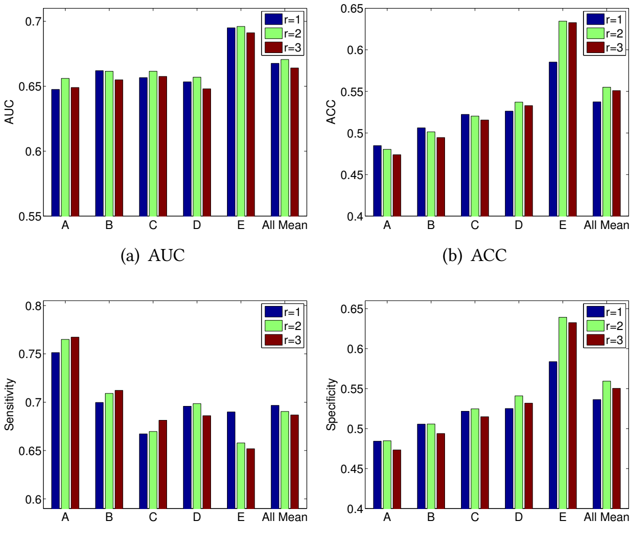
(c) Sensitivity

(d) Specificity

Figure 3: Performance of WLRM, miVLAD, miFV and miSVM for functional annotation of human PCIs. From (a) to (d) present the results of AUC, ACC, Sensitivity and Specificity, respectively.

for WLRM. The prediction performance is calculated for all of 94 GO terms. The experimental results of AUC, ACC, sensitivity and specificity are shown in Fig. 3.

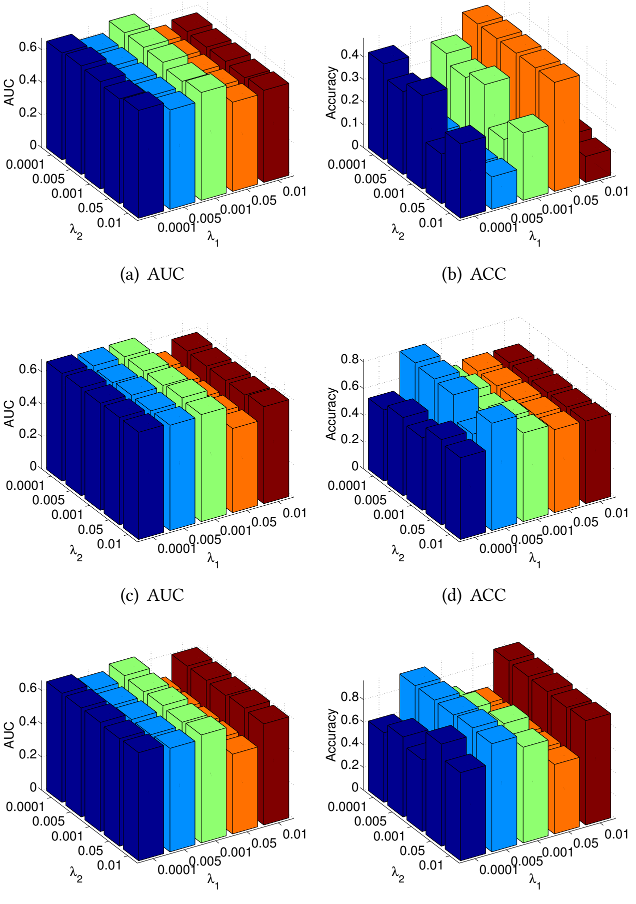
The results in Fig. 3 show that the AUC, ACC, sensitivity and specificity results of WLRM are better than those of miSVM, miFV and miVLAD. Specifically, Fig. 3 (a) and (c) indicate that when the number of GO terms is very small, the performance of our methods is much better than the results of other methods, as shown in groups A and B. In group A, the median AUC values of WLRM is 0.691, which are better than 0.645 of miSVM, 0.518 of miFV and 0.635 of miVLAD. The median sensitivity values of WLRM is 0.745, which are much higher than 0.65 of miSVM, 0.533 of miFV and 0.589 of miVLAD. The main reason is that we use the weight $\frac{\rho}{n_i}$ to solve the unbalanced problem and enhance the performance of WLERM. On the other hand, as shown in Fig. 3 (b) and (d), the performance of ACC and specificity is slightly better with an increasing GO



(a) AUC (b) ACC

(c) Sensitivity (d) Specificity

Figure 4: Performance variation of WLRM with respect to different values of the parameter *r*.



(a) AUC (b) ACC

(c) AUC (d) ACC

(e) AUC (f) ACC

Figure 5: Performance variation of WLRM with different values of ρ, λ_1 and λ_2. The top line is the mean AUC and ACC values of WLRM with $\rho = 1$. The middle line is the mean AUC and ACC values of WLRM with $\rho = 2$. The bottom line is the results of WLRM with $\rho = 3$.

term size. Because we obtain more information about positive genes with an increasing GO term size. Because WLRM is able to eliminate the effect of negative instances in positive bags by jointly selecting the positive instances and learning classification hyperplane. Overall, the performance of our proposed WLRM outperform other compared methods in most cases.

4.3 Effects of Parameters

In this experiment, we study the effects of different parameter values in terms of the classification performance. Recall that, our models have four parameters, i.e., r, ρ, λ_1 and λ_2. They can be separated two groups: first group is the parameter r, which control the number of selected positive instances; another group is parameters ρ, λ_1 and λ_2, which control the complexity of our WLRM.

We first study the effect of different values of r. The parameters r controls the cardinality of nonzero elements in each $\boldsymbol{u}_i$. The values of r is chosen from $\{1, 2, 3\}$. For other parameters, we adopt cross validation method to choose optimal values. The results of average AUC, ACC, sensitivity and specificity are shown in Fig. 4. We can observe that WLRM performs the best with $r = 2$. When $r = 3$, the performance of WLRM decreases. With the increse of r, the possibility of negative instance in positive bags is much more easily selected as positive instance.

We next show the effects of ρ, λ_1, and λ_2 for fixed $r = 2$. Fig. 5 shows the mean AUC and ACC results of WLRM with different values of ρ, λ_1 and λ_2. We can see from Fig. 5 that, parameter determination takes influence on the performance of WLRM. Different combinations of parameters may result in different classification models. Then, the AUC and ACC results change.

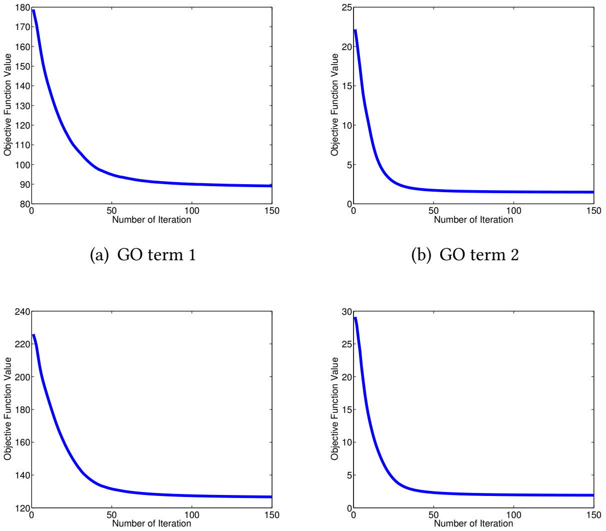
4.4 Convergence and Time Complexity

In this subsection, we take two groups of experiments. One group is convergence analysis experiments. We present the convergence characteristics of our WLRM on four different GO terms in Fig. 6. The objective function values are nonincreasing during the iterations. Our proposed method converges within 100 iterations.

Another group is the computational time of WLRM, miSVM, miFV and miVLAD on human genome dataset. All the algorithms are tested on a laptop with 4 processors (2.27 GHz for each) and 8 GB available RAM memory by Matlab implementations. The results are shown in Table 2. The results in Table 2 indicate that the running time of miSVM, miFV, miVLAD and WLRM increases linearly with the increase of the number of positive genes. Moreover, the running time of other MIL methods are at least 4.30 times of the running time of WLRM. When the number of positive genes is very small, i.e., the number in group A is from 20 to 27, the running time of other compared methods are at least 10.68 times of WLRM, especially the running time of miSVM is 31.39 times of WLRM. Overall, WLRM is much more efficient than others.

Table 2: Average running time ± the standard derivation (in seconds) of miSVM, miFV, miVLAD and WLRM for training the model on five groups of GO terms. The last row is the speedup of WLRM with respect to the runtime of the fastest one among other three methods.

Methods	Group A	Group B	Group C	Group D	Group E
miSVM	6340.0 ± 161.68	6385.5±243.14	6418.2±109.46	6455.2±215.87	6529.5±236.74
miFV	2872.0 ± 179.32	3169.7±198.71	4312.2±112.50	5693.5±197.89	8315.1±213.63
miVLAD	2157.6 ± 89.70	2661.5±135.43	3579.9±96.67	4939.3±110.87	6410.5±135.78
WLRM	**201.9±55.91**	**304.6±81.67**	**490.3±89.67**	**833.2±135.47**	**1489.7±283.67**
Speedup	**10.68**	**8.74**	**7.30**	**5.93**	**4.30**



(a) GO term 1

(b) GO term 2

(c) GO term 3

(d) GO term 4

Figure 6: Number of iterations vs. the objective value of WLRM on two different GO terms.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we develop a novel unified MIL framework to annotate functions of human PCIs. Based on this framework, we proposed a new method called WLRM based on the logistic loss. Specifically, we introduce an isoform weight vector for each positive gene and derive a nonconvex sparsity-inducing regularizes, which includes non-negative, l_0-norm and l_1-norm constraints on each isoform weights vector. The proposed method detects the key isoforms and embeds the original gene space into a discriminative feature space simultaneously. The isoform weight vectors can be obtained by sparse projections onto a simplex. Meanwhile, we develop an efficient block coordinate descent approach to solve our non-convex optimization problem. Finally, our WLRM is applied to predict the functions of human PCIs. There are several interesting directions to investigate in the future. First, we would like to find a more efficient and effective way of dealing with our non-convex optimization problem. Second, we would like to extend our model to the nonlinear case using kernel trick.

REFERENCES

[1] Robert A Amar, Daniel R Dooly, Sally A Goldman, and Qi Zhang. 2001. Multiple-instance learning of real-valued data. In *International Conference on Machine learning*. 3–10.

[2] Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* 201 (2013), 81–105.

[3] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2002. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*. 561–568.

[4] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, and others. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25, 1 (2000), 25–29.

[5] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2, 1 (2009), 183–202.

[6] Ko-Fan Chen and Damian C Crowther. 2012. Functional genomics in Drosophila models of human disease. *Briefings in Functional Genomics* 11, 5 (2012), 405–415.

[7] Yixin Chen, Jinbo Bi, and James Z Wang. 2006. MILES: Multiple-instance learning via embedded instance selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, 12 (2006), 1931–1947.

[8] Veronika Cheplygina, David MJ Tax, and Marco Loog. 2015. Multiple instance learning with bag dissimilarities. *Pattern Recognition* 48, 1 (2015), 264–275.

[9] ENCODE Project Consortium and others. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 7414 (2012), 57–74.

[10] Melissa J Davis, Muhammad SB Sehgal, and Mark A Ragan. 2010. Automatic, context-specific generation of Gene Ontology slims. *BMC Bioinformatics* 11, 1 (2010), 498.

[11] Eleonora de Klerk and Peter A.C. 't Hoen. 2015. Alternative mRNA transcription, processing, and translation: insights from RNA sequencing. *Trends in Genetics* 31, 3 (2015), 128–139.

[12] Thomas Derrien, Rory Johnson, Giovanni Bussotti, Andrea Tanzer, Sarah Djebali, Hagen Tilgner, Gregory Guernec, David Martin, Angelika Merkel, David G Knowles, and others. 2012. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Research* 22, 9 (2012), 1775–1789.

[13] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 1 (1997), 31–71.

[14] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine learning*. ACM, 272–279.

[15] Ridvan Eksi, Hong-Dong Li, Rajasree Menon, Yuchen Wen, Gilbert S. Omenn, Matthias Kretzler, and Yuanfang Guan. 2013. Systematically Differentiating Functions for Alternatively Spliced Isoforms through Integrating RNA-seq Data. *PLoS Comput Biol* 9, 11 (11 2013), e1003314.

[16] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874.

[17] Robert Fung and Kuo-Chu Chang. 2013. Weighing and integrating evidence for stochastic simulation in Bayesian networks. *arXiv preprint arXiv:1304.1504* (2013).

[18] Gene H Golub, Michael Heath, and Grace Wahba. 1979. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* 21, 2 (1979), 215–223.

[19] Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, and others. 2012. GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Research* 22, 9 (2012), 1760–1774.

[20] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications* 13, 4 (1998), 18–28.

[21] David W Hosmer Jr and Stanley Lemeshow. 2004. *Applied logistic regression.* John Wiley & Sons.

[22] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and others. 2008. KEGG for linking genomes to life and the

environment. *Nucleic Acids Research* 36, suppl 1 (2008), D480–D484.

[23] Minoru Kanehisa, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe. 2011. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research* (2011), gkr988.

[24] Anastasios T Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. 2013. Sparse projections onto the simplex.. In *Proceedings of The 30th International Conference on Machine Learning*, Vol. 28. JMLR, 235–243.

[25] Hong-Dong Li, Rajasree Menon, Gilbert S Omenn, and Yuanfang Guan. 2014. The emerging era of genomic data integration for analyzing splice isoform function. *Trends in Genetics* 30, 8 (2014), 340–347.

[26] Wenyuan Li, Shuli Kang, Chun-Chi Liu, Xianghong Jasmine Zhou, and others. 2014. High-resolution functional annotation of human transcriptome: predicting isoform functions by a novel multiple instance-based label propagation method. *Nucleic Acids Research* 42, 6 (2014), e39.

[27] Yan Li, David MJ Tax, Robert PW Duin, and Marco Loog. 2013. Multiple-instance learning as a classifier combining problem. *Pattern Recognition* 46, 3 (2013), 865–874.

[28] Guoqing Liu, Jianxin Wu, and Zhi-Hua Zhou. 2012. Key instance detection in multi-instance learning. In *ACML*, Vol. 25. 253–268.

[29] Oded Maron and Tomás Lozano-Pérez. 1998. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems* (1998), 570–576.

[30] Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady* 27 (1983). Issue 2.

[31] Bharat Panwar, Rajasree Menon, Ridvan Eksi, Hong-Dong Li, Gilbert S Omenn, and Yuanfang Guan. 2016. Genome-wide functional annotation of human protein-coding splice variants using multiple instance learning. *Journal of Proteome Research* 15, 6 (2016), 1747–1753.

[32] Gunnar Rätsch, Takashi Onoda, and K-R Müller. 2001. Soft margins for AdaBoost. *Machine Learning* 42, 3 (2001), 287–320.

[33] Burr Settles, Mark Craven, and Soumya Ray. 2008. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*. 1289–1296.

[34] Timothy Sterne-Weiler and Jeremy R Sanford. 2014. Exon identity crisis: disease-causing mutations that disrupt the splicing code. *Genome Biology* 15, 1 (2014), 201.

[35] Cole Trapnell, Lior Pachter, and Steven L Salzberg. 2009. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25, 9 (2009), 1105–1111.

[36] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols* 7, 3 (2012), 562–578.

[37] Kai Wang, Mingyao Li, and Hakon Hakonarson. 2010. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* 38, 16 (2010), e164–e164.

[38] Xiushen Wei, Jianxin Wu, and Zhihua Zhou. 2014. Scalable multi-instance learning. In *2014 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1037–1042.

[39] Xiushen Wei, Jianxin Wu, and Zhihua Zhou. 2017. Scalable algorithms for multi-instance learning. *IEEE Transactions on Neural Networks and Learning Systems* 28, 4 (2017), 975–987.

[40] Hui Y Xiong, Babak Alipanahi, Leo J Lee, Hannes Bretschneider, Daniele Merico, Ryan KC Yuen, Yimin Hua, Serge Gueroussov, Hamed S Najafabadi, Timothy R Hughes, and others. 2015. The human splicing code reveals new insights into the genetic determinants of disease. *Science* 347, 6218 (2015).

[41] Yin Wotao Xu, Yangyang. 2017. A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update. *Journal of Scientific Computing* (2017), 1–35.

[42] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320.