Analytical Biochemistry 473 (2015) 14–27

Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

mPLR-Loc: An adaptive decision multi-label classifier based on penalized logistic regression for protein subcellular localization prediction

Shibiao Wan [a], Man-Wai Mak [a,*], Sun-Yuan Kung [b]

[a] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
[b] Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA

ARTICLE INFO

Article history:
Received 6 March 2014
Received in revised form 29 September 2014
Accepted 21 October 2014
Available online 31 October 2014

Keywords:
Protein subcellular localization
Multi-location proteins
Adaptive decision
Logistic regression
Multi-label classification

ABSTRACT

Proteins located in appropriate cellular compartments are of paramount importance to exert their biological functions. Prediction of protein subcellular localization by computational methods is required in the post-genomic era. Recent studies have been focusing on predicting not only single-location proteins but also multi-location proteins. However, most of the existing predictors are far from effective for tackling the challenges of multi-label proteins. This article proposes an efficient multi-label predictor, namely mPLR-Loc, based on penalized logistic regression and adaptive decisions for predicting both single- and multi-location proteins. Specifically, for each query protein, mPLR-Loc exploits the information from the Gene Ontology (GO) database by using its accession number (AC) or the ACs of its homologs obtained via BLAST. The frequencies of GO occurrences are used to construct feature vectors, which are then classified by an adaptive decision-based multi-label penalized logistic regression classifier. Experimental results based on two recent stringent benchmark datasets (virus and plant) show that mPLR-Loc remarkably outperforms existing state-of-the-art multi-label predictors. In addition to being able to rapidly and accurately predict subcellular localization of single- and multi-label proteins, mPLR-Loc can also provide probabilistic confidence scores for the prediction decisions. For readers' convenience, the mPLR-Loc server is available online (http://bioinfo.eie.polyu.edu.hk/mPLRLocServer).

Introduction

Proteins need to be at the right spatiotemporal context within a cell to properly exert their biological functions. The information of protein subcellular localization is vitally important for understanding the functions of proteins and for identifying drug targets [1,2]. Aberrant protein subcellular localization is closely correlated to a broad range of human diseases such as Alzheimer's disease [3], kidney stone [4], primary human liver tumors [5], breast cancer [6], minor salivary gland tumors [7], pre-eclampsia [8], and Bartter syndrome [9]. To tackle the avalanche of newly discovered protein sequences in the post-genomic era, computational methods are required to assist or replace time-consuming and laborious wet-lab experiments such as fluorescent microscopy imaging, cell fractionation, and electron microscopy for predicting the subcellular locations of proteins.

Conventional methods for protein subcellular localization prediction can be roughly divided into sequence-based and knowledge-based. Sequence-based methods include (i) sorting-signals-based methods [10–12], (ii) homology-based methods [13–16], and (iii) composition-based methods [17,18]. Knowledge-based methods use information from knowledge databases such as using Gene Ontology (GO)[1] terms [19–29], Swiss–Prot keywords [30,13], and PubMed abstracts [31,32]. Although it is possible that the GO information may become less reliable when the proteins are with high sequence similarity but have diverse functions, it has been demonstrated that methods based on GO information are superior to methods based on other features [22].

Because there exist multi-location proteins that can simultaneously reside at, or move between, two or more subcellular locations, recent studies have focused on predicting both single-location and multi-location proteins. It is generally accepted that it is inappropriate to exclude the multi-label proteins or assume that multi-location proteins do not exist. Actually, multi-location proteins play important roles in some metabolic processes that

* Corresponding author.
E-mail address: enmwmak@polyu.edu.hk (M.-W. Mak).

[1] *Abbreviations used:* GO, Gene Ontology; ECC, ensemble of classifier chain; LP, label powerset; BR, binary relevance; SVM, support vector machine; KNN, K-nearest neighbor; AC, accession number; GOA, Gene Ontology Annotation; LR, logistic regression; LOOCV, leave-one-out cross-validation; OET, optimized evidence-theoretic.

http://dx.doi.org/10.1016/j.ab.2014.10.014
0003-2697/

take place in more than one cellular compartment (e.g., β-oxidation in the peroxisome and mitochondria; antioxidant defense in the cytosol, mitochondria, and peroxisome) [33]. Multi-label models have also been applied to identifying membrane proteins with both single and multiple functional types [34].

Existing multi-label classification models can be grouped into two main categories: (i) algorithm adaptation and (ii) problem transformation. Algorithm adaptation methods extend specific single-label algorithms to solve multi-label classification problems. Typical methods include multi-label C4.5 [35], AdaBoost.MH [36], and hierarchical multi-label decision trees [37]. Problem transformation methods transform a multi-label learning problem into one or more single-label classification problems [38] so that traditional single-label classifiers can be applied without modification. Typical methods include ensembles of classifier chains (ECCs) [39], label powerset (LP) [40], compressive sensing [41], and binary relevance (BR) [42]. Among them, BR is one of the most popular methods. For example, the binary relevance support vector machine (SVM)-based model transforms a multi-label problem into a number of binary classification problems, one for each label. Each binary classification problem is then handled by one binary SVM. Given a query instance, the predicted label is the union of the positive-class labels outputted by these binary SVMs.

Recently, several state-of-the-art multi-label predictors have been proposed to deal with the prediction of multi-label proteins, including Virus-mPLoc [43], Plant-mPLoc [44], iLoc-Virus [45], iLoc-Plant [46], K-nearest neighbor (KNN)-SVM ensemble classifier [47], mGOASVM [48], and other predictors [49–51]. They all use the GO (http://www.geneontology.org) information as the features and apply different multi-label classifiers to tackle the multi-label classification problem. However, these predictors provide only the final prediction results, and readers cannot obtain information (i.e., probabilistic confidence scores for each subcellular location) about how they make the prediction decisions.

This article proposes an efficient multi-label predictor, namely mPLR-Loc, for predicting subcellular localization of both single-label and multi-label proteins. Here, the prefix "m" stands for *multiple,* meaning that the predictor can deal with both single-label and multi-label proteins. Given a protein, a set of GO terms is retrieved by searching against the GO database using the accession numbers of homologous protein obtained via BLAST search as the keys. The frequencies of GO occurrences are used to formulate frequency vectors, which are then classified by a multi-label penalized logistic regression classifier equipped with an adaptive decision strategy. mPLR-Loc is different from existing state-of-the-art predictors in that (i) it uses a multi-label penalized logistic regression classifier equipped with an adaptive decision strategy that can tackle multi-label problems effectively, (ii) it not only rapidly and accurately provides the prediction results of subcellular localization for query proteins but also gives the probabilistic scores or confidence estimates for each of the subcellular locations, and (iii) it adopts a new strategy to incorporate richer and more useful homologous information from more distant homologs. Results on two recent benchmark datasets and a new independent test set demonstrate that these properties enable mPLR-Loc to substantially outperform other existing state-of-the-art predictors.

mPLR-Loc is designed for predicting viral and plant proteins. Actually, studying the subcellular localization of viral proteins can help biologists to obtain the information about their destructive tendencies and consequences [52]. The information of subcellular localization of *Viridiplantae* proteins is also crucial to elucidate their functions. As for predicting proteins of other species, because mPLR-Loc uses the information of GO terms that possess the cross-species properties [53], it is easy for mPLR-Loc to extend from predicting viral and plant proteins to predicting proteins of other species.

Legitimacy of using GO information

First, some people may be skeptical about using GO information for protein subcellular localization because the cellular component GO terms have already been annotated with cellular component categories. The GO comprises three orthogonal categories whose terms describe the cellular components, biological processes, and molecular functions of gene products. They argue that the only thing that needs to be done is to create a lookup table using the cellular component GO terms as the keys and the component categories as the hashed values. Such a naive solution, however, is undesirable and will lead to poor performance, as shown and explained in our previous studies [48,54].

Second, some people disprove the effectiveness of GO-based methods by claiming that only cellular component GO terms are useful and GO terms in the other two categories play no role in determining the subcellular localization. This concern has been explicitly and directly addressed by Lu and Hunter [55], who demonstrated that GO molecular function terms are also predictive of subcellular localization, particularly for nucleus, extracellular space, membrane, mitochondrion, endoplasmic reticulum, and Golgi apparatus. The in-depth analyses of the correlation between the molecular function GO terms and localization in Ref. [55] also provide an explanation of why GO-based methods outperform sequence-based methods.

Third, even though GO-based methods can predict novel proteins based on the GO information obtained from their homologous proteins [48,54], some people still argue that the prediction is equivalent to simply using the annotated localization of the homologs (i.e., using BLAST [56] with homologous transfer). This claim was clearly proved to be untenable in our previous study [54], which demonstrated that GO-based methods remarkably outperform methods that use only BLAST and homologous transfer (Table 4 in Ref. [54]). Besides, Briesemeister and coworkers [57] also suggested that using BLAST alone is not sufficient for reliable prediction.

Moreover, as suggested by Chou [58], as long as the input of query proteins for predictors is the sequence information without any GO annotation information and the output is the subcellular localization information, there is no difference between non-GO-based methods and GO-based methods, which should be regarded as equally legitimate for subcellular localization.

Some other articles [59,60] also provide strong arguments supporting the legitimacy of using GO information for subcellular localization. In particular, as suggested in Ref. [60], the good performance of GO-based methods is due to the fact that the feature vectors in the GO space can better reflect their subcellular locations than those in the Euclidean space or any other simple geometric space.

Feature extraction

The subcellular localization predictors use GO information as the features, which have been demonstrated to be superior to other features [22,54,61]. The feature extraction part consists of two steps: (i) retrieval of GO terms and (ii) construction of GO vectors.

Retrieval of GO terms

For a query protein, mPLR-Loc can deal with two possible cases: (i) the accession number (AC) is known and (ii) only the amino acid sequence is known. For proteins with known ACs, their respective GO terms are retrieved from the Gene Ontology Annotation (GOA) database (http://www.ebi.ac.uk/GOA) using the ACs as the searching keys. For a protein without an AC, its amino acid sequence is

presented to BLAST [56] to find its homologs, whose ACs are then used as keys to search against the GOA database.

Although the GOA database allows us to associate the AC of a protein with a set of GO terms, for some novel proteins neither their ACs nor the ACs of their top homologs have any entries in the GOA database; in other words, no GO terms can be retrieved by their ACs or the ACs of their top homologs. In such cases, the ACs of the homologous proteins, as returned from BLAST search, will be successively used to search against the GOA database until a match is found. In cases where no GO terms can be retrieved by the ACs or even by the ACs of all the homologs, backup methods that rely on other features, such as pseudo-amino-acid composition [18] and sorting signals [62], should be used. Fortunately, with the rapid progress of the GOA database [63], it is reasonable to assume that the homologs of the query proteins can retrieve at least one GO term [24]. Thus, it is rarely necessary to use backup methods to handle the situation where no GO terms can be found. The procedures are outlined in Fig. 1.

Construction of GO vectors

Given a dataset, the GO terms of all of its proteins are retrieved by using the procedures described in the previous subsection. Then, the number of distinct GO terms corresponding to the dataset is determined. Suppose that T distinct GO terms are found; these GO terms form a GO Euclidean space with T dimensions. For each sequence in the dataset, a GO vector is constructed by matching its GO terms to all of the T GO terms. Unlike the conventional 1–0 value [43,44], in this work term frequency [54,64] is used to construct the GO vectors. Similar to the 1–0 value approach, a protein is represented by a point in a Euclidean space. However, unlike the 1–0 approach, the term frequency approach uses the number of occurrences of individual GO terms as the coordinates. Specifically, the GO vector $\mathbf{q}_i$ of the i-th protein $\mathbb{Q}_i$ is defined as

$$\mathbf{q}_i = [b_{i,1}, \cdots, b_{i,j}, \cdots, b_{i,T}]^{\mathrm{T}}, b_{i,j} = \begin{cases} f_{i,j} & \text{, GO hit} \\ 0 & \text{, otherwise} \end{cases} \quad (1)$$

where $f_{i,j}$ is the number of occurrences of the j-th GO term (term frequency) in the i-th protein sequence. The rationale is that the term frequencies may also contain important information for classification and, therefore, should not be quantized to either 0 or 1. Note that $b_{i,j}$ values are analogous to the term frequencies commonly used in document retrieval.

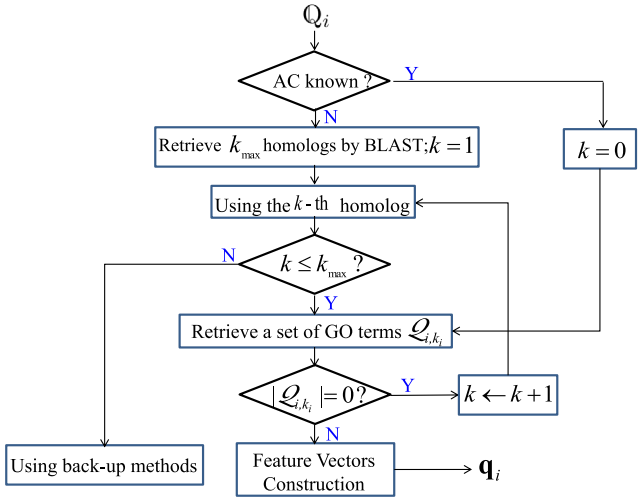

Fig.1. Procedures of retrieving GO terms. $\mathbb{Q}_i$: the i-th query protein; k_{max}: the maximum number of homologs retrieved by BLAST with the default parameter setting; $\mathbb{Q}_{i,ki}$: the set of GO terms retrieved by BLAST using the k_i-th homolog for the i-th query protein $\mathbb{Q}_i$; k_i: the k_i-th homolog used to retrieve the GO terms; $\mathbf{q}_i$: the output GO vector.

Multi-label penalized logistic regression classifier

Logistic regression (LR) is a powerful discriminative classifier that has a direct and explicit probabilistic interpretation built into its model [65]. Traditional logistic regression classifiers, including penalized logistic regression classifiers [66–68], are applicable only to multi-class classification. This section elaborates an efficient penalized multi-label logistic regression classifier, namely mPLR-Loc, equipped with an adaptive decision scheme.

Single-label penalized logistic regression

Suppose that for a two-class single-label problem, we are given a set of training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^{T+1}$ and $y_i \in \{0, 1\}$. In our case, $\mathbf{x}_i = \begin{bmatrix} 1 \\ \mathbf{q}_i \end{bmatrix}$, where $\mathbf{q}_i$ is defined in Eq. (1). Denote $Pr(Y = y_i|X = \mathbf{x}_i)$ as the posterior probability of the event that X belongs to class y_i given $X = \mathbf{x}_i$. In logistic regression, the posterior probability is defined as

$$Pr(Y = y_i|X = \mathbf{x}_i) = p(\mathbf{x}_i; \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i}}, \quad (2)$$

where $\boldsymbol{\beta}$ is a $(T + 1)$-dim parameter vector. When the number of training instances (N) is not significantly larger than the feature dimension $(T + 1)$, using logistic regression without any regularization often leads to over-fitting. To avoid over-fitting, an L_2 regularization penalty term is added to the penalized cross-entropy error function as follows:

$$\begin{aligned} E(\boldsymbol{\beta}) &= -\sum_{i=1}^{N}[y_i \log(p(\mathbf{x}_i; \boldsymbol{\beta})) + (1 - y_i) \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta}))] + \frac{1}{2}\rho\|\boldsymbol{\beta}\|_2^2 \\ &= -\sum_{i=1}^{N}\left[y_i\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i - \log(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i})\right] + \frac{1}{2}\rho\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} \end{aligned} \quad (3)$$

where ρ is a user-defined penalty parameter to control the degree of regularization.

To minimize $E(\boldsymbol{\beta})$, we may use the Newton–Raphson algorithm

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - \left(\frac{\partial^2 E(\boldsymbol{\beta}^{old})}{\partial\boldsymbol{\beta}^{old}\partial(\boldsymbol{\beta}^{old})^{\mathrm{T}}}\right)^{-1} \cdot \frac{\partial E(\boldsymbol{\beta}^{old})}{\partial\boldsymbol{\beta}^{old}}, \quad (4)$$

where

$$\frac{\partial E(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = -\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{p}) + \rho\boldsymbol{\beta} \quad (5)$$

and

$$\frac{\partial^2 E(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathrm{T}}} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \rho\mathbf{I} \quad (6)$$

See Appendix A for the derivations of Eqs. (5) and (6). In Eqs. (5) and (6), $\mathbf{y}$ and $\mathbf{p}$ are N-dim vectors whose elements are $\{y_i\}_{i=1}^N$ and $\{p(\mathbf{x}_i; \boldsymbol{\beta})\}_{i=1}^N$, respectively, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^{\mathrm{T}}$, and $\mathbf{W}$ is a diagonal matrix whose i-th diagonal element is $p(\mathbf{x}_i;\boldsymbol{\beta})[1 - p(\mathbf{x}_i;\boldsymbol{\beta})]$, $i = 1, 2, \ldots, N$.

Substituting Eqs. (5) and (6) into Eq. (4) gives the following iterative formula for estimating $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} + (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{p}) - \rho\boldsymbol{\beta}^{old}). \quad (7)$$

Multi-label penalized logistic regression

In an M-class multi-label problem, the training data set is written as $\{\mathbf{x}_i, \mathcal{Y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^{T+1}$ and $\mathcal{Y}_i \subset \{1, 2, \ldots, M\}$ is a set that may contain one or more labels. M independent binary one-versus-rest LRs are trained, one for each class. The labels $\{\mathcal{Y}_i\}_{i=1}^N$ are converted to *transformed labels* [48] $y_{i,m} \in \{0, 1\}$, where $i = 1, \ldots, N$ and $m = 1, \ldots, M$. The two-class update formula in Eq. (7) is then extended to

$$\boldsymbol{\beta}_m^{new} = \boldsymbol{\beta}_m^{old} + (\mathbf{X}^\top \mathbf{W}_m \mathbf{X} + \rho \mathbf{I})^{-1} (\mathbf{X}^\top (\mathbf{y}_m - \mathbf{p}_m) - \rho \boldsymbol{\beta}_m^{old}), \tag{8}$$

where $m = 1, \ldots, M$, $\mathbf{y}_m$ and $\mathbf{p}_m$ are vectors whose elements are $\{y_{i,m}\}_{i=1}^N$ and $\{p(\mathbf{x}_i; \boldsymbol{\beta}_m)\}_{i=1}^N$, respectively, and $\mathbf{W}_m$ is a diagonal matrix whose i-th diagonal element is $p(\mathbf{x}_i;\boldsymbol{\beta}_m)[1-p(\mathbf{x}_i;\boldsymbol{\beta}_m)]$, $i = 1, 2, \ldots, N$.

Given the i-th GO vector $\mathbf{q}_i$ of the query protein $\mathbb{Q}_i$, the score of the m-th LR is given by

$$s_m(\mathbb{Q}_i) = \frac{e^{\boldsymbol{\beta}_m^\top \mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}_m^\top \mathbf{x}_i}}, \text{where } \mathbf{x}_i = \begin{bmatrix} 1 \\ \mathbf{q}_i \end{bmatrix}. \tag{9}$$

The probabilistic nature of logistic regression enables us to assign confidence scores for the prediction decisions. Specifically, for the m-th location, its corresponding confidence score is $s_m(\mathbb{Q}_i)$. See Appendix B for the confidence scores produced by the mPLR-Loc server.

Adaptive Decision for LR (mPLR-Loc)

Because the LR scores of a binary LR classifier are posterior probabilities, the m-th class label will be assigned to $\mathbb{Q}_i$ only if $s_m(\mathbb{Q}_i) > 0.5$. To facilitate multi-label classification, the following decision scheme is adopted:

$$\mathcal{M}(\mathbb{Q}_i) = \bigcup_{m=1}^{M} \{\{m : s_m(\mathbb{Q}_i) > 0.5\} \cup \{m : s_m(\mathbb{Q}_i) \geqslant f(s_{\max}(\mathbb{Q}_i))\}\}, \tag{10}$$

where $f[s_{\max}(\mathbb{Q}_i)]$ is a function of $s_{\max}(\mathbb{Q}_i)$ and $s_{\max}(\mathbb{Q}_i) = \max_{m=1}^{M} s_m(\mathbb{Q}_i)$. In this work, we used a linear function as follows:

$$f(s_{\max}(\mathbb{Q}_i)) = \theta s_{\max}(\mathbb{Q}_i), \tag{11}$$

where $\theta \in (0.0, 1.0)$ is a parameter that can be optimized by using cross-validation experiments. Note that θ cannot be 0.0, or otherwise all of the M labels will be assigned to $\mathbb{Q}_i$. This is because $s_m(\mathbb{Q}_i)$ is a posterior probability, which is always equal to or greater than zero. Clearly, Eq. (10) suggests that the predicted labels depend on $s_{\max}(\mathbb{Q}_i)$, a function of the test instance (or protein). This means that the decision and its corresponding threshold are adaptive to the test protein. For ease of reference, we refer to this predictor as mPLR-Loc.

Experiments

Datasets

In this article, a virus dataset [43,45] and a plant dataset [46] were used to evaluate the performance of the proposed predictors. The virus and plant datasets were created from Swiss–Prot 57.9 and 55.3, respectively. The virus dataset contains 207 viral proteins distributed in six locations. Of the 207 viral proteins, 165 belong to one subcellular location, 39 belong to two locations, 3 belong to three locations, and none belongs to four or more locations. This means that approximately 20% of the proteins in the dataset are located in more than one subcellular location. The plant dataset contains 978 plant proteins distributed in 12 locations. Of the 978 plant proteins, 904 belong to one subcellular location, 71 belong to two locations, 3 belong to three locations, and none belongs to four or more locations. The sequence identity of both datasets was cut off at 25%. The breakdowns of these two datasets are listed in Figs. 2 and 3. As can be seen, both datasets are multi-class distributed and imbalanced. More detailed statistical properties of these two datasets are listed in Table 1.

In Table 1, M and N denote the number of actual (or distinct) subcellular locations and the number of actual (or distinct) proteins. Besides the commonly used properties for single-label classification, the following measurements [40] are used as well to explicitly quantify the multi-label properties of the datasets:

1. *Label cardinality (LC)*: LC is the average number of labels per data instance, which is defined as $LC = \frac{1}{N}\sum_{i=1}^{N} |\mathcal{L}(\mathbb{Q}_i)|$, where $\mathcal{L}(\mathbb{Q}_i)$ is the label set of the protein $\mathbb{Q}_i$ and $|\cdot|$ denotes the cardinality of a set.
2. *Label density (LD)*: LD is LC normalized by the number of classes, which is defined as $LD = \frac{LC}{M}$.
3. *Distinct label set (DLS)*: DLS is the number of label combinations in the dataset.

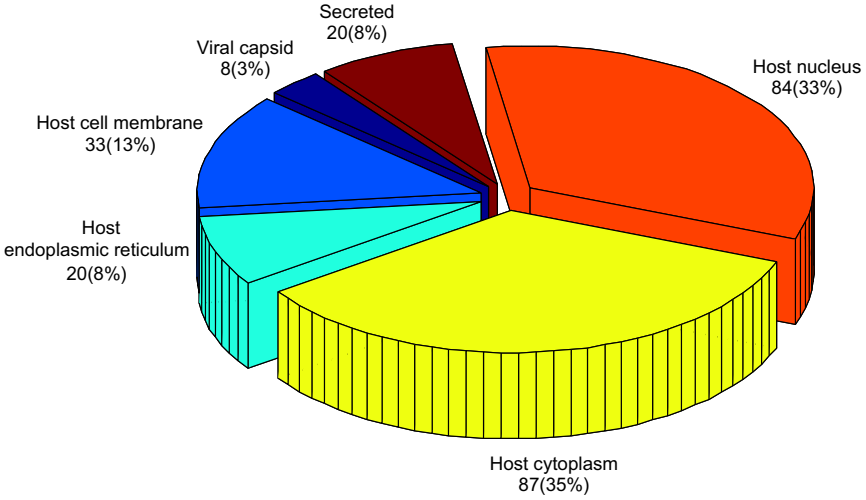

Fig.2. Breakdown of the virus dataset. The number of proteins shown in each subcellular location represents the number of "locative proteins" [45,48]. Here, 207 actual proteins have 252 locative proteins.

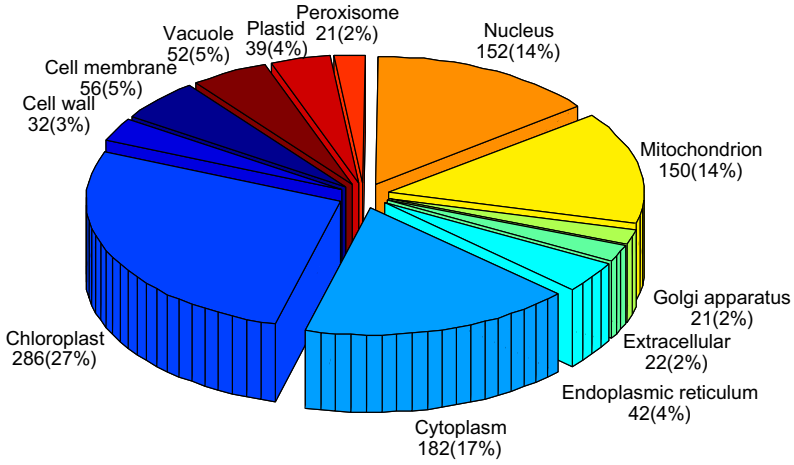

Fig.3. Breakdown of the plant dataset. The number of proteins shown in each subcellular location represents the number of "locative proteins" [45,48]. Here, 978 actual proteins have 1055 locative proteins.

Table 1
Statistical properties of the two datasets used in our experiments.

Dataset	*M*	*N*	*LC*	*LD*	*DLS*	*PDLS*	*TLN*
Virus	6	207	1.2174	0.2029	17	0.0821	252
Plant	12	978	1.0787	0.0899	32	0.0327	1055

Note. M, number of subcellular locations; N, number of actual proteins; *LC*, label cardinality; *LD*, label density; *DLS*, distinct label set; *PDLS*, proportion of distinct label set; *TLN*, total locative number.

4. *Proportion of distinct label set (PDLS)*: *PDLS* is *DLS* normalized by the number of actual data instances, which is defined as $PDLS = \frac{DLS}{N}$.
5. *Total locative number (TLN)*: *TLN* is the total number of locative proteins. This concept is derived from locative proteins in Ref. [45], which is further elaborated in the next subsection.

Among these measurements, *LC* is used to measure the degree of multi-labels in a dataset. For a single-label dataset, $LC = 1$; for a multi-label dataset, $LC > 1$. And the larger the *LC*, the higher the degree of multi-labels. *LD* takes into consideration the number of classes in the classification problem. For two datasets with the same *LC*, the lower the *LD*, the more difficult the classification. *DLS* represents the number of possible label combinations in the dataset. The higher the *DLS*, the more complicated the composition. *PDLS* represents the degree of distinct labels in a dataset. The larger the *PDLS*, the more probable the individual label sets are different from each other. From Table 1, we notice that although the number of proteins in the virus dataset ($N = 207$, $TLN = 252$) is smaller than that of the plant dataset ($N = 978$, $TLN = 1055$), the former ($LC = 1.2174$, $LD = 0.2029$) is a denser multi-label dataset than the latter ($LC = 1.0787$, $LD = 0.0899$).

Performance metrics

Compared with traditional single-label classification, multi-label classification requires more complicated performance metrics to better reflect the multi-label capabilities of classifiers. These measures include *Accuracy*, *Precision*, *Recall*, *F1 Score* (*F1*), and *Hamming Loss* (*HL*). Specifically, denote $\mathcal{L}(\mathbb{Q}_i)$ and $\mathcal{M}(\mathbb{Q}_i)$ as the true label set and the predicted label set for the i-th protein $\mathbb{Q}_i$ ($i = 1, \ldots, N$), respectively.[2] Then the five measurements are defined as follows:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)|}\right) \tag{12}$$

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i)|}\right) \tag{13}$$

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{L}(\mathbb{Q}_i)|}\right) \tag{14}$$

$$F1 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{2|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i)| + |\mathcal{L}(\mathbb{Q}_i)|}\right) \tag{15}$$

$$HL = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| - |\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{M}\right) \tag{16}$$

where $|\cdot|$ means counting the number of elements in the set therein and $\cap$ represents the intersection of sets.

Accuracy, *Precision*, *Recall*, and *F1* indicate the classification performance. The higher the measures, the better the prediction performance. Among them, *Accuracy* is the most commonly used criterion. *F1* is the harmonic mean of *Precision* and *Recall*, allowing us to compare the performance of classification systems by taking the trade-off between *Precision* and *Recall* into account. The *HL* [69,70] is different from other metrics. As can be seen from Eq. (16), when all of the proteins are correctly predicted, that is, $|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| = |\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|$ ($i = 1, \ldots, N$), then $HL = 0$, whereas other metrics will be equal to 1. On the other hand, when the predictions of all proteins are completely wrong, that is, $|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| = M$ and $|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)| = 0$, then $HL = 1$, whereas other metrics will be equal to 0. Therefore, the lower the *HL*, the better the prediction performance.

Two additional measurements [45,48] are often used in multi-label subcellular localization prediction. They are overall locative accuracy (*OLA*) and overall actual accuracy (*OAA*). The former is given by

$$OLA = \frac{1}{\sum_{i=1}^{N}|\mathcal{L}(\mathbb{Q}_i)|}\sum_{i=1}^{N}|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|, \tag{17}$$

and the latter is given by

$$OAA = \frac{1}{N}\sum_{i=1}^{N}\Delta[\mathcal{M}(\mathbb{Q}_i), \mathcal{L}(\mathbb{Q}_i)] \tag{18}$$

[2] Here, $N = 207$ for the virus dataset and $N = 978$ for the plant dataset.

where

$$\Delta[\mathcal{M}(\mathbb{Q}_i),\mathcal{L}(\mathbb{Q}_i)]=\begin{cases}1 & ,\text{if } \mathcal{M}(\mathbb{Q}_i)=\mathcal{L}(\mathbb{Q}_i)\\ 0 & ,\text{otherwise.}\end{cases} \quad (19)$$

According to Eq. (17), a locative protein is considered to be correctly predicted if any of the predicted labels matches any labels in the true label set. On the other hand, Eq. (18) suggests that an actual protein is considered to be correctly predicted only if *all* of the predicted labels match those in the true label set exactly. For example, for a protein coexisting in, say, three subcellular locations, if only two of the three are correctly predicted, or the predicted result contains a location not belonging to the three, the prediction is considered as incorrect. In other words, when and only when all of the subcellular locations of a query protein are exactly predicted without any over-prediction or under-prediction can the prediction be considered as correct. Therefore, *OAA* is a more stringent measure as compared with *OLA*. *OAA* is also more objective than *OLA*. This is because locative accuracy is liable to give biased performance measure when the predictor tends to over-predict, that is, giving large $|\mathcal{M}(\mathbb{Q}_i)|$ for many $\mathbb{Q}_i$. In the extreme case, if every protein is predicted to have all of the M subcellular locations, according to Eq. (17) the *OLA* is 100%. But obviously the predictions are wrong and meaningless. On the contrary, *OAA* is 0% in this extreme case, definitely reflecting the real performance.

Among all of the metrics mentioned above, *OAA* is the most stringent and objective. This is because if some (but not all) of the subcellular locations of a query protein are correctly predicted, the numerators of the other four measures (Eqs. (12)–(17)) are non-zero, whereas the numerator of *OAA* in Eq. (18) is 0 (thereby contributing nothing to the frequency count). Note that *OAA* and *HL* are equivalent to *absolute true* and *absolute false*, respectively, used in Ref. [58].

In statistical prediction, leave-one-out cross-validation (LOOCV) is considered to be the most rigorous and bias-free method [71]. Hence, LOOCV was used to examine the performance of mPLR-Loc.

Results and discussion

Effect of adaptive decisions on mPLR-Loc

Fig. 4A shows the performance of mPLR-Loc on the virus dataset for different values of θ (Eq. (11)) based on leave-one-out cross-validation. In all cases, the penalty parameter ρ of the logistic regression was set to 1.0. The performance of mPLR-Loc at $\theta = 0.0$ is not provided because according to Eq. (10) and Eq. (11) all of the query proteins will be predicted as having all of the M subcellular locations, defeating the purpose of prediction. As is evident from Fig. 4A, when θ increases from 0.1 to 1.0, the *OAA* of mPLR-Loc increases first, reaches the peak at $\theta = 0.5$, with *OAA* = 0.903, which is nearly 2% (absolute) higher than mGOASVM (0.889). The *Precision* achieved by mPLR-Loc increases until $\theta = 0.5$ and then remains almost unchanged when $\theta \geqslant 0.5$. On the contrary, *OLA* and *Recall* peak at $\theta = 0.1$, and these measures drop with θ until $\theta = 1.0$. Among these metrics, no matter how θ changes, *OAA* is no higher than the other five measurements.

An analysis of the predicted labels $\{\mathcal{L}(\mathbb{Q}_i); i = 1,\ldots,207\}$ suggests that the increase in *OAA* is due to the reduction in the number of over-prediction, that is, the number of cases where $|\mathcal{M}(\mathbb{Q}_i)|>|\mathcal{L}(\mathbb{Q}_i)|$. When $\theta > 0.5$, the benefit of reducing the over-prediction diminishes because the criterion in Eq. (10) becomes so stringent that some of the proteins were under-predicted, that is, the number of cases where $|\mathcal{M}(\mathbb{Q}_i)| < |\mathcal{L}(\mathbb{Q}_i)|$. When θ increases from 0.1 to 0.5, the number of cases where $|\mathcal{M}(\mathbb{Q}_i)| > |\mathcal{L}(\mathbb{Q}_i)|$ decreases while at the same time $|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|$ remains almost unchanged. In other words, the denominators of *Accuracy* and *F1* decrease while the numerators for both metrics remain almost unchanged, leading to better performance for both metrics. When $\theta > 0.5$, for the similar reason mentioned above, the increase in under-prediction outweighs the benefit of the reduction in over-prediction, causing performance loss. For *Precision*, when $\theta > 0.5$, the loss due to the stringent criterion is counteracted by the gain due to the reduction in $|\mathcal{M}(\mathbb{Q}_i)|$, the denominator of Eq. (13). Thus, the *Precision* increases monotonically when θ increases from 0.1 to 1. However, *OLA* and *Recall* decrease monotonically with respect to θ because the denominator of these measures (see Eqs. (17) and (14)) is independent of $|\mathcal{M}(\mathbb{Q}_i)|$ and the number of correctly predicted labels in the numerator decreases when the decision criterion is getting stricter.

Fig. 4B shows the performance of mPLR-Loc (with $\rho = 1$) on the plant dataset. This figures shows that the trends of *OLA*, *Accuracy*, *Precision*, *Recall*, and *F1* are similar to those of mPLR-Loc in the virus dataset. The figure also shows that the *OAA* achieved by mPLR-Loc is monotonically increasing with respect to θ and reaches the optimum at $\theta = 1.0$, in contrast to the results in the virus dataset where the *OAA* is almost unchanged when $\theta \geqslant 0.5$.

Effect of regularization on mPLR-Loc

Fig. 5 shows the performance of mPLR-Loc with respect to the parameter ρ (Eq. (8)) on the virus dataset. In all cases, the adaptive

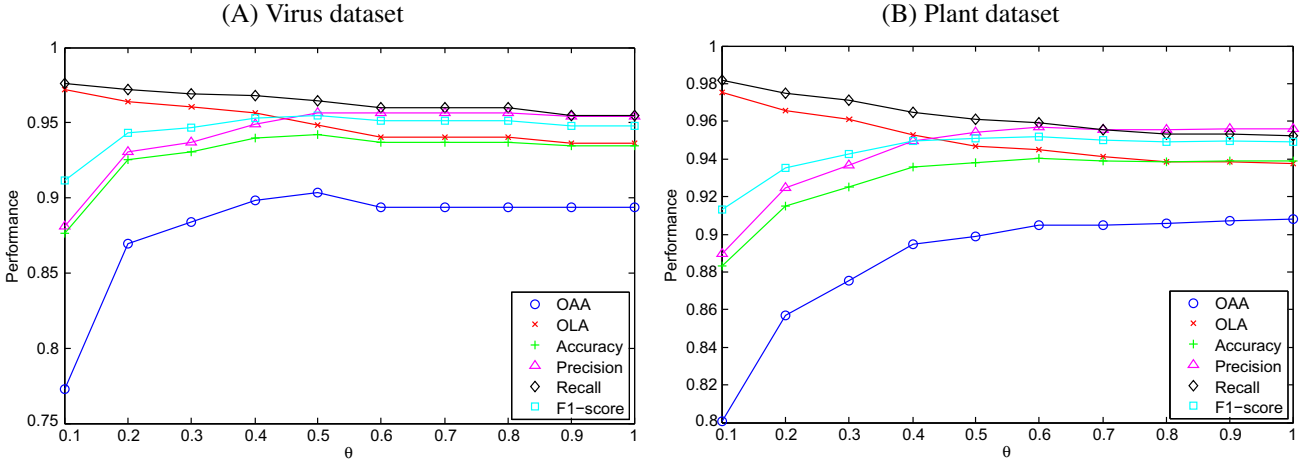

Fig.4. Performance of mPLR-Loc with respect to θ based on leave-one-out cross-validation on the virus dataset (A) and the plant dataset (B). See Eqs. (12)–(18) for the definitions of the performance measures in the boxed legend.

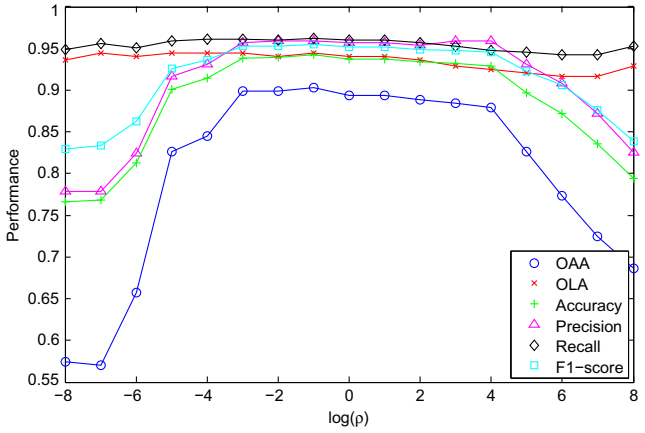

Fig.5. Performance of mPLR-Loc with respect to ρ in Eq. (8) based on leave-one-out cross-validation on the virus dataset. See Eqs. (12)–(18) for the definitions of the performance measures in the boxed legend.

thresholding parameter θ was set to 0.8. As can be seen, the variations of *OAA*, *Accuracy*, *Precision*, and *F1* with respect to ρ are very similar. More important, all four of these metrics show that there is a wide range of ρ values for which the performance is optimal. This suggests that introducing the penalty term in Eq. (3) not only helps to avoid numerical difficulty but also improves performance.

Fig. 5 shows that the *OLA* and *Recall* are largely unaffected by the change in ρ. This is understandable because the parameter ρ is to overcome numerical difficulty when estimating the LR parameters $\boldsymbol{\beta}$. More specifically, when ρ is small [say, $\log(\rho) < -5$], the value of ρ is insufficient to avoid matrix singularity in Eq. (7), which leads to extremely poor performance. When ρ is too large [say, $\log(\rho) > 5$], the matrix in Eq. (6) will be dominated by the value of ρ, which also causes poor performance. The *OAA* of mPLR-Loc reaches its maximum 0.903 at $\log(\rho) = -1$.

Comparing with state-of-the-art predictors

Tables 2 and 3 compare the performance of mPLR-Loc against several state-of-the-art multi-label predictors on the virus and plant dataset. All of these predictors derive the feature vectors from GO terms. From the classification perspective, Virus-mPLoc [43] uses an ensemble optimized evidence-theoretic (OET)-KNN classifier, iLoc-Virus [45] uses a multi-label KNN classifier, KNN-SVM [47] uses an ensemble of classifiers combining KNN and

Table 2
Comparing mPLR-Loc with state-of-the-art multi-label predictors based on leave-one-out cross-validation using the virus dataset.

Label	Subcellular location	LOOCV locative accuracy				
		Virus-mPLoc [44]	KNN-SVM [48]	iLoc-Virus [46]	mGOASVM [49]	mPLR-Loc
1	Viral capsid	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000
2	Host cell membrane	19/33 = 0.576	27/33 = 0.818	25/33 = 0.758	32/33 = 0.970	30/33 = 0.909
3	Host ER	13/20 = 0.650	15/20 = 0.750	15/20 = 0.750	17/20 = 0.850	17/20 = 0.850
4	Host cytoplasm	52/87 = 0.598	86/87 = 0.988	64/87 = 0.736	85/87 = 0.977	86/87 = 0.989
5	Host nucleus	51/84 = 0.607	54/84 = 0.651	70/84 = 0.833	82/84 = 0.976	81/84 = 0.964
6	Secreted	9/20 = 0.450	13/20 = 0.650	15/20 = 0.750	20/20 = 1.000	17/20 = 0.850
Overall actual accuracy (*OAA*)		–	–	155/207 = 0.748	184/207 = 0.889	187/207 = **0.903**
Overall locative accuracy (*OLA*)		152/252 = 0.603	203/252 = 0.807	197/252 = 0.782	244/252 = **0.968**	239/252 = 0.948
Accuracy		–	–	–	0.935	**0.942**
Precision		–	–	–	0.939	**0.957**
Recall		–	–	–	**0.973**	0.965
F1		–	–	–	0.950	**0.955**
HL		–	–	–	0.026	**0.023**

Note. –, the corresponding references do not provide the related metrics; Host ER, host endoplasmic reticulum. See Eqs. (12)–(18) for the definitions of the performance measures. The *P* value between the *OAA* of mPLR-Loc and mGOASVM on the virus dataset is 1.1750×10^{-4}.

Table 3
Comparing mPLR-Loc with state-of-the-art multi-label predictors based on leave-one-out cross-validation using the plant dataset.

Label	Subcellular location	LOOCV locative accuracy			
		Plant-mPLoc [45]	iLoc-Plant [47]	mGOASVM [49]	mPLR-Loc
1	Cell membrane	24/56 = 0.429	39/56 = 0.696	53/56 = 0.946	50/56 = 0.893
2	Cell wall	8/32 = 0.250	19/32 = 0.594	27/32 = 0.844	25/32 = 0.781
3	Chloroplast	248/286 = 0.867	252/286 = 0.881	272/286 = 0.951	281/286 = 0.983
4	Cytoplasm	72/182 = 0.396	114/182 = 0.626	174/182 = 0.956	164/182 = 0.901
5	Endoplasmic reticulum	17/42 = 0.405	21/42 = 0.500	38/42 = 0.905	35/42 = 0.833
6	Extracellular	3/22 = 0.136	2/22 = 0.091	22/22 = 1.000	19/22 = 0.864
7	Golgi apparatus	6/21 = 0.286	16/21 = 0.762	19/21 = 0.905	18/21 = 0.857
8	Mitochondrion	114/150 = 0.760	112/150 = 0.747	150/150 = 1.000	149/150 = 0.993
9	Nucleus	136/152 = 0.895	140/152 = 0.921	151/152 = 0.993	146/152 = 0.961
10	Peroxisome	14/21 = 0.667	6/21 = 0.286	21/21 = 1.000	21/21 = 1.000
11	Plastid	4/39 = 0.103	7/39 = 0.179	39/39 = 1.000	36/39 = 0.923
12	Vacuole	26/52 = 0.500	28/52 = 0.538	49/52 = 0.942	45/52 = 0.942
Overall actual accuracy (*OAA*)		–	666/978 = 0.681	855/978 = 0.874	888/978 = **0.908**
Overall locative accuracy (*OLA*)		672/1055 = 0.637	756/1055 = 0.717	1015/1055 = **0.962**	989/1055 = 0.937
Accuracy		–	–	0.926	**0.939**
Precision		–	–	0.933	**0.956**
Recall		–	–	**0.968**	0.952
F1		–	–	0.942	**0.949**
HL		–	–	0.013	**0.010**

Note. –, the corresponding references do not provide the related metrics. See Eqs. (12)–(18) for the definitions of the performance measures. The *P* value between the *OAA* of mPLR-Loc and mGOASVM on the plant dataset is 7.262×10^{-7}.

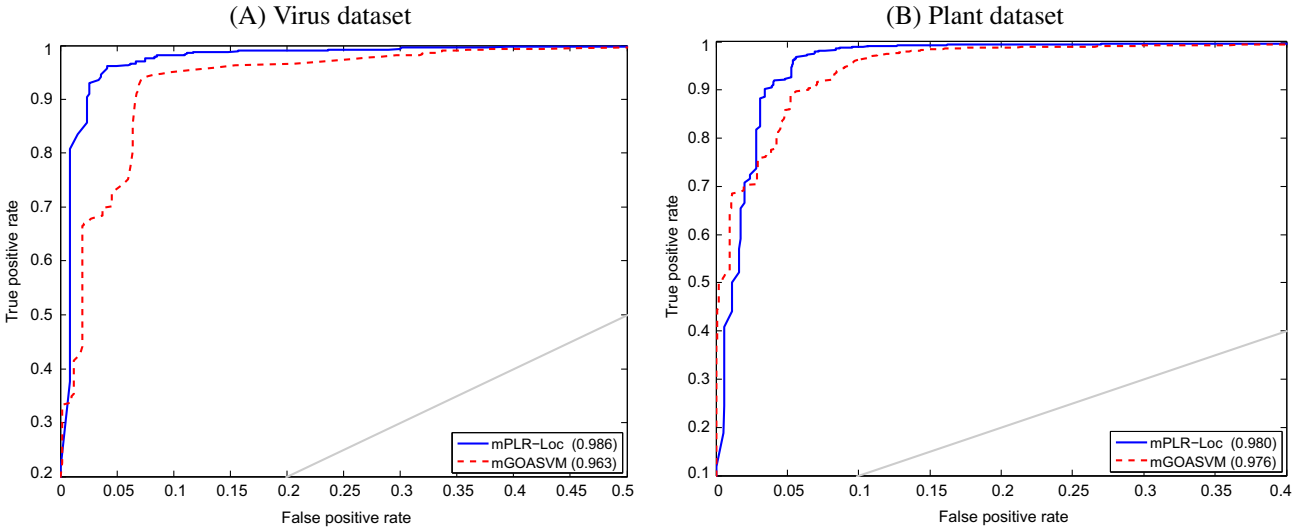

Fig.6. Receiver operating characteristic (ROC) curves of mPLR-Loc and mGOASVM on the virus dataset (A) and the plant dataset (B). The values in parentheses in the boxed legend are the respective area under the ROC curves (AUCs). The gray dotted line represents the ROC purely based on chance.

SVM, mGOASVM [48] uses a multi-label SVM classifier, and the mPLR-Loc uses a multi-label penalized logistic regression classifier incorporated with the proposed adaptive decision scheme.

As shown in Table 2, mPLR-Loc performs significantly better than Virus-mPLoc and iLoc-Virus. Both the *OLA* and *OAA* of mPLR-Loc are more than 15% (absolute) higher than iLoc-Virus. They also perform significantly better than KNN-SVM in terms of *OLA*. When comparing with mGOASVM, although the *OLA* of mPLR-Loc is slightly smaller than that of mGOASVM, the *OAA* of mPLR-Loc is 2% (absolute) higher than that of mGOASVM. In terms of *Accuracy*, *Precision*, *F1*, and *HL*, mPLR-Loc performs better than mGOASVM. In terms of *Recall*, mGOASVM performs the best among all of the predictors. This is understandable because according to the analysis in the "Effect of adaptive decisions on mPLR-Loc" subsection, the *Recall* decreases when θ increases. The results suggest that the mPLR-Loc performs better than the state-of-the-art classifiers. The individual locative accuracies of mPLR-Loc are remarkably higher than those of Virus-mPLoc, iLoc-Virus, and KNN-SVM and are comparable to those of mGOASVM.

Similar conclusions can be drawn from Table 3, where the superiority of mPLR-Loc over Plant-mPLoc, iLoc-Plant, and mGOASVM is more evident compared with that in Table 2.

Moreover, the *P* values [72] between the *OAA* of mPLR-Loc and mGOASVM on the virus and plant datasets are 1.1750×10^{-4} and 7.262×10^{-7}, respectively, suggesting that the performance of mPLR-Loc is significantly better than that of mGOASVM on both datasets.

To assess the prediction performance of mPLR-Loc at different decision thresholds, receiver operating characteristic (ROC) curves were used. Note that ROC curves are applicable to binary classification systems only. Because our subcellular localization problems are multi-label and multi-class, ROC curves cannot be directly applied. To tackle this problem, we adopted the one-versus-rest strategy to generate an ROC curve for each subcellular location and then averaged the ROC curves as the final output. Fig. 6A and B show the ROC curves of mPLR-Loc and mGOASVM for the virus dataset and plant dataset, respectively.[3] As can be seen from Fig. 6A, the area under the curve (AUC) of mPLR-Loc is larger than that of mGOASVM. Specifically, the AUC for mPLR-Loc is 0.986, whereas that for mGOASVM is 0.963, suggesting that on average mPLR-Loc performs better than mGOASVM. Although the ROC curves for Virus-mPLoc, KNN-SVM, and iLoc-Virus cannot be shown here due to the unavailability of prediction scores, by inferring from other performance measurements of these predictors, we can optimistically expect that the AUC of mPLR-Loc will be much larger than that of these predictors. Similar conclusions can be drawn from Fig. 6B for the plant dataset except that the improvement of mPLR-Loc over mGOASVM is not so significant as compared with the virus dataset. Specifically, the AUC for mPLR-Loc is 0.980, whereas that for mGOASVM is 0.976.

Prediction of novel proteins

To further demonstrate the effectiveness of mPLR-Loc, a novel and independent plant dataset was created to compare mPLR-Loc with state-of-the-art multi-label predictors using independent tests. To ensure that the test proteins are really novel to mPLR-Loc, the registration dates of these proteins in Swiss–Prot should be later than that of the training proteins. It is also important to ensure that none of these novel proteins appears in the GOA database used by mPLR-Loc. Because the plant dataset used for training the predictors was created on 29 April 2008 and the GOA database used by mPLR-Loc was released on 8 March 2011, we selected the proteins that were added to Swiss–Prot between 8 March 2011 and 9 July 2014 according to the strict criteria specified in Ref. [44]. In other words, this new dataset contains the latest novel proteins and has never been used by other researchers and in other studies. Specifically, this new plant dataset contains 564 plant proteins, of which 472 belong to one subcellular location, 85 belong to two locations, 6 belong to three locations, 1 belongs to four locations, and none belongs to five or more locations. This means that the number of locative proteins [46] is $(472 \times 1 + 85 \times 2 + 6 \times 3 + 1 \times 4 = 664)$. These locative proteins are distributed in 12 subcellular locations, which are detailed as follows: 36 in cell membrane, 7 in cell wall, 148 chloroplast, 146 in cytoplasm, 38 in endoplasmic reticulum, 18 in extracellular, 23 in Golgi apparatus, 63 mitochondrion, 144 in nucleus, 14 in peroxisome, 6 in plastid, and 21 in vacuole. Fig. 7A shows the breakdown of this novel dataset. As can be seen, the majority (76%) of plant proteins are located in chloroplast, cytoplasm, mitochondrion, and nucleus, whereas proteins in the other 8 subcellular locations

[3] Note that we cannot draw the ROC curves for other predictors because we cannot obtain their prediction scores to calculate the false positive rates and true positive rates at different operating points.

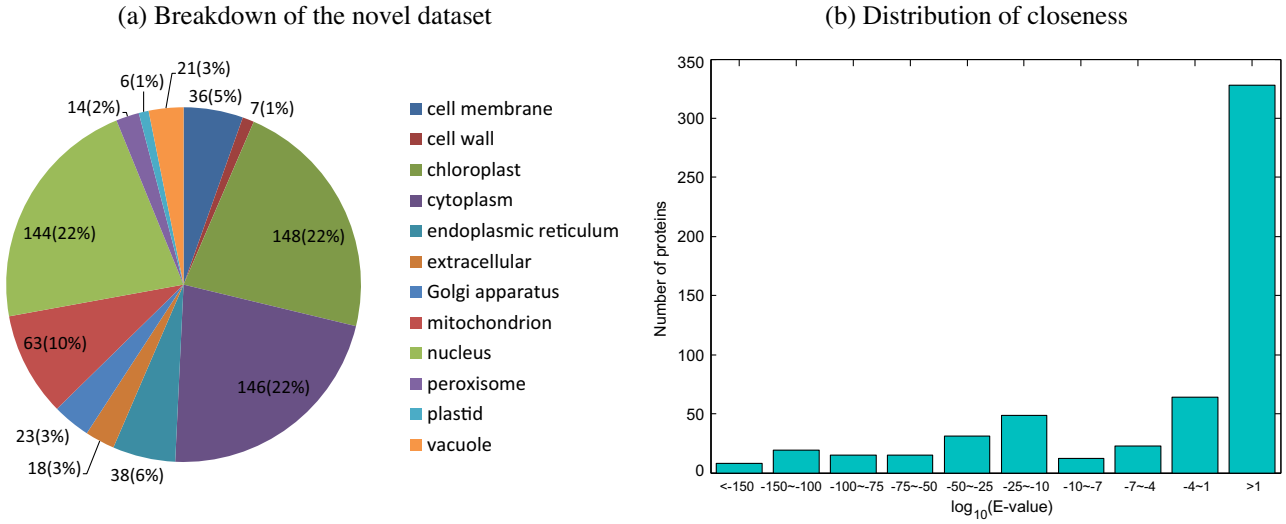

Fig.7. Information about the novel dataset: (A) breakdown of the novel plant dataset; (B) distribution of the closeness (based on E-values of BLAST) between the novel plant dataset and the training plant dataset.

totally account for less than 24%. The novel dataset is downloadable from the mPLR-Loc web server. For unbiased performance evaluation, the sequence similarity of this novel dataset was cut off at 25%.

Fig. 7B shows the distribution of the logarithm of E-values of the test proteins, which were obtained by using the training proteins as the repository and the test proteins as the query proteins in the BLAST search. If we use a common criterion that homologous proteins should have E-values less than 10^{-4}, then 172 of 564 (or 30.5%) test proteins are homologs of the training proteins. Note that this does not mean that BLAST can predict all 172 of these test proteins correctly. Actually, using BLAST's homology transfers (based on the CC field of the homologous proteins) achieves significantly lower accuracy than the homology rate, as validated in our previous study [29]. As shown in Table 4, the prediction accuracy of mPLR-Loc on this test set is significantly higher than this homology rate. This suggests that the information available in the GOA database plays a very important role in the prediction process.

Table 4 compares the performance of mPLR-Loc against several state-of-the-art multi-label plant predictors on the new plant dataset. All of the predictors use the 978 proteins of the plant dataset (see Fig. 3) for training the classifier and perform independent tests on the new 564 proteins. As can be seen, mPLR-Loc performs significantly better than Plant-mPLoc and iLoc-Plant in terms of all performance metrics. Surprisingly, when comparing with mGOASVM, mPLR-Loc also performs better than mGOASVM in terms of all performance metrics. In particular, the *OAA* of mPLR-Loc is nearly 3% better than that of mGOASVM. This suggests that mPLR-Loc performs robustly better than existing state-of-the-art predictors.

Table 4
Comparing mPLR-Loc with state-of-the-art multi-label plant predictors based on independent tests using the new plant dataset.

Label	Subcellular location	Independent test locative accuracy			
		Plant-mPLoc [45]	iLoc-Plant [47]	mGOASVM [49]	mPLR-Loc
1	Cell membrane	15/36 = 0.417	1/36 = 0.028	13/36 = 0.361	21/36 = 0.583
2	Cell wall	0/7 = 0	0/7 = 0	0/7 = 0	1/7 = 0.143
3	Chloroplast	91/148 = 0.615	77/148 = 0.520	127/148 = 0.858	126/148 = 0.851
4	Cytoplasm	20/146 = 0.137	35/146 = 0.240	31/146 = 0.212	41/146 = 0.281
5	Endoplasmic reticulum	4/38 = 0.105	5/38 = 0.132	16/38 = 0.421	13/38 = 0.342
6	Extracellular	0/18 = 0	0/18 = 0	3/18 = 0.167	3/18 = 0.167
7	Golgi apparatus	6/23 = 0.261	1/23 = 0.044	3/23 = 0.130	3/23 = 0.130
8	Mitochondrion	27/63 = 0.429	14/63 = 0.222	28/63 = 0.444	28/63 = 0.444
9	Nucleus	105/144 = 0.729	68/144 = 0.472	67/144 = 0.465	74/144 = 0.514
10	Peroxisome	6/14 = 0.429	0/14 = 0	8/14 = 0.571	9/14 = 0.643
11	Plastid	0/6 = 0	1/6 = 0.167	0/6 = 0	0/6 = 0
12	Vacuole	5/21 = 0.238	11/21 = 0.524	11/21 = 0.524	11/21 = 0.524
Overall actual accuracy (*OAA*)		165/564 = 0.293	161/564 = 0.286	238/564 = 0.422	254/564 = **0.450**
Overall locative accuracy (*OLA*)		279/664 = 0.420	213/664 = 0.321	307/664 = 0.462	330/664 = **0.497**
Accuracy		0.381	0.328	0.475	**0.509**
Precision		0.414	0.359	0.512	**0.552**
Recall		0.445	0.339	0.492	**0.527**
F1		0.413	0.342	0.493	**0.529**
HL		0.124	0.123	0.097	**0.090**

Note. The performance for Plant-mPLoc [45] and iLoc-Plant [47] are calculated based on their corresponding web servers. See Eqs. (12)–(18) for the definitions of the performance measures.

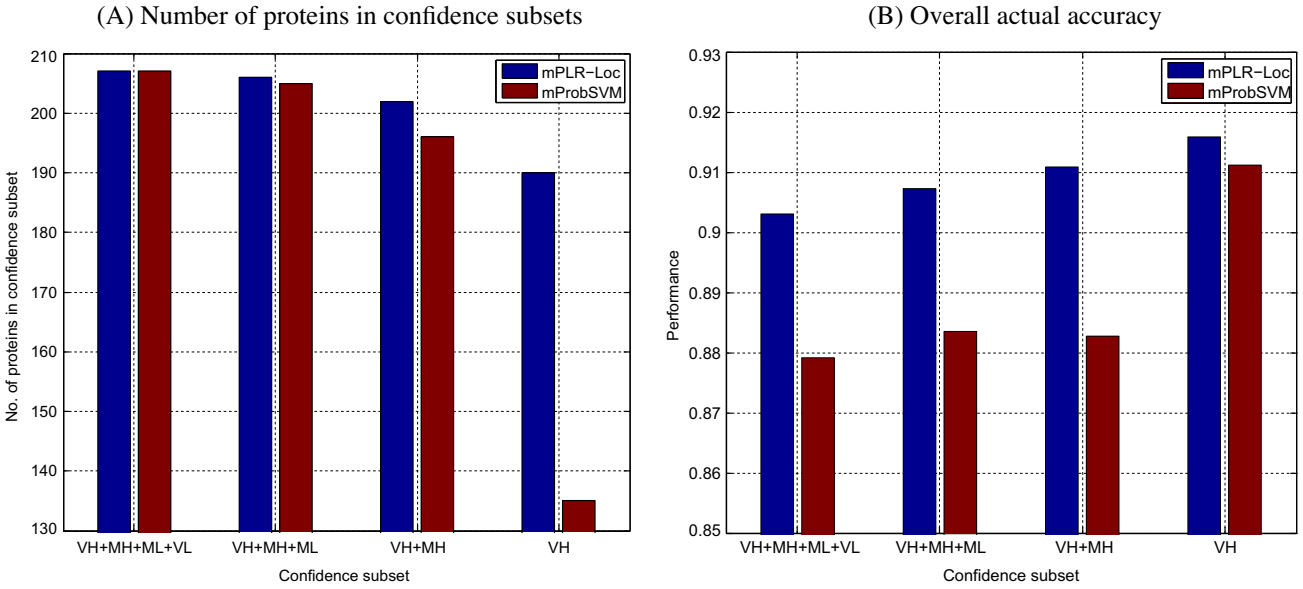

Fig.8. Comparing mPLR-Loc with multi-label probabilistic SVMs (mProbSVM) [72] on the number of proteins in confidence subsets (A) and the performance on different confidence subsets (B). Confidence subset: the union of different protein subgroups: very high (*VH*), median high (*MH*), median low (*ML*), and very low (*VL*). See text for the details of confidence subsets.

Biological significance of using GO term frequency features

The GOA database is constructed by various biological research communities around the world (http://geneontology.org/page/go-consortium-contributors-list). It is possible that some annotations for the same proteins are done by different GO consortium contributing groups around the world. In this case, it is likely that the annotations of the same biological process, molecular function, or cellular component for the same protein by different research groups are different or even contradictory, which may result in the inaccuracy or inconsistency of the GO annotations. In other words, there are inevitably some noisy data or outliers in the GOA database. In particular, when the traditional 1–0 value method [43,44] was used to extract the GO features, the influence of those "noise-contained"' GO terms will be emphasized because of their presence for a query protein. These noisy data and outliers may negatively affect the performance of machine-learning-based approaches.

For this concern, first we need to admit that these noisy data and outliers are likely to exist in the GOA database; unfortunately, it is not easy to distinguish them from correct GO annotations. Only wet-lab experimentalists can rely on their biological knowledge to discriminate these noisy data or outliers and remove them from the database. However, by using the term frequency information of GO features, we can somewhat suppress the influence of these noisy data and outliers. The reasons are elaborated below.

In this article, term frequency information was used to emphasize those annotations that are confirmed by different research groups. From our observations, the same GO term for the same protein may appear more than once in the GOA database, but possibly with different evidence codes or from different contributing databases. This means that these kinds of GO terms are validated several times by different research groups and by different ways, leading to the same annotation results. On the contrary, if different research groups annotate the same protein by different GO terms whose annotations are contradictory with each other, the frequencies of these GO terms for this protein should be low. In other words, the higher the frequency a GO term appears, the more times this GO annotation is confirmed by different research groups and the more credible the annotation of this GO term. By using the term frequency in our feature vectors, we can enhance the influence of those GO terms that appear more frequently; in other words, we can enhance the influence of those GO terms whose annotations are consistent with each other. Meanwhile, we can indirectly suppress the influence of those GO terms that appear less frequently; in other words, we can suppress the influence of those GO terms whose annotations are contradictory to each other.

The advantages of using the GO term frequency features is evident by the superior results shown in our previous studies [48,54], where using GO term frequency information performs significantly better than using the 1–0 value.[4]

Analysis of confidence levels

Classifiers that can produce posterior probabilities of classes are useful for many practical applications. The posterior probabilities indicate the confidence in assigning an instance to a particular class. In multi-class classification, assigning an unknown instance to the class with maximal posterior probability is a typical application of the probabilistic output scores produced by these classifiers.

Probabilistic scores are particularly useful in multi-label classification, where an instance may belong to more than one class. Standard SVMs, kNNs, or other conventional classifiers can produce only uncalibrated and non-probabilistic output scores. Unlike multi-class classification, decisions in multi-label classification cannot be based solely on the maximal output scores, which makes standard SVMs less effective. One possible way to solve this problem is to convert the SVM output scores into calibrated posterior probabilities [73]. However, the results in this subsection show that it is inferior to mPLR-Loc proposed in this article.

By using a penalized logistic regression classifier, the proposed mPLR-Loc predictor possesses intrinsic properties of generating probabilistic output scores. These probabilistic scores can be directly interpreted as confidence levels (i.e., the confidence in assigning an unknown instance to a certain class). The larger the score, the higher the confidence level. For example, in Fig. A11 of

[4] Note that because we have shown the advantages of using GO term frequency features over the 1–0 value method in our previous studies, to avoid repetition, we do not implement similar experiments in this article.

Appendix B, the posterior probabilities for the 12 locations of a query protein are [0,0,0,0.87,0,0,0,0,0.96,0,0,0]. According to the decision scheme in Eq. (10), the query protein will be assigned to the 4-th and 9-th classes, namely "cytoplasm" and "nucleus." Moreover, because the score in position 9 is larger than that in position 4, this protein is more likely to be located in "nucleus" than in "cytoplasm."

Based on this observation, we propose using the maximum score produced by the logistic regressions as the overall confidence level of a decision. Specifically, given a query protein $\mathbb{Q}_i$, the posterior score $s_m(\mathbb{Q}_i)$ for the m-th ($m \in \{1, \ldots, M\}$) location is determined by Eq. (9). Then, we find the maximum score among all of the locations:

$$s_{\max}(\mathbb{Q}_i) = \max_{m=1}^{M} s_m(\mathbb{Q}_i). \tag{20}$$

Then, we divide the confidence into four levels:

$$\mathbb{C} = \begin{cases} \text{very high } (VH) & \text{if } 0.8 \leqslant s_{\max}(\mathbb{Q}_i) \leqslant 1.0, \\ \text{median high } (MH) & \text{if } 0.5 \leqslant s_{\max}(\mathbb{Q}_i) < 0.8, \\ \text{median low } (ML) & \text{if } 0.2 \leqslant s_{\max}(\mathbb{Q}_i) < 0.5, \\ \text{very low } (VL) & \text{if } 0 \leqslant s_{\max}(\mathbb{Q}_i) < 0.2. \end{cases} \tag{21}$$

For ease of reference, "very high," "median high," "median low," and "very low" are abbreviated as *VH*, *MH*, *ML*, and *VL*, respectively. In other words, if $s_{\max}(\mathbb{Q}_i) \geqslant 0.8$, then the confidence of the decision is very high; on the contrary, if $s_{\max}(\mathbb{Q}_i) < 0.2$, then the confidence is very low, meaning that the decision may be wrong. Based on Eq. (21), the proteins in a dataset can be divided into four subgroups: $\mathbb{G}_{VH}$, $\mathbb{G}_{MH}$, $\mathbb{G}_{ML}$, and $\mathbb{G}_{VL}$. For example, $s_{\max}$ of proteins in $\mathbb{G}_{VL}$ are all less than 0.2.

To demonstrate the effectiveness of the confidence levels and the superiority of mPLR-Loc over other probabilistic classifiers, we have compared mPLR-Loc with a multi-label probabilistic SVM classifier [73] (mProbSVM for short) using different confidence subsets derived from the virus dataset. Here, a confidence subset is the union of protein subgroups whose proteins receive confidence scores higher than or equal to a specific confidence level.[5] For example, *VH* + *MH* in the *x*-axis label of Fig. 8A represents the union of $\mathbb{G}_{VH}$ and $\mathbb{G}_{MH}$, meaning that the proteins in this subset have confidence scores larger than or equal to 0.5.

According to Ref. [73], SVM scores can be converted to probabilistic scores through a sigmoid function. This idea can be extended to multi-label multi-class classification as follows. Given a query protein $\mathbb{Q}_i$, the calibrated probabilistic score $p_m^{svm}(\mathbb{Q}_i)$ for the m-th location can be defined as

$$p_m^{svm}(\mathbb{Q}_i) = \frac{1}{1 + e^{(A \cdot s_m^{svm}(\mathbb{Q}_i) + B)}}, \tag{22}$$

where A and B can be trained via cross-validation and $s_m^{svm}(\mathbb{Q}_i)$ is the uncalibrated SVM score of the query protein $\mathbb{Q}_i$ for the m-th location.

Fig. 8A shows the numbers of proteins in each of these confidence subsets produced by mPLR-Loc and mProbSVM. The excessively small number of proteins in the *VH* subset produced by mProbSVM implies that mProbSVM is not very confident in classifying the majority of the proteins in the dataset. Fig. 8A also shows that for all of the confidence subsets, mPLR-Loc can always find a larger number of proteins than mProbSVM. This phenomenon, together with the results in Fig. 8B, suggests that mPLR-Loc not only performs better than mProbSVM in terms of classification accuracy but also classifies more proteins at a higher confidence level than mProbSVM. Although mProbSVM achieves a performance comparable to that of mPLR-Loc in the *VH* subset, the number of proteins in this subset for mProbSVM (135 of 207) is much smaller than that for mPLR-Loc (190 of 207). This means that even for this stringent condition, mPLR-Loc is still better than mProbSVM in terms of classification accuracy and classification confidence.

[5] It is logically acceptable that if a decision with lower confidence is trustworthy, then those decisions with higher confidence should also be trustworthy.

Conclusions

This article has proposed an efficient multi-label predictor, namely mPLR-Loc, which is based on multi-label penalized logistic regression incorporated with an adaptive decision scheme to predict subcellular localization of both single- and multi-label proteins. Given a query protein, a GO-based feature vector is constructed by exploiting the information in the GOA database. The GO vector is presented to one-versus-rest penalized logistic regression classifiers to obtain M scores, where M is the number of classes with a single label. The scores are then compared with an adaptive decision threshold that is proportional to the maximum of the M scores for predicting the number of labels as well as the class label(s) of the query protein.

Comparing with existing multi-label predictors, mPLR-Loc has the following advantages: (i) it uses a multi-label penalized logistic regression classifier equipped with an adaptive decision strategy that can tackle multi-label problems effectively; (ii) not only can it rapidly and accurately provide prediction decisions, it also is able to give probabilistic confidence scores for the prediction decisions; and (iii) it adopts a successive search strategy to incorporate useful homologous information for constructing discriminative feature vectors.

Experimental results on two recent benchmark datasets demonstrate that mPLR-Loc performs significantly better than existing state-of-the-art multi-label predictors specializing in virus or plant proteins. For readers' convenience, mPLR-Loc is available online (http://bioinfo.eie.polyu.edu.hk/mPLRLocServer).

Acknowledgment

This work was supported in part by The Hong Kong Polytechnic University, Hong Kong SAR, China grants G-YL78 and G-YN18.

Appendix A. Derivatives for penalized logistic regression

In the "Single-label penalized logistic regression" subsection, to minimize $E(\boldsymbol{\beta})$, we may use the Newton–Raphson algorithm to obtain Eq. (4), where the first and second derivatives of $E(\boldsymbol{\beta})$ are as follows:

$$\frac{\partial E(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\sum_{i=1}^{N} \mathbf{x}_i (y_i - p(\mathbf{x}_i; \boldsymbol{\beta})) + \rho \boldsymbol{\beta} = -\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{p}) + \rho \boldsymbol{\beta} \tag{A23}$$

and

$$\begin{aligned} \frac{\partial^2 E(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} &= \sum_{i=1}^{N} \left[\frac{\partial \mathbf{x}_i p(\mathbf{x}_i; \boldsymbol{\beta})}{\partial \boldsymbol{\beta}^{\mathrm{T}}} \right] + \rho \mathbf{I} \\ &= \sum_{i=1}^{N} \mathbf{x}_i \left[\frac{\partial}{\partial \boldsymbol{\beta}^{\mathrm{T}}} \left(\frac{e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}} \right) \right] + \rho \mathbf{I} \\ &= \sum_{i=1}^{N} \mathbf{x}_i \left[\frac{e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i} \mathbf{x}_i^{\mathrm{T}} (1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}) - e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i} \mathbf{x}_i^{\mathrm{T}}}{(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i})^2} \right] + \rho \mathbf{I} \\ &= \sum_{i=1}^{N} \mathbf{x}_i \left[\frac{\mathbf{x}_i^{\mathrm{T}} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}} \cdot \frac{1}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}} \right] + \rho \mathbf{I} \\ &= \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} p(\mathbf{x}_i; \boldsymbol{\beta})(1 - p(\mathbf{x}_i; \boldsymbol{\beta})) + \rho \mathbf{I} = \mathbf{X}^{\mathrm{T}} \mathbf{W} \mathbf{X} + \rho \mathbf{I}. \end{aligned} \tag{A24}$$

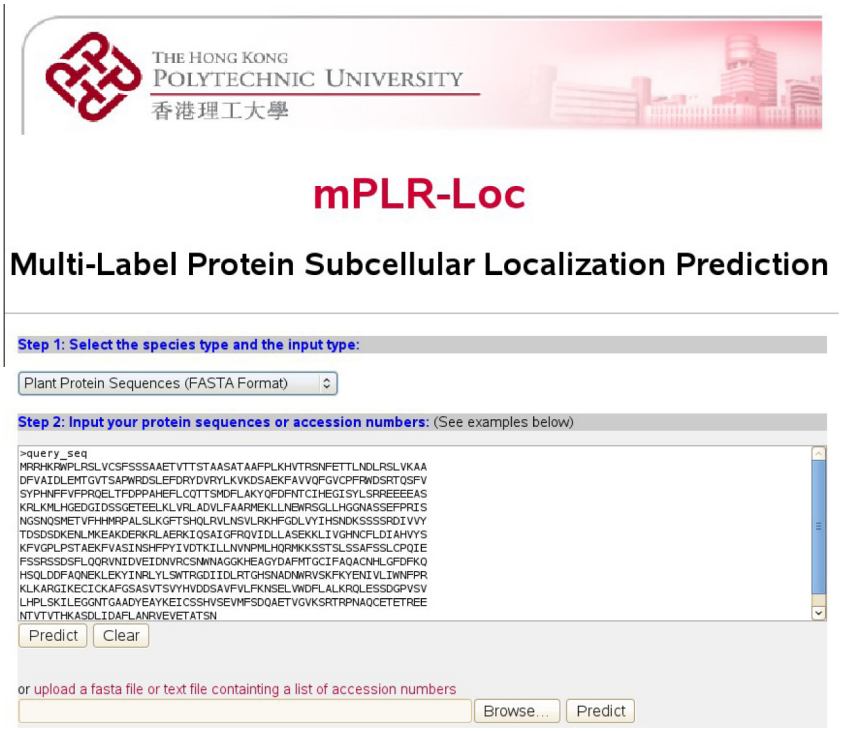

Fig.A9. An example of using a plant protein sequence in the Fasta format as input to the mPLR-Loc server.

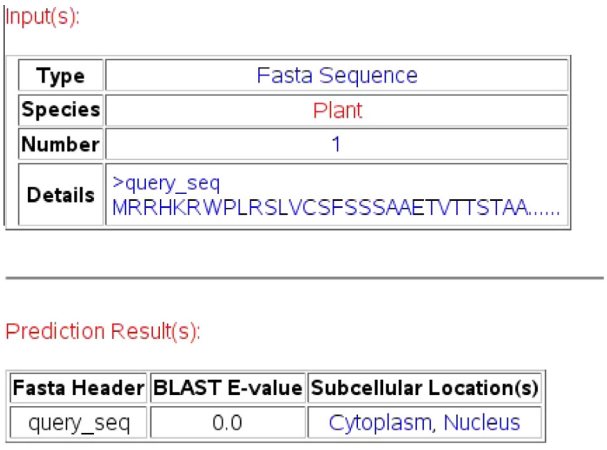

Fig.A10. Prediction results of the mPLR-Loc server for the plant protein sequence input in Fig. A9.

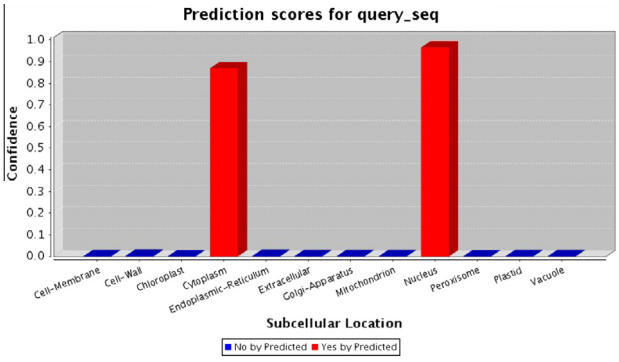

Fig.A11. Confidence scores of the mPLR-Loc server for the plant protein sequence input in Fig. A9.

In Eqs. (A23) and (A24), **y** and **p** are N-dim vectors whose elements are $\{y_i\}_{i=1}^{N}$ and $\{p(\mathbf{x}_i;\boldsymbol{\beta})\}_{i=1}^{N}$, respectively, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^{\mathrm{T}}$, $\mathbf{W}$ is a diagonal matrix whose i-th diagonal element is $p(\mathbf{x}_i;\boldsymbol{\beta})[1 - p(\mathbf{x}_i;\boldsymbol{\beta})]$, $i = 1, 2, \ldots, N$.

Appendix B. mPLR-Loc web server

For readers' convenience, a web server for mPLR-Loc has been developed. The mPLR-Loc server can deal with two species (i.e., virus and plant) and two different input types (i.e., protein sequences in Fasta format and protein accession numbers in UniProtKB format). After going to the home page of the mPLR-Loc server, select a combination of species type and input type. Then, input the query protein sequences or accession numbers or upload a file containing a list of accession numbers or protein sequences. For example, Fig. A9 shows the screenshot that uses a plant protein sequence in Fasta format as input. After clicking the "Predict" button and waiting approximately 13 s, the prediction results as shown in Fig. A10 and the probabilistic scores as shown in Fig. A11 will be produced. The prediction result in Fig. A10 includes the Fasta header, BLAST E-value, and predicted subcellular location(s). Fig. A11 shows the confidence in the predicted subcellular location(s). In this figure, mPLR-Loc predicts the query sequence as "cytoplasm" and "nucleus" with confidence scores greater than 0.8 and 0.9, respectively.

References

[1] K.C. Chou, Y.D. Cai, Predicting protein localization in budding yeast, Bioinformatics 21 (2005) 944–950.
[2] G. Lubec, L. Afjehi-Sadat, J.W. Yang, J.P. John, Searching for hypothetical proteins: theory and practice based upon original data and literature, Prog. Neurobiol. 77 (2005) 90–127.
[3] M.D. Kaytor, S.T. Warren, Aberrant protein deposition and neurological disease, J. Biol. Chem. 274 (1999) 37507–37510.
[4] M.C. Hung, W. Link, Protein localization in disease and therapy, J. Cell Sci. 124 (2011) 3381–3392.

[5] V. Krutovskikh, G. Mazzoleni, N. Mironov, Y. Omori, A.M. Aguelon, M. Mesnil, F. Berger, C. Partensky, H. Yamasaki, Altered homologous and heterologous gap–junctional intercellular communication in primary human liver tumors associated with aberrant protein localization but not gene mutation of connexin 32, Int. J. Cancer 56 (1994) 87–94.

[6] Y. Chen, C.F. Chen, D.J. Riley, D.C. Allred, P.L. Chen, D.V. Hoff, C.K. Osborne, W.H. Lee, Aberrant subcellular localization of BRCA1 in breast cancer, Science 270 (1995) 789–791.

[7] J.B. Campbell, J. Crocker, P.M. Shenoi, S-100 protein localization in minor salivary gland tumours: an aid to diagnosis, J. Laryngol. Otol. 102 (1988) 905–908.

[8] X. Lee, J.C.J. Keith, N. Stumm, I. Moutsatsos, J.M. McCoy, C.P. Crum, D. Genest, D. Chin, C. Ehrenfels, R. Pijnenborg, F.A.V. Assche, S. Mi, Downregulation of placental syncytin expression and abnormal protein localization in pre-eclampsia, Placenta 22 (2001) 808–812.

[9] A. Hayama, T. Rai, S. Sasaki, S. Uchida, Molecular mechanisms of Bartter syndrome caused by mutations in the BSND gene, Histochem. Cell Biol. 119 (2003) 485–493.

[10] O. Emanuelsson, H. Nielsen, S. Brunak, G. von Heijne, Predicting subcellular localization of proteins based on their N-terminal amino acid sequence, J. Mol. Biol. 300 (2000) 1005–1016.

[11] H. Nielsen, J. Engelbrecht, S. Brunak, G. von Heijne, A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites, Int. J. Neural Syst. 8 (1997) 581–599.

[12] K. Nakai, M. Kanehisa, Expert system for predicting protein localization sites in gram-negative bacteria, Proteins 11 (1991) 95–110.

[13] Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, R. Eisner, Predicting subcellular localization of proteins using machine-learned classifiers, Bioinformatics 20 (2004) 547–556.

[14] M.W. Mak, J. Guo, S.Y. Kung, PairProSVM: protein subcellular localization based on local pairwise profile alignment and SVM, IEEE/ACM Trans. Comput. Biol. Bioinform. 5 (2008) 416–422.

[15] R. Mott, J. Schultz, P. Bork, C. Ponting, Predicting protein cellular localization using a domain projection method, Genome Res. 12 (2002) 1168–1174.

[16] S.W. Zhang, Y.L. Zhang, H.F. Yang, C.H. Zhao, Q. Pan, Using the concept of Chou's pseudo amino acid composition to predict protein subcellular localization: an approach by incorporating evolutionary information and von Neumann entropies, Amino Acids 34 (2008) 565–572.

[17] H. Nakashima, K. Nishikawa, Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, J. Mol. Biol. 238 (1994) 54–61.

[18] K.C. Chou, Prediction of protein cellular attributes using pseudo amino acid composition, Proteins 43 (2001) 246–255.

[19] S. Wan, M.W. Mak, S.Y. Kung, R3P-Loc: a compact multi-label predictor using ridge regression and random projection for protein subcellular localization, J. Theor. Biol. 360 (2014) 34–45.

[20] K.C. Chou, Y.D. Cai, Prediction of protein subcellular locations by GO-FunD-PseAA predictor, Biochem. Biophys. Res. Commun. 320 (2004) 1236–1239.

[21] S. Wan, M. W. Mak, S. Y. Kung, Protein subcellular localization prediction based on profile alignment and gene ontology, in: 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP'11), 2011, pp. 1–6.

[22] K.C. Chou, H.B. Shen, Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers, J. Proteome Res. 5 (2006) 1888–1897.

[23] S. Wan, M. W. Mak, S. Y. Kung, GOASVM: Protein subcellular localization prediction based on gene ontology annotation and SVM, in: 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'12), 2012, pp. 2229–2232.

[24] S. Mei, Multi-label multi-kernel transfer learning for human protein subcellular localization, PLoS ONE 7 (6) (2012) e37716.

[25] S. Wan, M. W. Mak, S. Y. Kung, Semantic similarity over gene ontology for multi-label protein subcellular localization, Engineering 5 (2013) 68–72. URL: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=38539.

[26] S.W. Zhang, Y.F. Liu, Y. Yu, T.H. Zhang, X.N. Fan, MSLoc-DT: a new method for predicting the protein subcellular location of multispecies based on decision templates, Anal. Biochem. 449 (2014) 164–171.

[27] S. Wan, M. W. Mak, B. Zhang, Y. Wang, S. Y. Kung, Ensemble random projection for multilabel classification with application to protein subcellular localization, in: 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14), 2014, pp. 5999–6003. doi: 10.1109/ICASSP.2014.6854755.

[28] W.Z. Lin, J.A. Fang, X. Xiao, K.C. Chou, ILoc-Animal: a multi-label learning classifier for predicting subcellular localization of animal proteins, Mol. BioSyst. 9 (2013) 634–644.

[29] S. Wan, M.W. Mak, S.Y. Kung, HybridGO-Loc: mining hybrid features on gene ontology for predicting subcellular localization of multi-location proteins, PLoS ONE 9 (3) (2014) e89545.

[30] R. Nair, B. Rost, Sequence conserved for subcellular localization, Protein Sci. 11 (2002) 2836–2847.

[31] S. Brady, H. Shatkay, EpiLoc: A (working) text-based system for predicting protein subcellular location, in: Pacific Symposium on Biocomputing, 2008, pp. 604–615.

[32] A. Fyshe, Y. Liu, D. Szafron, R. Greiner, P. Lu, Improving subcellular localization prediction using text classification and the gene ontology, Bioinformatics 24 (2008) 2512–2517.

[33] J.C. Mueller, C. Andreoli, H. Prokisch, T. Meitinger, Mechanisms for multiple intracellular localization of human mitochondrial proteins, Mitochondrion 3 (2004) 315–325.

[34] C. Huang, J.Q. Yuan, A multilabel model based on Chou's pseudo-amino acid composition for identifying membrane proteins with both single and multiple functional types, J. Membr. Biol. 246 (2013) 327–334.

[35] A. Clare, R. D. King, Knowledge discovery in multi-label phenotype data, in: Proceedings of the Fifth European Conference on Principles of Data Mining and Knowledge Discovery, 2001, pp. 42–53.

[36] R.E. Schapire, Y. Singer, BoosTexter: a boosting-based system for text categorization, Machine Learn. 39 (2/3) (2000) 135–168.

[37] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, Machine Learn. 2 (73) (2008) 185–214.

[38] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (2004) 1757–1771.

[39] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2009, pp. 254–269.

[40] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: O. Maimon, I. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, second ed., Springer, New York, 2010, pp. 667–685.

[41] D. Hsu, S. M. Kakade, J. Langford, T. Zhang, Multi-label prediction via compressed sensing, in: Advances in Neural Information Processing Systems 22, 2009, pp. 772–780.

[42] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, Int. J. Data Warehousing Mining 3 (2007) 1–13.

[43] H.B. Shen, K.C. Chou, Virus-mPLoc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites, J. Biomol. Struct. Dyn. 26 (2010) 175–186.

[44] K.C. Chou, H.B. Shen, Plant-mPLoc: a top-down strategy to augment the power for predicting plant protein subcellular localization, PLoS ONE 5 (6) (2010) e11335.

[45] X. Xiao, Z.C. Wu, K.C. Chou, ILoc-Virus: a multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites, J. Theor. Biol. 284 (2011) 42–51.

[46] Z.C. Wu, X. Xiao, K.C. Chou, ILoc-Plant: a multi-label classifier for predicting the subcellular localization of plant proteins with both single and multiple sites, Mol. BioSyst. 7 (2011) 3287–3297.

[47] L.Q. Li, Y. Zhang, L.Y. Zou, Y. Zhou, X.Q. Zheng, Prediction of protein subcellular multilocalization based on the general form of Chou's pseudo amino acid composition, Protein Pept. Lett. 19 (2012) 375–387.

[48] S. Wan, M.W. Mak, S.Y. Kung, MGOASVM: multi-label protein subcellular localization based on gene ontology and support vector machines, BMC Bioinformatics 13 (2012) 290.

[49] J. He, H. Gu, W. Liu, Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites, PLoS ONE 7 (6) (2011) e37155.

[50] S. Wan, M. W. Mak, B. Zhang, Y. Wang, S. Y. Kung, An ensemble classifier with random projection for predicting multi-label protein subcellular localization, in: 2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2013, pp. 35–42. doi: http://dx.doi.org/10.1109/BIBM.2013.6732715.

[51] L.Q. Li, Y. Zhang, L.Y. Zou, C.Q. Li, B. Yu, X.Q. Zheng, Y. Zhou, An ensemble classifier for eukaryotic protein subcellular location prediction using Gene Ontology categories and amino acid hydrophobicity, PLoS ONE 7 (1) (2012) e31057.

[52] H.B. Shen, K.C. Chou, Virus-PLoc: a fusion classifier for predicting the subcellular localization of viral proteins within host and virus-infected cells, Biopolymers 85 (2006) 233–240.

[53] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Ntale, C. O'Donovan, N. Redaschi, L.S. Yeh, UniProt: The Universal Protein knowledgebase, Nucleic Acids Res. 32 (2004) D115–D119.

[54] S. Wan, M.W. Mak, S.Y. Kung, GOASVM: a subcellular location predictor by incorporating term-frequency gene ontology into the general form of Chou's pseudo-amino acid composition, J. Theor. Biol. 323 (2013) 40–48.

[55] Z. Lu, L. Hunter, GO molecular function terms are predictive of subcellular localization, in: Proceedings of Pacific Symposium of Biocomputing (PSB'05), 2005, pp. 151–161.

[56] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.

[57] S. Briesemeister, T. Blum, S. Brady, Y. Lam, O. Kohlbacher, H. Shatkay, SherLoc2: a high accuracy hybrid method for predicting subcellular localization of proteins, J. Proteome Res. 8 (2009) 5363–5366.

[58] K.C. Chou, Some remarks on predicting multi-label attributes in molecular biosystems, Mol. BioSyst. 9 (2013) 1092–1100.

[59] X. Wang, G.Z. Li, A multi-label predictor for identifying the subcellular locations of singleplex and multiplex eukaryotic proteins, PLoS ONE 7 (5) (2012) e36317.

[60] K.C. Chou, H.B. Shen, Cell-PLoc: a package of web-servers for predicting subcellular localization of proteins in various organisms, Nat. Protoc. 3 (2008) 153–162.

[61] W.L. Huang, C.W. Tung, S.W. Ho, S.F. Hwang, S.Y. Ho, ProLoc-GO: utilizing informative Gene Ontology terms for sequence-based prediction of protein subcellular localization, BMC Bioinformatics 9 (2008) 80.

[62] K. Nakai, Protein sorting signals and prediction of subcellular localization, Adv. Protein Chem. 54 (2000) 277–344.
[63] D. Barrel, E. Dimmer, R.P. Huntley, D. Binns, C. O'Donovan, R. Apweiler, The GOA database in 2009—an integrated Gene Ontology Annotation resource, Nucleic Acids Res. 37 (2009) D396–D403.
[64] S. Wan, M. W. Mak, S. Y. Kung, Adaptive thresholding for multi-label SVM classification with application to protein subcellular localization prediction, in: 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13), 2013, pp. 3547–3551.
[65] D.W. Hosmer, S. Lemeshow, Applied Logistic Regression, second ed., John Wiley, New York, 2000.
[66] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, J. Comput. Graph. Stat. (2001) 1081–1088
[67] J. Zhu, Classification of gene microarrays by penalized logistic regression, Biostatistics 5 (2004) 427–443.
[68] S.K. Shevade, S.S. Keerthi, A simple and efficient algorithm for gene selection using sparse logistic regression, Bioinformatics 19 (2003) 2246–2253.
[69] K. Dembczynski, W. Waegeman, W. Cheng, E. Hullermeier, On label dependence and loss minimization in multi-label classification, Machine Learn. 88 (1–2) (2012) 5–45.
[70] W. Gao, Z. H. Zhou, On the consistency of multi-label learning, in: Proceedings of the 24th Annual Conference on Learning Theory, 2011, pp. 341–358.
[71] T. Hastie, R. Tibshirani, J. Friedman, The Element of Statistical Learning, Springer-Verlag, New York, 2001.
[72] L. Gillick, S. J. Cox, Some statistical issues in the comparison of speech recognition algorithms, in: 1989 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'89), 1989, pp. 532–535.
[73] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Adv. Large Margin Classifiers 10 (3) (1999) 61–74.