BioSystems 113 (2013) 50–57

Contents lists available at SciVerse ScienceDirect

BioSystems

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

Using radial basis function on the general form of Chou's pseudo amino acid composition and PSSM to predict subcellular locations of proteins with both single and multiple sites

Chao Huang*, Jingqi Yuan

Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 11 November 2012
Received in revised form 10 April 2013
Accepted 24 April 2013

Keywords:
Pseudo amino acid composition
Multi-label
Subcellular localization
Neural network

ABSTRACT

Prediction of protein subcellular location is a meaningful task which attracted much attention in recent years. A lot of protein subcellular location predictors which can only deal with the single-location proteins were developed. However, some proteins may belong to two or even more subcellular locations. It is important to develop predictors which will be able to deal with multiplex proteins, because these proteins have extremely useful implication in both basic biological research and drug discovery. Considering the circumstance that the number of methods dealing with multiplex proteins is limited, it is meaningful to explore some new methods which can predict subcellular location of proteins with both single and multiple sites. Different methods of feature extraction and different models of predict algorithms using on different benchmark datasets may receive some general results. In this paper, two different feature extraction methods and two different models of neural networks were performed on three benchmark datasets of different kinds of proteins, i.e. datasets constructed specially for Gram-positive bacterial proteins, plant proteins and virus proteins. These benchmark datasets have different number of location sites. The application result shows that RBF neural network has apparently superiorities against BP neural network on these datasets no matter which type of feature extraction is chosen.

1. Introduction

The knowledge of protein subcellular locations is very important because the function of a protein and its role in a cell are closely correlated with its subcellular location (Ehrlich et al., 2002; Glory and Murphy, 2007). It is also very crucial and useful during the process of drug development. For example, bacteria play an important role in both basic research and drug design, owing to the fact that they are the workhorses for the fields of molecular biology, genetics and biochemistry (Xiao et al., 2011b).

With the fast development of large-scale genome, large number of protein sequences is continuously created. Depending on multifarious biochemical experiments to receive the information of protein subcellular localization is unpractical, because these experiments are both resource-intensive and time-consuming. Actually, a series of classifiers or predictors have been developed to identify protein subcellular localization (Cai et al., 2010; Chou and Shen, 2006a,b; Emanuelsson et al., 2000; Hu et al., 2012a,b; Jin et al., 2008; Lin et al., 2008, 2009; Luo, 2012; Matsuda et al., 2005; Niu et al., 2008; Pierleoni et al., 2006; Shen and Chou, 2007a,b; Su et al., 2007; Tejedor-Estrada et al., 2012; Wang et al., 2012; Zhou and Doctor, 2003). All of them can only deal with single-location protein sequence. But the phenomenon that proteins simultaneously exist at or move between different subcellular localizations has been broadly discovered in various kinds of proteins, so it is interesting and meaningful to focus on the development of multi-location protein classifiers.

There are several studies on multi-label prediction dealing with protein sequences. Gpos-mPloc (Shen and Chou, 2009) is a software to predict protein subcellular localization of Gram-positive bacteria. Meanwhile, it can also deal with multiple-location proteins as well. Plant-mPloc (Chou and Shen, 2010) is a top-down strategy which serves to predict single or multiple subcellular localization of plant protein subcellular. Virus-mPloc (Shen and Chou, 2010) is a fusion classifier which was developed by combining the functional domain information, gene ontology information and sequential evolutionary information to predict single or multiple protein subcellular localization of virus. In order to improve the prediction quality, three revised editions based on these predictors were developed. They are called: "iLoc-Gpos (Wu et al.,

Abbreviations: PSSM, position-specific scoring matrix; PseACC, pseudo amino acid composition; BP, back-propagation; RBF, radial basis function.

* Corresponding author. Tel.: +86 15821254988.
E-mail address: huangchao_sjtu@aliyun.com (C. Huang).

0303-2647/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.biosystems.2013.04.005

2012)", "iLoc-Plant (Wu et al., 2011)", and iLoc-Virus (Xiao et al., 2011a).

These studies were all using GO (Gene Ontology) database method which is based on biological process, cellular component and molecular function (Ashburner et al., 2000; Chou and Shen, 2010). The GO approach is not an ab initio approach but a higher-level approach. In current study, to the ab initio purpose, we try to introduce new applications of algorithms which are based on the features extracted directly from the amino-acid sequence or evolution information deriving from its primitive database without any prior knowledge. It is also very interesting to study performances of algorithms this paper used by adapting the GO approach. Considering that there are still some difficulties in the method of feature extraction, it will be our research topic for future study. To provide the readership with the updated view about the GO approach, a profound and penetrative discussion has been given in Section VI of a recent comprehensive review (Chou, 2013) and Section 3 of a recent paper (Lin et al., 2013).

Considering the fact that using BP neural network is very time-consuming, we only choose three feature extraction methods which are proved very efficient and representative in recent studies. We still make our effort to improve our prediction engines in order to make them more easily to compare with more feature extraction methods.

Usually, to construct a highly credible and powerful classifier for protein prediction, the following rules need to be followed (Chou, 2011): (1) a rigorous benchmark dataset, (2) proper descriptors of protein data, (3) the model of algorithm or method, and (4) evaluation criterion. In this paper, we built two multi-label learning models to predict three multi-location protein benchmark datasets according these rules mentioned above. The basic consideration of the first model is a multi-label neutral network deriving from the common BP neural network by modifying it in two respects (Zhang, 2006). The basic consideration of the second model is another multi-label neural network deriving from the common RBF algorithm (Zhang, 2009). It has two layers. In the first layer, by using K-means clustering method on examples from every single class, prototype vectors of its basis functions are set to centroids of clustered groups (Zhang, 2009). Then, the second-layer weights are determined by a sum-of-squares function which is minimized, so the information included in the source vectors can be fully absorbed during the process of optimizing (Zhang, 2009). Application to three rigorous benchmark dataset shows that RBF neutral network is superior to another well-developed multi-label neural network no matter which type of feature extraction is chosen.

2. Materials and methods

2.1. Dataset

2.1.1. Gram-positive bacterial protein benchmark dataset

We choose to use the same dataset S1 in constructing Gpos-mPloc (Shen and Chou, 2009) as the benchmark dataset for this paper. This dataset includes both singleplex and multiplex proteins and was established specialized for Gram-positive bacterial proteins. It covers 4 subcellular location sites containing 519 Gram-positive bacterial protein sequences, and of the 519 different proteins, 515 belong to only 1 location, 4 to 2 locations, and none have three or more locations. The detailed procedures were described particularly in early study (Shen and Chou, 2009). After strictly following the procedures mentioned in this study, a benchmark data set S1 covering 4 subcellular locations is received. The information of the benchmark dataset is listed in Table 1. All the proteins located into 4 subcellular locations can be represented by the procedure described in Zhu et al. (2009).

This dataset can be downloaded at the website http://www.csbio.sjtu.edu.cn/bioinf/Gpos-multi/Data.htm.

2.1.2. Plant protein benchmark dataset

We choose to use the same dataset S2 in constructing Plant-mPloc (Chou and Shen, 2010) as the benchmark dataset for this paper. This dataset includes both singleplex and multiplex proteins and was established specialized for plant proteins. It covers 12 subcellular location sites containing 978 plant protein sequences, and of the 978 different proteins, 904 belong to only 1 location, 71 to 2 locations, 3 to 3 locations, and none have four or more locations. The detailed procedures were described particularly in early study (Chou and Shen, 2010). After strictly following the procedures mentioned in this study, a benchmark data set S2 covering 12 subcellular locations is received. In the same subcellular group, sequences which have more than 25% sequence identity were excluded. The information of the benchmark dataset is listed in Table 2.

As mention above, the representation of the benchmark dataset can be dealt with similarly to that of Gram-positive bacterial protein benchmark dataset.

This dataset can be downloaded at the website http://www.csbio.sjtu.edu.cn/bioinf/plant-multi/Data.htm.

Table 1
Detail of the Gram-positive bacterial protein benchmark dataset S1 derived from Gpos-mPloc (Shen and Chou, 2009).

Order	Subcellular location	Number of proteins
1	Cell membrane	174
2	Cell wall	18
3	Cytoplasm	208
4	Extracell	123
Total number of locative proteins		523
Total number of different proteins		519

Of the 519 different proteins, 515 belongs to only 1 location, 4 to 2 locations, and none have 3 or more locations; i.e., total 523 locative proteins.

2.1.3. Viral protein benchmark dataset

We choose to use the same dataset S3 in constructing Virus-mPloc (Shen and Chou, 2010) as the benchmark dataset for this paper. This dataset includes both singleplex and multiplex proteins and was established specialized for viral proteins. It covers 6 subcellular location sites containing 207 viral protein sequences, and of the 207 different proteins, 165 belong to only 1 location, 39 to 2 locations, 3 to 3 locations, and none have four or more locations. The detailed procedures were described particularly in early study (Shen and Chou, 2010). After strictly following the procedures mentioned in this study, a benchmark data set S3 covering 6 subcellular locations is received. In the same subcellular group, sequences which have more than 25% sequence identity were excluded. The information of the benchmark dataset is listed in Table 3. As mention above, the representation of the benchmark dataset can be dealt with similarly to that of Gram-positive bacterial benchmark dataset.

This dataset can be downloaded at the website http://www.csbio.sjtu.edu.cn/bioinf/virus-multi/Data.htm.

2.2. Feature extraction

2.2.1. Position-specific scoring matrix (PSSM)

Mapping the protein sequences into effective mathematical describers which contain enough information about their subcellular localization is one of the most important steps of developing a powerful classifier. In this paper, we use the PSSM matrix which can incorporate the evolution information of protein sequences to represent the sample of single protein sequence. The construction process is described as follows:

Table 2
Detail of the plant protein benchmark dataset S2 taken from Plant-mPloc (Chou and Shen, 2010).

Order	Subcellular location	Number of proteins
1	Cell membrane	56
2	Cell wall	32
3	Chloroplast	286
4	Cytoplasm	182
5	Endoplasmic reticulum	42
6	Extracellular	22
7	Golgi apparatus	21
8	Mitochondrion	150
9	Nucleus	152
10	Peroxisome	21
11	Plastid	39
12	Vacuole	52
Total number of locative proteins		1055
Total number of different proteins		978

Of the 978 different proteins, 904 belong to only 1 location, 71 to 2 locations, 3 to 3 locations, and none have four or more locations; i.e., total 1055 locative proteins.

Table 3
Detail of the viral protein benchmark dataset S3 taken from Virus-mPloc (Shen and Chou, 2010).

Order	Subcellular location	Number of proteins
1	Viral capsid	8
2	Host cell membrane	33
3	Host endoplasmic reticulum	20
4	Host cytoplasm	87
5	Host nucleus	84
6	Secreted	20
Total number of locative proteins		252
Total number of different proteins		207

Of the 207 different proteins, 165 belong to only 1 location, 39 to 2 locations, 3 to 3 locations, and none have four or more locations; i.e., total 252 locative proteins.

Step 1. According to studies (Chou and Shen, 2007a,b; Wu et al., 2011), the sequential evolution information of protein P can be represented as follows:

$$\mathrm{PSSM} = \begin{bmatrix} U^{\oplus}_{1\to 1} & U^{\oplus}_{1\to 2} & \cdots & U^{\oplus}_{1\to 20} \\ U^{\oplus}_{2\to 1} & U^{\oplus}_{2\to 2} & \cdots & U^{\oplus}_{2\to 20} \\ \vdots & \vdots & \ddots & \vdots \\ U^{\oplus}_{L\to 1} & U^{\oplus}_{L\to 2} & \cdots & U^{\oplus}_{L\to 20} \end{bmatrix} \quad (1)$$

This is a $L \times 20$ matrix, where L is the length of P (counted in the total number of its constituent amino acids as shown in Eq. (1)), $U^{\oplus}_{i\to j}$ represents the score of the amino acid residue in the ith position of the protein sequence which is changed into amino acid type j during the process of evolution. In the matrix, the Arabic numerals 1,2,. . .,20 are used to represent the 20 native amino acid types according to the alphabetical order of their single character codes. The $L \times 20$ scores in Eq. (1) were produced by using PSI-BLAST software (Altschul et al., 1997; Schaffer et al., 2001) to search the UniprotKB/Swiss-Prot database (released on 201108) through three iterations with 0.001 as the E-value cutoff which is used for multiple sequence alignment against the sequence of the protein P. To make the descriptor become a uniform sized matrix which can be dealt with procedure conveniently, following steps serve to solve the problem.

Step 2. This step is a standardization procedure. Use the data in PSSM of Eq. (1) to define a new matrix M as represented by

$$M = \begin{bmatrix} U_{1\to 1} & U_{2\to 1} & \cdots & U_{L\to 1} \\ U_{1\to 2} & U_{2\to 2} & \cdots & U_{L\to 2} \\ \vdots & \vdots & \ddots & \vdots \\ U_{1\to 20} & U_{2\to 20} & \cdots & U_{L\to 20} \end{bmatrix}^{T} \quad (2)$$

with

$$U_{i\to j} = \frac{U^{\oplus}_{i\to j} - (1/20)\sum_{k=1}^{20} U^{\oplus}_{i\to k}}{\sqrt{(1/20)\sum_{l=1}^{20}\left(U^{\oplus}_{i\to l} - (1/20)\sum_{k=1}^{20} U^{\oplus}_{i\to k}\right)^2}} \times (i = 1, 2, \ldots, L;\ \ j = 1, 2, \ldots, 20) \quad (3)$$

After that, the protein sample sequence P can be represented by

$$\bar{P} = [\bar{U}_1 \quad \bar{U}_2 \quad \cdots \quad \bar{U}_{20}]^T \quad (4)$$

where

$$\bar{U}_j = \frac{1}{L}\sum_{i=1}^{L} U_{i\to j} \quad (j = 1, 2, \cdots, 20) \quad (5)$$

Step 3. In order to utilize the complete sequence-order information, we use the pseudo position-specific scoring matrix (PsePSSM) as defined by

$$P^{\mu}_{\mathrm{pre}} = \left[\bar{U}_1 \quad \bar{U}_2 \quad \cdots \quad \bar{U}_{20} \quad \Psi^{\mu}_1 \quad \Psi^{\mu}_2 \quad \cdots \quad \Psi^{\mu}_{20}\right]^T \quad (6)$$

to replace the protein sequence P, where

$$\Psi^{\mu}_j = \frac{1}{L-\mu}\sum_{i=1}^{L-\mu}[U_{i\to j} - U_{(i+\mu)\to j}]^2 \quad (j = 1, 2, \cdots, 20;\ \ \mu < L) \quad (7)$$

The Ψ^1_j is amount to the correlation factor through combining the most contiguous PSSM values associated with protein and the amino acid type; Ψ^2_j is represent the correlation factor through combining the second most contiguous PSSM values associated with protein and the amino type; and so on. Considering the fact that the length of the shortest protein sequence we used is 50, so the legitimate value of μ must be less than 50, according to (Zhu et al., 2009), we choose the $\mu = 2$ in this study.

2.2.2. Chou's pseudo amino acid based features (PseACC)

This is a type of protein descriptor proposed by Chou (2001) which avoid losing sequence ordering information from the protein samples.

Suppose a protein including L amino acid residues can be represented as:

$$P = Q_1 Q_2 Q_3 Q_4 \ldots Q_L \quad (8)$$

where Q_1 is the residue at the first position along the sequence and Q_2 the residue at the second position and so forth.

The sequence-order information can be indirectly represented by the following equations:

$$\delta_\theta = \frac{1}{L-\theta}\sum_{i=1}^{L-\theta}\Omega(Q_i, Q_{i+1}), \quad (\theta = 1, 2, \cdots, \eta \ \text{ and } \ \eta < L) \quad (9)$$

where L denotes the length of the protein and the δ_θ is called the θth correlation factor which harbors the sequence order information between all the θ most contiguous residues. The correlation function $\Omega(Q_i, Q_{i+1})$ is defined by:

$$\Omega(Q_i, Q_j) = \frac{1}{3}\{[F(Q_j) - F(Q_i)]^2 + [G(Q_j) - G(Q_i)]^2 + [H(Q_j) - H(Q_i)]^2\} \quad (10)$$

where $F(Q_i)$, $G(Q_i)$ and $H(Q_i)$ are evaluated the value of hydrophobicity, hydrophilicity and mass, respectively. There are also another three types of value (pK_1, pK_2 and pI) can be used. In this study, the method of feature extraction which uses the value of hydrophobicity, hydrophilicity and mass is named "PseACC1", and the method of feature extraction which uses the value of pK_1, pK_2 and pI is named "PseACC2". These data were achieved from the webserver PseACC (Shen and Chou, 2008). Before we use these values, a standard conversion described by Eq. (4) of Chou (2001) should be conducted.

Then a sample protein P can be represented as

$$P = [\mu_1, \mu_2, \cdots, \mu_{20}, \quad \mu_{20+1}, \mu_{20+2}, \cdots, \mu_{20+\eta}] \quad \eta < La \quad (11)$$

where the first 20 components are the same to conventional AAC, The other components which are related to η different ranks of sequence-order correlation factors which can be calculated directly using Eqs. (2)–(6) by Chou (2001), and we set $\eta = 20$ and the weight factor used in these components to be 0.05 for easy handing.

Recently, two powerful web-servers (Cao et al., 2013; Du et al., 2012) were established to generate various different modes of Chou's PseAAC. Furthermore, the concept of PseAAC and its general form have been extended to the feature vectors of DNA and nucleotides (Chen et al., 2012, 2013), as well as other biological samples (see, e.g., Huang et al., 2012; Li et al., 2012).

2.3. Algorithms for classification

In this study, we adapt the definitions used by (Zhang, 2006, 2009). Define $\mathbb{Q} = H^d$ to be the input vector space, $\mathbb{C} = \{1, 2, 3, \ldots, C\}$ to be the set of C possible labels and $\mathbb{Z} = \{(q_i, t_i), 1 \le i \le N\}$ to be a train set in which $q_i \in \mathbb{Q}$, $t_i \subseteq \mathbb{C}$, using $\mathbb{Z}$ to train the multi label classier. Usually, the learning model will output a real valued vector based on the function $g: \ \mathbb{Q} \times \mathbb{C} \Rightarrow \mathrm{H}$. Considering q_i and its corresponding label set t_i, $g(*,*)$ is devised to have the character of $g(q_i, c_1) < g(q_i, c_2)$ when any $c_1 \notin t_1$ and $c_2 \in t_2$. Apparently, the result yields bigger values for labels belonging to t_i rather than those not belonging to t_i. In addition, we use $rank_g(g_i, c)$ to represent the ranking function derived from $g(q_i, c)$ which computes the rank of $c (c \in \mathbb{C})$. Clearly, the larger value of $g(q_i, c)$ corresponds with the higher rank of c. The multi label classifier $t(*)$ can also be computed by $g(*,*)$ as: $t(q_i) = \{c | g(q_i, c) > v(q_i), c \in \mathbb{C}\}$, where $v(*)$ is a threshold function usually set to be 0 for easy handling.

2.3.1. BP neural network

Back-propagation (BP) neural network is a kind of traditional multi-layer feed-forward neural networks, and it is adapted in learning from multi-label examples (Zhang, 2006). Detailed information about BP network can be received in Zhang (2006).

Changing traditional feed-forward neural network handling single-label examples into multi-label example needs revisions in two respects. The first is to define some new error function instead of the sum-of-squares function. The second is introducing some revision for the classical learning algorithm to minimize the error

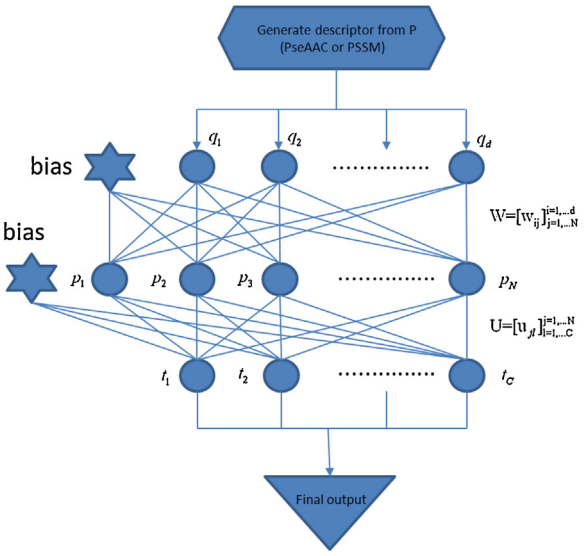

Fig. 1. Multiple label learning process and the BP network structure.

function newly defined. These details of two crucial revisions can be viewed in Zhang (2006).

Fig. 1 depicts the multiple label learning process and the BP network structure. The detail of its structure can be learned in Zhang (2006). Its source code can be downloaded at http://cse.seu.edu.cn/people/zhangml/Resources.htm#codes.

2.3.2. RBF neural network

Radial basis function (RBF) was derived from some old pattern techniques, and was one of the most popular models of neural networks. It was devised to match a two layer neural network in which each unit in the first layer associated with a radical activated function while the output unit implants a weighted sum of hidden unit corresponding to a possible label. Detailed information about RBF network can be received in Zhang (2009).

Fig. 2 depicts the multiple label learning process and the RBF network structure. The detail of its structure can be learned in Zhang (2009). Its source code can be downloaded at http://cse.seu.edu.cn/people/zhangml/Resources.htm#codes.

2.3.3. Evaluation measures

Given a multi-label test data set $\chi = \left\{(x_i, X_i)1 \le i \le n\right\}$, based on the definition in Section 2.3, the following popular multi-label evaluation metrics (Schapire and Singer, 2000; Zhang, 2006, 2009; Zhang et al., 2009; Zhang and Zhou, 2007) are used:

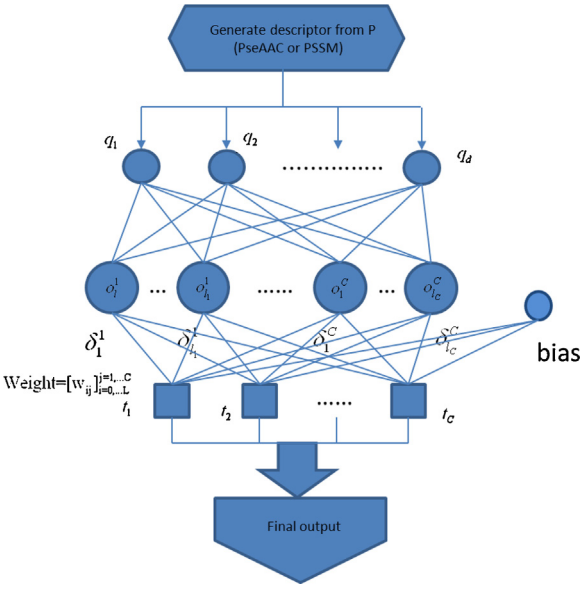

Fig. 2. Multiple label learning process and the RBF network structure.

(1) Hamming loss:

$$\text{hamming\_loss}\quad \chi(t) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{C}|t(x_i)\Delta X_i| \tag{12}$$

Δ represents the symmetric difference between two data sets

(2) One error

$$\text{one\_error}\quad \chi(g) = \frac{1}{n}\sum_{i=1}^{n}[[\arg\max_{c\in\mathbb{C}} g(x_i, c)] \notin X_i] \tag{13}$$

(3) Coverage:

$$\text{coverage}\quad \chi(g) = \frac{1}{n}\sum_{i=1}^{n}\max_{c\in X_i} rank_g(x_i, c) - 1 \tag{14}$$

(4) Ranking loss:

$$\text{ranking\_loss}\quad \chi(g) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|X_i||\bar{X}_i|}\left|\text{RAL}(x_i)\right|,$$

$$\text{where}\quad \text{RAL}(x_i) = \{(t_1, t_2)|g(x_i, t_1) \le g(x_i, t_2),\quad (t_1, t_2)\in X_i \times \bar{X}_i\} \tag{15}$$

(5) Average precision:

$$\text{average\_prec}\quad \chi(g) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|X_i|}\sum_{c\in X_i}\text{AVP}(x_i),$$

$$\text{where}\quad \text{AVP}(x_i) = \frac{\left|c'|rank_g(x_i, c') \le rank_g(x_i, c),\quad c'\in X_i\right|}{rank_g(x_i, c)} \tag{16}$$

To sum up, hamming loss evaluates the times that an instance-label pair is misclassified; one-error evaluates the times that the top-ranked label is not in the set of proper labels of the instance; Coverage evaluates the number of steps needed, on the average, to move down the label list in order to cover all the proper labels attached to an instance; Ranking loss examines the average fraction of label pairs that are reversely ordered for the instance; Average precision evaluates the average fraction of labels which are ranked above a particular label $c \in X$ and really are in X. Note that for the first four metrics, the smaller the better, and for the last one, the larger the better performance.

In recent years, the "absolute true" or "exact match" rate was also introduced (Chou et al., 2011, 2012) to measure the accuracy of a predictor in dealing with a system containing both single and multiple location proteins (see Eqs. (26) and (27) in Chou et al., 2011 and Eqs. (28) and (29) in Chou et al., 2012). According to the definition of the absolute true metrics, when and only when all the subcellular locations of a query protein are exactly predicted without any under prediction or over prediction, can the prediction be scored with 1; otherwise, 0. Therefore, the absolute true scale is much more strict and harsh than all the other metrics. However, the multi-label system is much more complicated than the single system, and it is still immature to use this metric broadly. In this paper, the five metrics defined by Eqs. (12)–(16) were adopted.

Owing to the fact that it is still a very difficult and complicated problem for how to define a set of metrics to properly measure the prediction quality for a multi-label, recently, a new set of five different metrics to measure the prediction quality for a multi-label system from five different angles was given in Eq. (16) of Chou (2013) and Eq. (21) of Lin et al. (2013). The aforementioned "absolute true" rate was included in this set of metrics. In these two papers, an in-depth discussion about the five metrics and their implications were also elaborated. Due to the comprehensiveness and understandability of these metrics, it is very meaningful and interesting to make efforts to apply these five metrics in our future study.

3. Results and discussion

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical applications: independent dataset test, sub-sampling test, and jackknife test (Chou et al., 2011, 2012; He et al., 2010; Hu et al., 2011; Wu et al., 2011, 2012; Xiao et al., 2011a,b). Since the subsampling test and the jackknife test can be performed with one benchmark dataset and that the independent dataset test can be treated as a special case of the subsampling test, one benchmark dataset would suffice to serve all the three kinds of cross validations

Table 4
Performance of each compared algorithm (mean ± SD) on Gram-Positive bacterial protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.3786 ± 0.0334	0.4177 ± 0.0237	0.4195 ± 0.0195	**0.1401 ± 0.0132**	0.1537 ± 0.0153	0.1522 ± 0.0256
One-error↓	0.7631 ± 0.0708	0.7631 ± 0.0708	0.7629 ± 0.0352	**0.2832 ± 0.0302**	0.3045 ± 0.0347	0.3120 ± 0.0564
Coverage↓	1.2448 ± 0.2474	1.1717 ± 0.1148	1.1713 ± 0.0625	**0.4392 ± 0.0395**	0.4509 ± 0.0683	0.4854 ± 0.0776
Ranking loss↓	0.4128 ± 0.0823	0.3885 ± 0.0388	0.3883 ± 0.0209	**0.1443 ± 0.0118**	0.1476 ± 0.0202	0.1599 ± 0.0281
Average precision↑	0.5484 ± 0.0517	0.5545 ± 0.0429	0.5547 ± 0.0218	**0.8366 ± 0.0153**	0.8267 ± 0.0181	0.8191 ± 0.0310

Table 5
Computation time of each compared algorithm (mean ± SD) on Gram-Positive bacterial protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC + RBF	PseACC2 + RBF
Computation time	1015.8 ± 13.6 s	1117.6 ± 90.5 s	994.2 ± 10.0 s	0.2 ± 0.2 s	**0.08 ± 0.008 s**	**0.08 ± 0.002**

Table 6
Performance of each compared algorithm (mean ± SD) on Plant protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.0920 ± 0.0042	0.0901 ± 0.0014	0.0905 ± 0.0017	**0.0861 ± 0.0020**	0.0868 ± 0.0038	0.0882 ± 0.0035
One-error↓	0.8446 ± 0.0291	0.8169 ± 0.0715	0.8446 ± 0.0237	0.6105 ± 0.0340	**0.5828 ± 0.0251**	0.5941 ± 0.0184
Coverage↓	4.6653 ± 0.3379	3.0874 ± 0.3662	3.0864 ± 0.3156	2.0798 ± 0.0417	2.0604 ± 0.1432	**2.0538 ± 0.1928**
Ranking loss↓	0.4076 ± 0.0282	0.2692 ± 0.0298	0.2689 ± 0.0296	0.1753 ± 0.0069	**0.1735 ± 0.0107**	0.1741 ± 0.0142
Average precision↑	0.3160 ± 0.0303	0.4174 ± 0.0556	0.4083 ± 0.0273	0.5846 ± 0.0217	**0.5949 ± 0.0193**	0.5898 ± 0.0134

Table 7
Computation time of each compared algorithm (mean ± SD) on Plant protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Computation time	2190.2 ± 36.2 s	2085.5 ± 6.4 s	1991.5 ± 5.2	0.4 ± 0.2 s	**0.3 ± 0.02 s**	**0.3 ± 0.01**

Table 8
Performance of each compared algorithm (mean ± SD) on virus protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.2358 ± 0.0339	0.2635 ± 0.0339	0.2617 ± 0.0138	0.1844 ± 0.0094	0.1764 ± 0.0161	**0.1763 ± 0.0130**
One-error↓	0.5944 ± 0.0574	0.5944 ± 0.0574	0.5947 ± 0.0611	0.4645 ± 0.0921	0.4443 ± 0.0671	**0.4394 ± 0.0354**
Coverage↓	1.6915 ± 0.3110	1.5125 ± 0.2018	1.5488 ± 0.2464	**1.1941 ± 0.2854**	1.2660 ± 0.1180	1.2516 ± 0.2254
Ranking loss↓	0.2953 ± 0.0648	0.2605 ± 0.0434	0.2676 ± 0.0377	**0.1909 ± 0.0547**	0.2003 ± 0.0262	0.1977 ± 0.0287
Average precision↑	0.6075 ± 0.0621	0.6279 ± 0.0445	0.6246 ± 0.0306	0.7120 ± 0.0652	0.7150 ± 0.0379	**0.7164 ± 0.0201**

Table 9
Computation time of each compared algorithm (mean ± SD) on virus protein data by using 5-fold cross-validation.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Computation time	448.1 ± 39.5 s	425.7 ± 2.9 s	426.0 ± 2.9 s	0.2 ± 0.2 s	**0.04 ± 0.004** s	**0.04 ± 0.002**

Table 10
Performance of each compared algorithm on Gram-positive bacterial protein data by using independent dataset test by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.3725	0.3725	0.4020	0.1422	0.1716	0.1422
One-error↓	0.7451	0.7647	0.7647	0.3725	0.3137	0.3333
Coverage↓	1.1373	1.0784	1.1765	0.5490	0.6275	0.5490
Ranking loss↓	0.3791	0.3595	0.3856	0.1765	0.1944	0.1748
Average precision↑	0.5654	0.5670	0.5588	0.7908	0.8039	0.8007

Table 11
Computation time of each compared algorithm on Gram-positive bacterial protein data by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Computation time	1113.6 s	1150.4 s	1132.4 s	0.08 s	0.06 s	0.06 s

Table 12
Performance of each compared algorithm on Plant protein data by using independent dataset test by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.0911	0.0988	0.0868	0.0816	0.0885	0.0851
One-error↓	0.9278	0.9278	0.8247	0.6186	0.5979	0.5979
Coverage↓	5.1237	3.4330	2.8247	2.1546	2.0206	2.1340
Ranking loss↓	0.4463	0.2964	0.2488	0.1852	0.1776	0.1803
Average precision↑	0.2668	0.3662	0.4311	0.5752	0.5866	0.5794

Table 13
Computation time of each compared algorithm on Plant protein data by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Computation time	2094.2 s	2073.1 s	2381.0 s	0.19 s	0.23 s	0.21 s

Table 14
Performance of each compared algorithm on virus protein data by using independent dataset test by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Hamming loss↓	0.3083	0.3500	0.2750	0.1750	0.1417	0.2167
One-error↓	0.7000	0.7500	0.6500	0.4500	0.3500	0.3500
Coverage↓	2.3000	1.9500	1.8000	1.6000	1.0500	1.3000
Ranking loss↓	0.3968	0.3588	0.3000	0.2488	0.1536	0.1850
Average precision↑	0.5354	0.5054	0.5858	0.6925	0.7850	0.7538

(Chou et al., 2011, 2012; Wu et al., 2011, 2012; Xiao et al., 2011a,b). However, as elucidated in Chou and Shen (2008) and demonstrated by Eqs. (28)–(30) in Chou (2011), among the three test methods, the jackknife test is deemed the least arbitrary that can always yield a unique result for a given benchmark dataset. Accordingly, the jackknife test has been widely recognized and increasingly used by investigators to examine the quality of various predictors (Chen et al., 2012; Chou, 2011; Chou and Shen, 2008; Esmaeili et al., 2010; Hayat and Khan, 2012; Lin et al., 2012; Mei, 2012; Mohabatkar et al., 2011, 2013; Nanni et al., 2012; Sun et al., 2012). However, to reduce the computational time, we adopted the 5-fold cross-validation in this study as done by many investigators with SVM as the prediction engine.

Meanwhile, independent dataset test was also adopted in our study. The independent testing dataset was constructed by selecting 10% samples from each benchmark dataset randomly, and the rest of 90% samples in each benchmark dataset were used to construct the training dataset accordingly.

Tables 4, 6 and 8 provide the test results of six combination of test method on the Gram-positive bacterial protein, plant protein and virus protein benchmark datasets respectively. For each of these datasets, 5-fold cross-validation is performed on them and the performance (mean value ± standard deviation) out of five independent runs are presented. As the table shown, for each evaluation measurement, "↓" represents the smaller the better, whereas the "↑" represents the larger the better. In addition, the best result on each measurement is presented in bold face.

Table 4 shows that the combination of PSSM and RBF yields the best performance on every evaluation measurement. Table 6 shows that the combination of PseACC1 and RBF achieves rather superior performance on three evaluation measurements only inferior to the combination of PSSM and RBF on hamming loss and the combination of PseACC2 and RBF on coverage. Table 8 shows that the combination of PseACC2 and RBF achieves superior performance on three evaluation measurements only inferior to the combination of PSSM and RBF on coverage and ranking loss. What is more, the result using RBF significantly outperform that using BP neural network on every evaluation measurement no matter which dataset is tested.

On the whole, Tables 4, 6 and 8 reveal that RBF neural network is a superior neural network based multi-label algorithm no matter which type of feature extraction was chosen. Moreover, it deserves to notice that one combination (methods of feature extraction + prediction algorithm) does not necessary yield the best performance on different dataset in every metric, so it is very interesting and meaningful to find the best combination when we focus on a specific dataset. Moreover, establishing a fusion classier which incorporate several different methods of feature extraction may be a more reliable and robust choice when we want to develop a solution which can deal with different kind of biological datasets simultaneously.

Tables 5, 7 and 9 give the computation time (training time + testing time) by each combination, where all the tests were conducted on a PC with 2G RAM and Quad-Core of AMD Phenom™ 9550 CPUs each running at 2.20 GHz. From these tables, it is clear that the combinations using BP neural network are much more time-consuming than combinations using RBF neural network.

Tables 10–15 give the test results and the computation time for six combination of test method on the Gram-positive bacterial protein, plant protein and virus protein benchmark datasets

Table 15
Computation time of each compared algorithm on virus protein data by using independent dataset test.

Evaluation criterion	Algorithm					
	PSSM + BP	PseACC1 + BP	PseACC2 + BP	PSSM + RBF	PseACC1 + RBF	PseACC2 + RBF
Computation time	450.8 s	444.3 s	447.3 s	0.03 s	0.04 s	0.03 s

by using independent dataset test, respectively. It is necessary to point out that the independent dataset test adopted here was just an exposition of practical prediction, because the independent dataset test bears much more sort of arbitrariness than 5-fold cross-validation during the process of our study. Therefore, the 5-fold cross-validation was adopted as the criterion for measuring the effectiveness of predictors in our study.

Compared with the single-label problems, the multi-label problems are far much more difficult and complicated to deal with. Considering the fact that the BP neural network even has several drawbacks when it is adopted to deal with single-label problems, it still needs to improve a lot in order to meet the high demand of multi-label system. In this study, the RBF neural network also shows it is a more reliable and stable prediction model than BP neural network due to some unreasonable results existed in the process of cross-validation test by using BP neural network.

4. Conclusions

In this paper, six combinations of feature extraction method and predict algorithm were conducted on three multi-label benchmark datasets. The predict algorithms are two neural networks which are revised from traditional neural networks in order to be competent for the prediction of multi-label task. Comparative studies on three benchmark datasets show that RBF neural network achieve rather competitive performance to BP neural network no matter which type of feature extraction is used, and the combination of PseACC (1 and 2) and RBF achieves the superior performance in general.

Even though the Gram-positive bacterial protein benchmark dataset has less multiple subcellular localization, we still chose it based on the fact that this kind of Gram-positive proteins is also a very important kind of protein deserving our attention, the procedure to establish this dataset is exactly the same as the procedure of dataset S2 and S3 and the dataset has already been used in two successful studies (Shen and Chou, 2009; Wu et al., 2012). With the avalanche of protein sequences generated nowadays, we will make effort to establish new Gram-positive proteins dataset in our future studies.

Exploring whether better performance can be received by using other method of feature extraction or other predict algorithm is still a meaningful task. Moreover, studying the result received by setting different parameters in neural network is also a meaningful task.

Taking account of the fact that user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful predictors (Chou and Shen, 2009), we shall make efforts in our future work to provide a web-server for the methods presented in this study.

Ethical standards

The experiments comply with the current laws of the country in which they were performed.

Acknowledgments

C. Huang would like to express his thank to reviewers for their suggestions. This study was supported by the Doctoral Program of Higher Education of China (Grant No. 20110073110018).

References

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

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G., 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29.

Cai, Y.D., Lu, L., Chen, L., He, J.-F., 2010. Predicting subcellular location of proteins using integrated-algorithm method. Mol. Divers. 14, 551–558.

Cao, D.S., Xu, Q.S., Liang, Y.Z., 2013. Propy: a tool to generate various modes of Chou's PseAAC. Bioinformatics 29, 960–962.

Chen, W., Feng, P.M., Lin, H., Chou, K.C., 2013. iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. Nucleic Acids Res. 41, e68, http://dx.doi.org/10.1093/nar/gks1450.

Chen, W., Lin, H., Feng, P.M., Ding, C., Zuo, Y.C., Chou, K.C., 2012. iNuc-PhysChem: a sequence-based predictor for identifying nucleosomes via physicochemical properties. PLoS ONE 7, e47843.

Chou, K.C., 2001. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins 43, 246–255.

Chou, K.C., 2011. Some remarks on protein attribute prediction and pseudo amino acid composition. J. Theor. Biol. 273, 236–247.

Chou, K.C., 2013. Some remarks on predicting multi-label attributes in molecular biosystems. Mol. Biosyst. 9, 1092–1100.

Chou, K.C., Shen, H.B., 2006a. Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization. Biochem. Biophys. Res. Commun. 347, 150–157.

Chou, K.C., Shen, H.B., 2006b. Large-scale predictions of Gram-negative bacterial protein subcellular locations. J. Proteome Res. 5, 3420–3428.

Chou, K.C., Shen, H.B., 2007a. MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem. Biophys. Res. Commun. 360, 339–345.

Chou, K.C., Shen, H.B., 2007b. Recent progress in protein subcellular location prediction. Anal. Biochem. 370, 1–16.

Chou, K.C., Shen, H.B., 2008. Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms. Nat. Protocols 3, 153–162.

Chou, K.C., Shen, H.B., 2009. REVIEW: recent advances in developing web-servers for predicting protein attributes. Nat. Sci. 1, 63–92.

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

Chou, K.C., Wu, Z.C., Xiao, X., 2011. iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. PLoS ONE 6, e18258.

Chou, K.C., Wu, Z.C., Xiao, X., 2012. iLoc-Hum: using the accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites. Mol. Biosyst. 8, 629–641.

Du, P., Wang, X., Xu, C., Gao, Y., 2012. PseAAC-builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions. Anal. Biochem. 425, 117–119.

Ehrlich, J.S., Hansen, M.D., Nelson, W.J., 2002. Spatio-temporal regulation of Rac1 localization and lamellipodia dynamics during epithelial cell–cell adhesion. Dev. Cell 3, 259–270.

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

Esmaeili, M., Mohabatkar, H., Mohsenzadeh, S., 2010. Using the concept of Chou's pseudo amino acid composition for risk type prediction of human papillomaviruses. J. Theor. Biol. 263, 203–209.

Glory, E., Murphy, R.F., 2007. Automated subcellular location determination and high-throughput microscopy. Dev. Cell 12, 7–16.

Hayat, M., Khan, A., 2012. Discriminating outer membrane proteins with Fuzzy K-nearest Neighbor algorithms based on the general form of Chou's PseAAC. Protein Pept. Lett. 19, 411–421.

He, Z., Zhang, J., Shi, X.H., Hu, L.L., Kong, X., Cai, Y.D., Chou, K.C., 2010. Predicting drug-target interaction networks based on functional groups and biological features. PLoS ONE 5, e9603.

Hu, L.L., Feng, K.Y., Cai, Y.D., Chou, K.C., 2012a. Using protein–protein interaction network information to predict the subcellular locations of proteins in budding yeast. Protein Pept. Lett. 19, 644–651.

Hu, L.L., Huang, T., Cai, Y.D., Chou, K.C., 2011. Prediction of body fluids where proteins are secreted into based on protein interaction network. PLoS ONE 6, e22989.

Hu, Y., Li, T., Sun, J., Tang, S., Xiong, W., Li, D., Chen, G., Cong, P., 2012b. Predicting Gram-positive bacterial protein subcellular localization based on localization motifs. J. Theor. Biol. 308, 135–140.

Huang, T., Wang, J., Cai, Y.D., Yu, H., Chou, K.C., 2012. Hepatitis C virus network based classification of hepatocellular cirrhosis and carcinoma. PLoS ONE 7, e34460.

Jin, Y.H., Niu, B., Feng, K.Y., Lu, W.C., Cai, Y.D., Li, G.Z., 2008. Predicting subcellular localization with AdaBoost Learner. Protein Pept. Lett. 15, 286–289.

Li, B.Q., Huang, T., Liu, L., Cai, Y.D., Chou, K.C., 2012. Identification of colorectal cancer related genes with mRMR and shortest path in protein–protein interaction network. PLoS ONE 7, e33393.

Lin, H., Ding, H., Guo, F.B., Zhang, A.Y., Huang, J., 2008. Predicting subcellular localization of mycobacterial proteins by using Chou's pseudo amino acid composition. Protein Pept. Lett. 15, 739–744.

Lin, H., Wang, H., Ding, H., Chen, Y.L., Li, Q.Z., 2009. Prediction of subcellular localization of apoptosis protein using Chou's pseudo amino acid composition. Acta Biotheor. 57, 321–330.

Lin, W.Z., Fang, J.A., Xiao, X., Chou, K.C., 2012. Predicting secretory proteins of malaria parasite by incorporating sequence evolution information into pseudo amino acid composition via grey system model. PLoS ONE 7, e49040.

Lin, W.Z., Fang, J.A., Xiao, X., Chou, K.C., 2013. iLoc-Animal: a multi-label learning classifier for predicting subcellular localization of animal proteins. Mol. Biosyst. 9, 634–644.
Luo, H., 2012. Predicted protein subcellular localization in dominant surface ocean bacterioplankton. Appl. Environ. Microbiol. 78, 6550–6557.
Matsuda, S., Vert, J.P., Saigo, H., Ueda, N., Toh, H., Akutsu, T., 2005. A novel representation of protein sequences for prediction of subcellular location using support vector machines. Protein Sci. 14, 2804–2813.
Mei, S., 2012. Predicting plant protein subcellular multi-localization by Chou's PseAAC formulation based multi-label homolog knowledge transfer learning. J. Theor. Biol. 310, 80–87.
Mohabatkar, H., Beigi, M.M., Abdolahi, K., Mohsenzadeh, S., 2013. Prediction of allergenic proteins by means of the concept of Chou's pseudo amino acid composition and a machine learning approach. Med. Chem. 9, 133–137.
Mohabatkar, H., Mohammad Beigi, M., Esmaeili, A., 2011. Prediction of GABAA receptor proteins using the concept of Chou's pseudo-amino acid composition and support vector machine. J. Theor. Biol. 281, 18–23.
Nanni, L., Lumini, A., Gupta, D., Garg, A., 2012. Identifying bacterial virulent proteins by fusing a set of classifiers based on variants of Chou's pseudo amino acid composition and on evolutionary information. IEEE/ACM Trans. Comput. Biol. Bioinform. 9, 467–475.
Niu, B., Jin, Y.-H., Feng, K.-Y., Lu, W.-C., Cai, Y.-D., Li, G.-Z., 2008. Using AdaBoost for the prediction of subcellular location of prokaryotic and eukaryotic proteins. Mol. Divers. 12, 41–45.
Pierleoni, A., Martelli, P.L., Fariselli, P., Casadio, R., 2006. BaCelLo: a balanced subcellular localization predictor. Bioinformatics 22, e408–e416.
Schaffer, A.A., Aravind, L., Madden, T.L., Shavirin, S., Spouge, J.L., Wolf, Y.I., Koonin, E.V., Altschul, S.F., 2001. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res. 29, 2994–3005.
Schapire, R.E., Singer, Y., 2000. BoosTexter: a boosting-based system for text categorization. Mach. Learn. 39, 135–168.
Shen, H.B., Chou, K.C., 2007a. Gpos-PLoc: an ensemble classifier for predicting subcellular localization of Gram-positive bacterial proteins. Protein Eng. Des. Sel. 20, 39–46.
Shen, H.B., Chou, K.C., 2007b. Virus-PLoc: a fusion classifier for predicting the subcellular localization of viral proteins within host and virus-infected cells. Biopolymers 85, 233–240.
Shen, H.B., Chou, K.C., 2008. PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal. Biochem. 373, 386–388.
Shen, H.B., Chou, K.C., 2009. Gpos-mPLoc: a top-down approach to improve the quality of predicting subcellular localization of Gram-positive bacterial proteins. Protein Pept. Lett. 16, 1478–1484.
Shen, H.B., Chou, K.C., 2010. Virus-mPLoc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites. J. Biomol. Struct. Dyn. 28, 175–186.
Su, E.C., Chiu, H.S., Lo, A., Hwang, J.K., Sung, T.Y., Hsu, W.L., 2007. Protein subcellular localization prediction based on compartment-specific features and structure conservation. BMC Bioinform. 8, 330.
Sun, X.Y., Shi, S.P., Qiu, J.D., Suo, S.B., Huang, S.Y., Liang, R.P., 2012. Identifying protein quaternary structural attributes by incorporating physicochemical properties into the general form of Chou's PseAAC via discrete wavelet transform. Mol. Biosyst. 8, 3178–3184.
Tejedor-Estrada, R., Nonell, S., Teixido, J., Sagrista, M.L., Mora, M., Villanueva, A., Canete, M., Stockert, J.C., 2012. An artificial neural network model for predicting the subcellular localization of photosensitisers for photodynamic therapy of solid tumours. Curr. Med. Chem. 19, 2472–2482.
Wang, K., Hu, L.L., Shi, X.H., Dong, Y.S., Li, H.P., Wen, T.Q., 2012. PSCL: predicting protein subcellular localization based on optimal functional domains. Protein Pept. Lett. 19, 15–22.
Wu, Z.C., Xiao, X., Chou, K.C., 2011. iLoc-Plant: a multi-label classifier for predicting the subcellular localization of plant proteins with both single and multiple sites. Mol. Biosyst. 7, 3287–3297.
Wu, Z.C., Xiao, X., Chou, K.C., 2012. iLoc-Gpos: a multi-layer classifier for predicting the subcellular localization of singleplex and multiplex Gram-positive bacterial proteins. Protein Pept. Lett. 19, 4–14.
Xiao, X., Wu, Z.C., Chou, K.C., 2011a. 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, 42–51.
Xiao, X., Wu, Z.C., Chou, K.C., 2011b. A multi-label classifier for predicting the subcellular localization of Gram-negative bacterial proteins with both single and multiple sites. PLoS ONE 6, e20592.
Zhang, M.L., 2006. Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. 18, 1338–1351.
Zhang, M.L., 2009. ML-RBF: RBF neural networks for multi-label learning. Neural Process. Lett. 29, 61–74.
Zhang, M.L., Peña, J.M., Robles, V., 2009. Feature selection for multi-label naive Bayes classification. Inform. Sci. 179, 3218–3229.
Zhang, M.L., Zhou, Z.-H., 2007. ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. 40, 2038–2048.
Zhou, G.P., Doctor, K., 2003. Subcellular location prediction of apoptosis proteins. Proteins 50, 44–48.
Zhu, L., Yang, J., Shen, H.B., 2009. Multi label learning for prediction of human protein subcellular localizations. Prot. J. 28, 384–390.