Computational Biology and Chemistry 35 (2011) 218–229

Contents lists available at ScienceDirect

Computational Biology and Chemistry

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

CE-PLoc: An ensemble classifier for predicting protein subcellular locations by fusing different modes of pseudo amino acid composition

Asifullah Khan, Abdul Majid*, Maqsood Hayat

Department of Information and Computer Sciences, Pakistan Institute of Engineering and Applied Sciences, P.O. 45650, Nilore, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 23 February 2011
Received in revised form 17 May 2011
Accepted 18 May 2011

Keywords:
Ensemble classifier
Protein subcellular location
Pseudo amino acid composition
Dipeptide composition

ABSTRACT

Precise information about protein locations in a cell facilitates in the understanding of the function of a protein and its interaction in the cellular environment. This information further helps in the study of the specific metabolic pathways and other biological processes. We propose an ensemble approach called "*CE-PLoc*" for predicting subcellular locations based on fusion of individual classifiers. The proposed approach utilizes features obtained from both dipeptide composition (*DC*) and amphiphilic pseudo amino acid composition (*PseAAC*) based feature extraction strategies. Different feature spaces are obtained by varying the dimensionality using *PseAAC* for a selected base learner. The performance of the individual learning mechanisms such as support vector machine, nearest neighbor, probabilistic neural network, covariant discriminant, which are trained using *PseAAC* based features is first analyzed. Classifiers are developed using same learning mechanism but trained on *PseAAC* based feature spaces of varying dimensions. These classifiers are combined through voting strategy and an improvement in prediction performance is achieved. Prediction performance is further enhanced by developing *CE-PLoc* through the combination of different learning mechanisms trained on both *DC* based feature space and *PseAAC* based feature spaces of varying dimensions. The predictive performance of proposed *CE-PLoc* is evaluated for two benchmark datasets of protein subcellular locations using accuracy, *MCC*, and *Q*-statistics. Using the jackknife test, prediction accuracies of 81.47 and 83.99% are obtained for 12 and 14 subcellular locations datasets, respectively. In case of independent dataset test, prediction accuracies are 87.04 and 87.33% for 12 and 14 class datasets, respectively.

1. Introduction

Considering the fact that the number of newly found proteins is consistently increasing, the importance of automatically annotating the subcellular attributes of uncharacterized proteins and their timely utilization in drug discovery is self-evident. Precise knowledge of a protein's function requires appropriate subcellular localization (Deng et al., 2003), as it has been observed that a protein may lose its functions if not properly localized (Boden and Teasdale, 2008). In addition, information about subcellular location may provide understanding about the engagement in specific metabolic pathways (Garg et al., 2005). Accurate information about the locations of proteins is thus significant to judge the biological nature and task of their activity (Jaimovich et al., 2006; Shen and Burger, 2007). Another motivation behind the research of developing accurate computational models for predicting subcellular localization of unknown proteins is the high temporal cost and expenses incurred in case of experimental methods (Chou and Cai, 2002; Zhang et al., 2006a,b,c; Mei et al., 2011). Experimental methods provide accurate results but unfortunately, for some of the species, proteomics and microscopic detection are almost impossible to conduct, so the demand for developing efficient and reliable computational models is increasing.

In the literature, both individual and fusion of classifier strategies are used to accurately predict subcellular localization. Early attempts are mainly based on individual classifiers. Covariant discriminant (*CDC*) (Chou, 2000, 2001; Chou and Elrod, 1999; Pan et al., 2003), nearest neighbor (*NN*) (Chou and Shen, 2006b; Jia et al., 2007; Khan et al., 2008), and support vector machine (*SVM*) (Chou and Cai, 2002; Shen and Burger, 2010) classifiers are used in conjunction with various feature extraction techniques. However, the prediction system based on a single learning algorithm is limited due to the variation in both length and order of the complicated protein sequences. Nevertheless, there are some proteins, which may be simultaneously existing at, or moving between, two or more different subcellular locations. Chou et al. have developed some user-friendly predictor, which handle multiple location problems of proteins in various organisms. For example: iLoc-Euk (Chou et al., 2011); Euk-mPLoc 2.0 (Chou and Shen, 2010a); Plant-mPLoc (Chou

* Corresponding author.
E-mail addresses: asif@pieas.edu.pk (A. Khan), abdulmajiid@pieas.edu.pk (A. Majid), maqsood.hayat@gmail.com (M. Hayat).

1476-9271/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2011.05.003

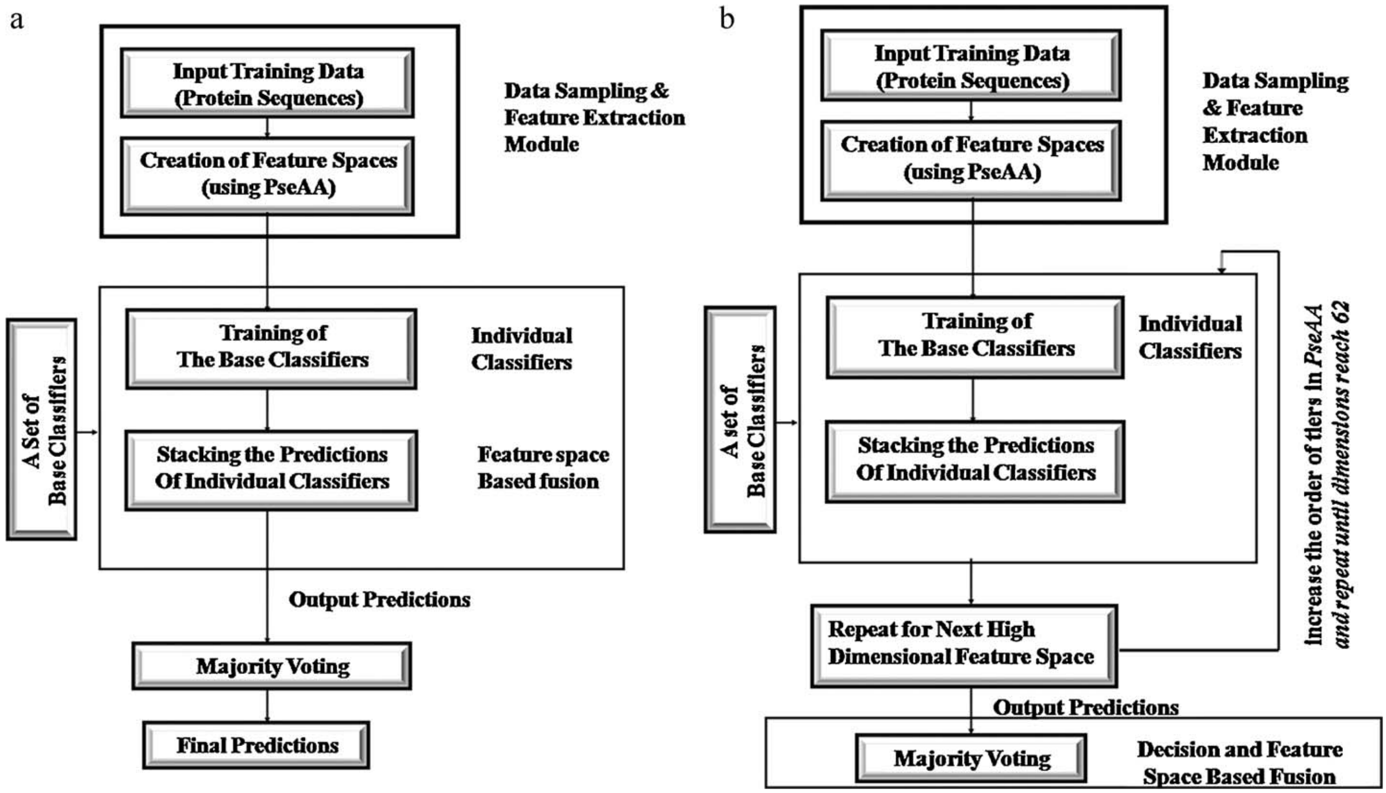

Fig. 1. (a) Framework of the proposed *IE* system for predicting the protein subcellular localizations. (b) Framework of the proposed *CE-PLoc* system. It is to be noted that in case of $CH00_{12}$ dataset, *DC* features of fixed dimensions (reduced using kPCA) are also provided to the set of base classifiers.

and Shen, 2010b); Cell-PLoc (Chou and Shen, 2008); and Cell-PLoc 2.0 (Chou and Shen, 2010c) are useful predictors. In a recent work (Chou, 2011), a comprehensive review regarding the development of useful predictors for subcellular locations, is provided. The steps that must be taken into account are: (i) select among the existing valid benchmark dataset for the predictor training and testing; (ii) express the protein samples mathematically, which can accurately reflect their intrinsic relation with the attribute to be predicted; (iii) establish or build a powerful and effective algorithm to operate the prediction; and (iv) perform holdout or cross-validation tests appropriately to objectively evaluate the predicted accuracy of the predictor.

It is well known that combining a variety of classifiers often yields superior accuracy than the individual classifiers. Researchers have proposed a range of ensemble strategies in the field of computational biology (Daun et al., 2008). An ensemble of *CDC* classifier using amphiphilic pseudo amino acid composition (*PseAAC*) has been developed by Chou and Shen (2006b). An ensemble classifier "*Hum-PLoc*" is built by fusing individual classifiers produced through *K*-nearest neighbor rule (Chou and Shen, 2006a). Individual classifiers are trained on the hybridized features of gene ontology and amphiphilic *PseAAC*. *SVM* based new ensemble tools are developed for predicting the subcellular localizations of Gram-negative (Shen and Chou, 2010), lysine acetylation sites (Xu et al., 2010), and apoptosis proteins (Zhang et al., 2009), to perform cancer classification (Anand and Suganthan, 2009), and to carry out protein structural classification (Anand and Suganthan, 2009). Similarly, an idea of "*unite and conquer*" for combining classifiers is proposed (Shen and Burger, 2007). Genetic algorithm based optimized ensemble classifier (*GAOEC*) (Guo and Gao, 2008).

However, most of the above-mentioned ensembles do not combine classifiers that are individually trained on feature spaces of varying dimensions as well as exploit different learning mechanisms. In this regard, our *CE-PLoc* can be considered as a one-step forward approach, which attempts to not only exploit the variation in feature spaces but also to exploit diversity in the decision spaces. Variation in the feature spaces is obtained using *PseAAC*, while diversity in decision space is obtained by using different learning mechanisms. Our comprehensive analysis demonstrates that the fusion of individual classifiers with good diversity is more effective as compared to individual classifiers and those ensemble systems that either exploit diversity in feature space (Chou and Shen, 2006b) or in decision space (Shen et al., 2007). The performance of *CE-PLoc* approach is reported using both jackknife and independent dataset tests. Comparative analysis shows that our *CE-PLoc* approach outperforms the approaches proposed by Chou and Shen (2006b) and Fayyaz et al. (2007) for 14 classes dataset. Further, improved prediction is also observed as compared to the existing approaches (Chou and Cai, 2002; Gao et al., 2005; Chou, 2001; Shi et al., 2007, 2008) for 12 classes dataset.

2. Proposed methodology

The basic architecture of the proposed scheme for predicting the subcellular localization is shown in Fig. 1(a) and (b). The proposed strategy of fusion of classifiers is composed of two stages. In the first stage of the proposed *CE-PLoc* scheme, individual ensemble (*IE*) classifiers are produced by selecting a learning mechanism, which exploits variation in feature spaces. This step is repeated by selecting other learning mechanism. In the second stage, a combined ensemble (*CE*) is developed by fusing all *IE* classifiers using entire-pool voting scheme, thus exploiting the diversity in decision spaces. The *CE* classifier built in this way is expected to be more effective as compared to the *IE* classifier.

Four diverse types of learning mechanisms are employed for *PseAAC* feature strategy: (1) nearest neighbor (*NN*); (2) probabilistic neural network (*PNN*); (3) *SVM*, and (4) *CDC*. All these learning mechanisms, except *SVM*, are inherently based on the

Table 1
Number of proteins sequences in each subcellular locations for two datasets.

Sr. #	Subcellular locations	$CH00_{12}$		$CH03_{14}$	
		$CH00_{12}$–*J*	$CH00_{12}$–*I*	$CH03_{14}$–*J*	$CH03_{14}$–*I*
1	Chloroplast	145	112	316	855
2	Cytoplasm	571	761	1113	186
3	Cytoskeletons	34	19	249	131
4	End. reticulum	49	106	289	136
5	Extracell	224	95	393	1252
6	Golgi appar.	25	4	90	41
7	Lysosome	37	31	123	57
8	Mitochondria	84	163	389	762
9	Nucleus	272	418	399	914
10	Peroxisome	27	23	147	84
11	Vacule	24	–	86	17
12	Plasma memb.	699	762	–	–
13	Cell memb.	–	–	69	24
14	Cell wall	–	–	71	35
15	Centriole	–	–	65	4
Total locations		2191	2494	3799	4498

Note that the vacuole and chloroplast proteins exist only in plants. Here xx–*J* and xx–*I* denote data set for jack-knife and independent tests, respectively.

proximity-based classification. *SVM* classifier constructs a separation boundary with maximum margin of separation to classify data samples. *CDC* is reported to perform well on this problem by exploiting the variation in the *PseAAC* based features of a protein sequence (Chou and Shen, 2006b). Similarly, *NN* has been reported to perform well on difficult classification tasks regarding protein sequences (Chou and Shen, 2006b; Jia et al., 2007; Khan et al., 2008). The neural network approach based *PNN* classifier utilizes Bayes theory to estimate the likelihood of sample being part of a learned category/class (Duda et al., 2001).

To validate our ensemble approach, we have considered another dataset of 12 human protein subcellular locations and the prediction results are reported for two schemes. In the first scheme, we took the same four base classifiers for *PseAAC* feature strategy. In the second scheme, the performance of ensemble approach is investigated by exploiting the combination of feature extraction strategies, i.e. dipeptide composition (*DC*) and amphiphilic *PseAAC*. For dipeptide features, we added more base learners, i.e. back propagation neural network (*BPNN*) and learning vector quantization based neural network (*LVQNN*) are also used from (*ANN* toolbox (MATLAB7.0). Reduced coulomb energy (*RCE*) (Duda et al., 2001) and evidence theoretic *kNN* (*ET-kNN*) (Zouhal and Denoeux, 1998) have also been employed as base classifiers. Further, Kernel Principal Component Analysis (*KPCA*) technique is utilized to reduce the dimensionality of *DC* features.

2.1. Datasets

We have adopted the same benchmark datasets (termed as $CH00_{12}$ and $CH03_{14}$) that have been reported by Chou (2000) and Chou and Cai (2003), respectively. These datasets have been originally extracted from the SWISS-PROT databank (Bairoch and Apweiler, 1997) release #35.0 and #40.0. To reduce redundancy and homology bias in the protein sequences, both datasets have been passed through window screening (Chou, 2000; Chou and Cai, 2003). For a specific subcellular location, protein sequence has maximum similarity <80% and <25% for $CH00_{12}$ and $CH03_{14}$ datasets, respectively. The jackknife test dataset, in $CH03_{14}$, mostly contains the protein sequences that have been included in previous release 35.0 of SWISS-PROT databank. However, for the independent dataset test, in $CH03_{14}$, new protein sequences have been extracted from the new release #40.0. The main features of both $CH00_{12}$ and $CH03_{14}$ datasets are the high difference of similarity (80% and 25%) and large number of cell compartments, i.e. 12 and 14, respectively. The detail of protein sequences in both datasets is provided in Table 1. The scope of $CH03_{14}$ dataset is larger as compared to $CH00_{12}$, not only due to more number of classes but also due to the inclusion of new proteins, which have not been present in the previous release of SWISS-PROT databank. Further, $CH03_{14}$ dataset is more generalized due to higher stringent condition of sequence similarity.

2.2. Feature extraction

In this section, we will explain briefly *PseAAC* and *DC* feature extraction strategies.

2.2.1. Amphiphilic PseAAC

The pseudo amino acid composition (*PseAAC*) based features have been extensively used (Du et al., 2009; Zhang et al., 2008; Sahu and Panda, 2010). Using *AAC*, much of the important hidden information in protein sequences is lost. Thus, to avoid losing this hidden information, the *PseAAC* has been introduced (Chou, 2001, 2005) for protein sequence representation. *PseAAC* has thus been employed for various predictions of protein problems such as predicting *GPCR-type* (Xiao et al., 2011), protein subcellular localization (Li and Li, 2008), out membrane proteins (Lin, 2008), cyclin proteins (Mohabatkar, 2010), enzyme family class (Qiu et al., 2010), and protein structural class (Sahu and Panda, 2010). The 16 different forms of *PseAAC* are those that are able to incorporate the functional domain information, *GO* (gene ontology) information, Cellular Automaton image information, sequential evolution information, among many others, for more details see a recent comprehensive review (Chou, 2009). *PseAAC* can be used to represent a protein sequence with a discrete model yet without completely losing its sequence-order information (Chou, 2005; Kavousi et al., 2010; Naveed and Khan, 2011; Rehman and Khan, 2011). Let us assume that we have N proteins feature vectors ($\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N$) derived from protein dataset. Each $\mathbf{P}_i$ belong to one of the V classes with labels $Q_1, Q_2, \ldots, Q_V$. A kth subcellular protein feature vector from a class V can be expressed as:

$$\mathbf{P}_v^k = \left[p_{v,1}^k, p_{v,2}^k, \ldots, p_{v,20}^k, \ldots, p_{v,\Phi}^k\right]^{\mathbf{T}} \quad (1)$$

where $p_{v,1}, p_{v,2}, \ldots, p_{v,20}$ are the frequencies of occurrence of 20 amino acids. The elements $p_{v,21}, p_{v,22}, \ldots, p_{v,\Phi}$ are the 1st-tier to $(\xi - 1)$-tier correlation factors of an amino acids in the protein chain based on two indices of hydrophobicity and hydrophilicity (Chou and Cai, 2003; Chou and Shen, 2006b). 1st-Tier correlation factors represent the sequence order correlation between all the first most contiguous residues along a protein chain, while $(\xi - 1)$-tier represent the same between all the $\xi - 1$ most contiguous residues. The aim of *PseAAC* feature strategy is to better reflect the sequence order and length effect of a protein sequence compared to conventional amino acid composition.

2.2.2. Dipeptide composition (DC)

DC features are also used for the training of individual classifiers. The *DC* is very useful to predict the contents of protein secondary structure (Tan et al., 2007). It represents the occurrence frequency of each two adjacent amino acid residues. It is used to describe the global information about each protein sequence in the form of 400-dimensional (400-D) feature vector. An advantage of *DC* over amino acid composition is that it uses some sequence-order information. The 400-D features of each protein are calculated as:

$$Fdip(i) = \left(\frac{total\,number\,of\,dip(i)}{total\,number\,of\,all\,dipeptides} \right) \qquad (2)$$

where $Fdip(i)$ shows the fraction of ith dipeptide out of 400 dipeptides. In our current work, we reduce the dimensions of 400-D features by using *KPCA* (Duda et al., 2001). The goal of the *KPCA* is the same as that of the classical *PCA*. However, this method is capable of finding the non-linear combinations too. The details of *KPCA* are available in Schölkopf et al. (1998) and Suykens et al. (2003). Mathematically, objective of *KPCA* can be formulated as: $\max\limits_{w}\sum_{i=1}^{N}[w^T(\varphi(x_k) - \mu_\varphi]^2$, where $\mu_\varphi = \left(\frac{1}{N}\right)\sum_{k=1}^{N}(\varphi(x_k)$ is used for centering the data in the feature space and $\varphi(.) : R^n \rightarrow R^{n_h}$ is the mapping function to high-dimensional feature space. This formulation helps to find the maximum variance of the projected variables for the given N data points while keeping the norm of w small. By taking the conditions of optimality from the Lagrangian constrained optimization, $w = \sum_{k=1}^{N}[\alpha_k(\varphi(x_k) - \mu_\varphi]$, and defining eigenvalue problem, $\Omega_c\alpha = \lambda\alpha$, we can compute values of centered kernel matrix Ω_c. In this work, we have selected the optimal width δ of *RBF* kernel to be four and the dimension reduction of 400-D vectors is performed by selecting 30 eigenvectors corresponding to largest eigenvalues.

2.3. Base learners

In this section, we will describe five most important base learners that we have adopted for training in the proposed scheme.

2.3.1. Nearest neighbor (NN)

The proximity based *NN* classifies objects based on closest training examples in the feature space. Owing to its simplicity and effective performance, *NN* is used extensively in protein prediction related studies. The training objects are first mapped into multidimensional feature space and the space is partitioned into regions by class labels of the training samples. A point in this space is then assigned to class *C* if it is the closest class label among training samples. Usually, Euclidean distance is used as a metric for measuring proximity. For a protein sequence **P** under question, how can one predict its class label? According to the *NN* principle, we have to find generalized distance between **P** and $\mathbf{P}_i$:

$$S(\mathbf{P}, \mathbf{P}_i) = 1 - \left(\frac{\mathbf{P} \cdot \mathbf{P}_i}{\left\|\mathbf{P}\right\| \left\|\mathbf{P}_i\right\|} \right) \quad (i = 1, 2, \ldots N) \qquad (3)$$

where $\mathbf{P} \cdot \mathbf{P}_i$ is the dot product of vectors **P** and $\mathbf{P}_i$, and $||\mathbf{P}||$ and $||\mathbf{P}_i||$ are respectively their modulus.

Then the minimum of the generalized distances is computed as:

$$S(\mathbf{P}, \mathbf{P}_k) = Min\{S(\mathbf{P}, \mathbf{P}_1), S(\mathbf{P}, \mathbf{P}_2) \ldots, S(\mathbf{P}, \mathbf{P}_N)\} \qquad (4)$$

The protein sequence **P** under question is then assigned the category corresponding to the training protein $\mathbf{P}_k$.

2.3.2. Covariant discriminant classifier (CDC)

The standard protein vector for class Q_v is given by:

$$\bar{\mathbf{P}}_v = [\bar{p}_{v,1}, \bar{p}_{v,2}, \ldots \bar{p}_{v,20}, \ldots \bar{p}_{v,\Phi}]^{\mathbf{T}} \qquad (5)$$

where $\bar{p}_{v,i} = \frac{1}{N_v}\sum_{k=1}^{N_v} p_{v,i}^k \quad (i = 1, 2, \ldots, \Phi)$ and N_v represents the number of samples in class v.

Now *CDC* is based on the computation of Mahalanobis distance D_{Mh}^2 from the mean of each class in feature space. The similarity between query protein **P** and each of standard protein vector $\mathbf{P}_v$ is defined by the following covariant discriminant function:

$$\mathcal{M}(\mathbf{P}, \bar{\mathbf{P}}_v) = D_{Mh}^2(\mathbf{P}, \bar{\mathbf{P}}_v) + \ln\left|C_v\right| \quad (v = 1, 2, \ldots, V) \qquad (6)$$

$$D_{Mh}^2(\mathbf{P}, \bar{\mathbf{P}}_v) = (\mathbf{P} - \bar{\mathbf{P}}_v)^{\mathrm{T}} C_v^{-1} (\mathbf{P} - \bar{\mathbf{P}}_v) \qquad (7)$$

C_v is the covariance matrix for the subcategory v and is given by:

$$c_v = \begin{bmatrix} c_{1,1}^v & c_{1,2}^v & \cdots & c_{1,\Phi}^v \\ c_{2,1}^v & c_{2,2}^v & \cdots & c_{2,\Phi}^v \\ \vdots & \vdots & \ddots & \vdots \\ c_{\Phi,1}^v & c_{\Phi,2}^v & \cdots & c_{\Phi,\Phi}^v \end{bmatrix} \qquad (8)$$

where individual elements are given by $c_{i,j}^v = \frac{1}{N_v - 1}\sum_{k=1}^{N_v}(p_{v,i}^k - \bar{p}_{v,i})(p_{v,j}^k - \bar{p}_{v,j}) \quad (i, j = 1, 2, \ldots, \Phi), C_v^{-1}$ is the inverse of covariance matrix, and $|C_v|$ is the determinant of covariance matrix. All the computed Mahalanobis distances are compared and the one with the minimum distance found is the respective target class Q_τ of the query protein **P**:

$$\mathbf{m}(\mathbf{P}, \bar{\mathbf{P}}_\tau) = Min\{\mathbf{m}(\mathbf{P}, \bar{\mathbf{P}}_1), \mathbf{m}(\mathbf{P}, \bar{\mathbf{P}}_2), \ldots, \mathbf{m}(\mathbf{P}, \bar{\mathbf{P}}_V)\}$$
$$\tau = 1, 2, \ldots, \text{or}\, V \qquad (9)$$

2.3.3. Probabilistic neural networks (PNN)

PNN posses the computational power and flexibility attributes similar to that of back-propagated neural networks, while at the same time has the simplicity and transparency of traditional statistical classification approaches. It is based on Bayes theory and estimates the likelihood of a sample being part of a learned category. *PNN* consists of four layers; an input, pattern, summation, and decision layers. The input node has *N* nodes each corresponding to one independent variable. These input nodes are then fully connected to the *M* nodes of the pattern layer. One particular node in pattern layer corresponds to one training object only. An input vector $\mathbf{P}_i$ is processed by pattern node j using an activation function, whose most popular form is the exponential one:

$$u_{ij} = \exp\left(\frac{-\left\|\mathbf{P}_j - \mathbf{P}_i\right\|^2}{\sigma^2} \right) \qquad (10)$$

where u_{ij} denotes the output of the pattern node j and δ is a smoothing factor that controls the width of the activation function. In the current work, the optimal value for the spread δ of *PNN* is found to be 0.9.

If the $\left\|\mathbf{P}_j - \mathbf{P}_i\right\|$ distance between the input vector $\mathbf{P}_i$ and the vector $\mathbf{P}_j$ of the pattern node j increases, similarity between the two data vectors decreases and vice versa. The results of the pattern nodes are provided to the summation layer, which contains v competitive nodes each corresponding to one class. Now each summation node v is connected to the pattern nodes that are associated to training objects of class v. For an input vector $\mathbf{P}_i$, the summation node k receives the outputs of the associated pattern nodes for producing an output:

$$f_v(P_i) = \frac{1}{M_v}\sum_{\forall P_i \in Q_v} u_{ij} \tag{11}$$

where Q_v is the label of the class corresponding to the summation node v, while N_v is the number of training objects belonging to this class. If we assume that all data vectors are normalized then, the previous equation can be formulated as:

$$f_v(P_i) = \frac{1}{N_v}\sum_{\forall P_i \in Q_v} \exp\left(\frac{(P_j P_i^T - 1)}{\sigma^2}\right) \tag{12}$$

The outputs of the summation layer can be expressed in terms of the posterior class membership probabilities:

$$P(Q_i = v\,\big|\mathbf{P}_i) = \frac{f_v(\mathbf{P}_i)}{\sum_{v=1}^{V} f_v(\mathbf{P}_i)} \tag{13}$$

Using the above equation, a classification rule is incorporated at the decision layer for assigning the input vector to $\mathbf{P}_i$ to a particular class. The straightforward approach is to select the class whose $P(v|\mathbf{P}_i)$ is maximum.

2.3.4. Support vector machine (SVM)

SVM model is well documented in the literature of statistical learning theory. It is being extensively used in the field of computational biology (Xu et al., 2010; Yeh and Mao, 2006; Zhang et al., 2009; Li et al., 2006). *SVM* models find a decision surface that has maximum distance to the closest points in the training set. The classification problem is solved as a quadratic optimization problem. The training principle of *SVM* is to find an optimal linear hyperplane such that the classification error for new test samples is minimized. For linearly separable sample points, hyperplane is determined by maximizing the distance between the support vectors. For elaboration, we use same basic notations of *SVMs* theory to establish the equation of hyperplane. For a linearly separable data of N training pairs (x_i, y_i) the function of a decision surface S is defined as:

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i x_i^T.x + bias \tag{14}$$

where the coefficient $\alpha_i > 0$ is the Langrange multiplier in an optimization problem. A pattern vector x_i that corresponds to $\alpha_i > 0$ is called a support vector. The $f(x)$ is independent of the dimension of the feature space. In order to find an optimal hyperplane surface S for non-separable patterns, solution of the following optimization problem is sought:

$$\Psi(w,\zeta) = \frac{1}{2}w^T w + C\sum_{i=1}^{N}\zeta_i$$

subject to the condition $y_i(w^T\Psi(x_i) + bias) \geq 1 - \zeta_i, \quad \zeta_i \geq 0.$ (15)

where C is the penalty parameter of the error term $\sum_{i=1}^{N}\zeta_i$. It represents the cost of constraint violation of those data point, which occur on the wrong side of the decision boundary, and $\Psi(x)$ is the nonlinear mapping. The weight vector w minimizes the cost function term $w^T w$.

The nonlinear input data is mapped to higher dimension through a mapping function $\Psi(x)$ such that $\Psi : R^N \to F^M$, $M \gg N$. Each point in the new feature space is defined by a kernel function $K(x_i, x_j) = \Psi(x_i)\cdot\Psi(x_j)$. The nonlinear decision surface S can now be constructed by a new function $f(x)$ as:

$$f(x) = \sum_{i=1}^{N_S}\alpha_i y_i K(x_i, x) + bias = \sum_{i=1}^{N_S}\alpha_i y_i \Psi(x_i)\cdot\Psi(x) + bias \tag{16}$$

where N_S is the number of support vectors. Mathematically, Radial Basis Function (*RBF*) kernel function is defined as: $K(x_i, x_j) = \exp(-\left\|x_i - x_j\right\|^2/(2\sigma)^2)$, where parameter σ shows the width of Gaussian function.

To develop IE^{SVM} classifiers for multi-classification problem, one-vs-all strategy is adopted in *LIBSVM* software (Chang and Lin, 2008). This software solves *SVM* problem using Nonlinear Quadratic Programming technique. During parameters optimization, average accuracy of *SVM* models is maximized. Using jackknife test, which is considered as the most effective one (Khan et al., 2005; Majid et al., 2006), optimal values of cost function C and kernel width σ are found to be 100.0 and 0.00625, respectively. However, to evaluate the performance of the *SVM* models for 4498 independent protein data samples, first *SVM* model is developed for 3799 training protein samples. The optimal values of C and σ are computed to be 8.01 and 0.0625, respectively.

2.3.5. Evidence theoretic kNN

The mathematical details of *ET-kNN* (Zouhal and Denoeux, 1998) is based on the Dempster–Shafer theory (Denoeux, 1995). In this rule, each neighbor pattern to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that pattern. Due to this evidence, the mass of each basic belief is assigned to each subset concerned. Such masses are obtained for each of K-nearest neighbors of the pattern under consideration and aggregated using Dempster's combination rule. A decision value is assigned to a pattern of the class with maximum credibility. In the *ET-kNN* rule, it is not addressed how to optimally select parameters. However, optimal parameter values can be found by minimizing an error function (Zouhal and Denoeux, 1998). The *ET-kNN* rule obtained through this optimization scheme leads to a substantial improvement in classification accuracy.

2.4. IE development using PseAAC

In the first stage, individual ensembles (*IE*) are produced by selecting a base classifier and exploiting variation in feature spaces. Variation in feature spaces is exploited using amphiphilic *PseAAC* by varying dimensions from 20 to 62, i.e. $\Phi = 20 + 2(i - 1)$, where $i = 1, 2, \ldots, \xi$. Here, $\xi = 22$, represents the number of *IE* classifiers. The ensemble *IE* has ξ classifiers and thus their individual predictions R_i can be expressed as:

$$\{R_1, R_2, R_3 \ldots, R_\xi\} \in \{Q_1, Q_2, Q_3, \ldots, Q_V\} \tag{17}$$

Now the *IE* based voting mechanism for the kth feature vector of protein can be formulated as:

$$Z_j^{IE} = \sum_{i=1}^{\xi} w_i \Delta(R_i, Q_j), \quad j = 1, 2, \ldots V \tag{18}$$

where w_i represents weight factor. Here, for simplicity, its value is set to unity and $\Delta(R_i, Q_j) = \begin{cases} 1 & \text{if} \quad R_i \in Q_j \\ 0 & \text{otherwise} \end{cases}$.

Finally, the class of the query protein is assigned by *IE* to the class γ that obtains maximum votes:

$$Z_\gamma^{IE} = Max\left\{Z_1^{IE}, Z_2^{IE}, \ldots Z_V^{IE}\right\} \tag{19}$$

2.5. CE development using PseAAC

In 2nd stage, the diverse decision spaces generated by *IE* classifiers are combined. In this way, the shortcoming/overfitting of one type of *IE* ensemble may be circumvented by the other types of *IE* ensembles.

We first describe the entire-pool voting scheme for developing *CE*. Let $l = 1, 2, 3, \ldots, L$ represents the number of different base learners. We compute the votes of each class for *CE* ensemble as:

$$Z_j^{CE} = \sum_{i=1}^{L*\xi} w_i \Delta(\boldsymbol{R}_i, Q_j), \quad j = 1, 2, \ldots V \tag{20}$$

The predicted class τ by the *CE* classifier will be decided by using the *Max* function:

$$Z_\tau^{CE} = Max\left\{Z_1^{CE}, Z_2^{CE}, \ldots Z_V^{CE}\right\} \tag{21}$$

Similarly, we also perform the fusion of the *IE* classifiers based on sub-pool voting mechanism with the aim of exploiting the variation in the decision spaces of the *IE* classifiers. Using Eqs. (4) and (8), respectively, we thus generate and combine *IE* ensembles to develop *CE*:

$$Z_j^{CE} = \sum_{l=1}^{L} w_l Z_j^{IE_l} \quad j = 1, 2, \ldots V \tag{22}$$

Weight factor the value of w_l is set to unity. However, the performance of the proposed *CE* could be further improved, if useful weight strategy like the one we proposed by Fayyaz et al. (2007) is adopted. In jackknife test, if a tie occurs for a query protein; then decision of the highest performing ensemble (IE^{SVM}) is considered. If the highest performing ensemble also delivers a tie, then the vote of the 2nd highest performing ensemble (IE^{NN}) is considered. However, we have experimentally observed that tie occurs rarely, i.e. 4 out of 3799.

2.6. Evaluation methods and quality measures

Holdout and cross-validation tests are widely used to assess the performance of a predictor in practical applications. Cross-validation methods can be categorized into three types: self-consistency, independent dataset, and jackknife test (Chou and Zhang, 1995). However, among the three cross-validation methods, the jackknife test is considered the most rigorous on owing to its ability of yielding a unique result for a given benchmark dataset. It has thus been increasingly used by investigators to examine the performance of various predictors (Hayat and Khan, 2011; Kandaswamy et al., 2011; Mohabatkar, 2010; Hu et al., 2011; He et al., 2010; Afridi et al., 2011). During the jackknife test, each of the proteins in the dataset is in turn singled out as a test sample and all the remaining samples are used to train the classifier.

In this work, we have reported the performance of our approach using five quality measures as: accuracy (*Acc*), sensitivity (*Sn*), specificity (*Sp*), Mathew's correlation coefficient (*MCC)*, and *Q*-statistic. *MCC* measure accounts for both over-predictions and under-predictions (Matthews, 1975). This measure is a discrete version of Pearson's correlation coefficient taking values in the interval $[-1, 1]$. Specificity estimates the precision of the predictor. *Q*-statistic is used to measure the diversity among individual classifiers in ensemble. For protein sequences of *i*th subcellular location in a cell, these measures can be defined as:

$$Acc\,(i) = \left(\frac{p(i) + n(i)}{p(i) + n(i) + p(i) + o(i)}\right) \tag{23}$$

$$Sn\ \ (i) = \frac{p(i)}{p(i) + o(i)} \tag{24}$$

$$Sp\ \ (i) = \frac{p(i)}{p(i) + u(i)} \tag{25}$$

$$MCC(i) = \frac{p(i)n(i) - u(i)o(i)}{\sqrt{[p(i) + u(i)][p(i) + o(i)][n(i) + u(i)][n(i) + o(i)]}} \tag{26}$$

where $p(i)$, the number of observed positive samples, predicted as positive samples; $n(i)$, the number of observed negative samples, predicted as negative samples; $u(i)$, the number of observed negative samples, predicted as positive samples; $o(i)$, The number of observed positive samples, predicted as negative samples.

The numerical value of *Q*-statistic shows the error exhibited by individual classifiers in ensemble. For any two base classifiers C_i

Table 2
Accuracy obtained by individual classifiers for $CH03_{14}$ dataset using jackknife test.

Dim.	Classifier	*Acc* %	Classifier	*Acc* %	Classifier	*Acc* %	Classifier	*Acc* %
20	*CDC* (Φ1)	59.70	NN (Φ1)	77.86	PNN (Φ1)	78.10	*SVM* (Φ1)	76.26
22	*CDC* (Φ2)	62.83	NN (Φ2)	76.70	PNN (Φ2)	76.83	*SVM* (Φ2)	77.02
24	*CDC* (Φ3)	64.56	NN (Φ3)	77.18	PNN (Φ3)	77.07	*SVM* (Φ3)	78.15
26	*CDC* (Φ4)	65.73	NN (Φ4)	76.91	PNN (Φ4)	76.91	*SVM* (Φ4)	77.97
28	*CDC* (Φ5)	67.17	NN (Φ5)	76.41	PNN (Φ5)	76.41	*SVM* (Φ5)	78.49
30	*CDC* (Φ6)	67.78	NN (Φ6)	75.73	PNN (Φ6)	75.73	*SVM* (Φ6)	78.86
32	*CDC* (Φ7)	68.33	NN (Φ7)	75.55	PNN (Φ7)	75.44	*SVM* (Φ7)	78.78
34	*CDC* (Φ8)	68.04	NN (Φ8)	75.41	PNN (Φ8)	75.28	*SVM* (Φ8)	79.34
36	*CDC* (Φ9)	68.18	NN (Φ9)	75.04	PNN (Φ9)	74.86	*SVM* (Φ9)	78.89
38	*CDC* (Φ10)	68.50	NN (Φ10)	74.75	PNN (Φ10)	74.49	*SVM* (Φ10)	78.99
40	*CDC* (Φ11)	68.51	NN (Φ11)	74.91	PNN (Φ11)	74.68	*SVM* (Φ11)	79.31
42	*CDC* (Φ12)	68.78	NN (Φ12)	74.75	PNN (Φ12)	74.44	*SVM* (Φ12)	79.44
44	*CDC* (Φ13)	69.07	NN (Φ13)	74.39	PNN (Φ13)	74.04	*SVM* (Φ13)	79.07
46	*CDC* (Φ14)	69.57	NN (Φ14)	74.07	PNN (Φ14)	73.57	*SVM* (Φ14)	78.55
48	*CDC* (Φ15)	69.70	NN (Φ15)	73.57	PNN (Φ15)	73.12	*SVM* (Φ15)	78.55
50	*CDC* (Φ16)	69.28	NN (Φ16)	73.62	PNN (Φ16)	73.01	*SVM* (Φ16)	77.99
52	*CDC* ($\Phi\Phi$17)	69.04	NN (Φ17)	73.62	PNN (Φ17)	72.99	*SVM* (Φ17)	77.73
54	*CDC* (Φ18)	68.46	NN (Φ18)	73.15	PNN (Φ18)	72.46	*SVM* (Φ18)	77.31
56	*CDC* (Φ19)	68.07	NN (Φ19)	73.31	PNN (Φ19)	72.38	*SVM* (Φ19)	74.99
58	*CDC* (Φ20)	67.20	NN (Φ20)	73.36	PNN (Φ20)	72.20	*SVM* (Φ20)	73.91
60	*CDC* (Φ21)	66.96	NN (Φ21)	73.36	PNN (Φ21)	72.15	*SVM* (Φ21)	72.43
62	*CDC* (Φ22)	67.44	NN (Φ22)	72.86	PNN (Φ22)	71.54	*SVM* (Φ22)	72.02

Note: Individual base classifiers $C(\Phi_i)$ $(I = 1, 2, \ldots, \xi)$ are trained using feature spaces of dimensions; $\Phi_i = 20 + 2\ (I - 1)$ with $(I = 1, 2, \ldots, \xi)$. 'Dim.' represents dimension of feature space.

and C_j, the Q-statistic is defined (Kuncheva and Whitaker, 2003) as follows:

$$Q_{i,j} = \frac{ad - bc}{ad + bc} \tag{27}$$

where a and d represent the frequency of both classifiers making correct and incorrect predictions, respectively. However, b shows the frequency when first classifier is correct and second is incorrect; c is the frequency of second classifier being correct and first incorrect. The value of Q varies between −1 and 1. For statistically independent classifiers, the value of $Q_{i,j}$ is zero. Classifiers that try to recognize the same objects correctly have $Q > 0$, and those which commit errors on different objects render $Q < 0$.

The average value of Q-statistic among all pairs of the L base classifiers in *CE* ensemble is calculated as:

$$Q_{avg} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} Q_{i,k} \tag{28}$$

The positive value of Q_{avg} shows that classifiers recognize the same objects correctly. The positive value of Q_{avg} (<1) shows that base classifiers in the *CE* ensemble exhibit diversity.

3. Results

At first, we present the results of *CE* ensemble using simple *PseAAC* and composite *PseAAC* + *DC* features for $CH03_{12}$ dataset. However, for $CH03_{14}$ dataset, results are reported using *PseAAC* features alone. This is because using only *PseAAC* based features, the proposed *CE* ensemble is able to produce better results on $CH03_{14}$ dataset compared to the existing approaches.

3.1. Performance using $CH03_{14}$ dataset

3.1.1. Prediction of individual classifiers

The jackknife classification accuracy of individual classifiers is presented in Table 2. It is observed that *NN*, *PNN*, and *SVM* perform better compared to *CDC*. On the average, *SVM* classifier predicts relatively better than *NN* and *PNN*, except for first $SVM_{\Phi 1}$ classifier, where $NN_{\Phi 1}$ and $PNN_{\Phi 1}$ have an edge over $SVM_{\Phi 1}$. It is noticed that with the increase of dimensions of the feature space, the performance of *CDC* classifier first increases up to dimensions = 48 and then decreases. This degradation might be due to the curse of dimensionality. However, varying feature space does not highly affect the performance of *SVM* classifiers as compared to *NN* and *PNN*. In case of *NN* and *PNN*, accuracy of *NN* and *PNN* decreases with increased dimension. On the other hand, in case of independent dataset as shown in Table 3, *NN* classifier yields the highest accuracy. *CDC* classifier on average improves its performance with increase in dimensions, however, *SVM*'s performance drops with increase in dimensions.

The *MCC* values in case of jackknife test for the different learning mechanisms across the 14 classes of $CH03_{14}$ dataset are shown in Figs. 2–5 (for *CDC*, *NN*, *PNN*, and *SVM*, respectively). These figures show the behavior of the learning mechanism across the different classes when the dimensions of *PseAAC* based features are increased. As discussed earlier, the increase in *PseAAC* from 20 to 62 with a step of 2 generates the 22 classifiers for each learning mechanism. It can be observed from Fig. 2 that in case of *CDC*, for the first two classes; chloroplast and cytoplasm, the *MCC* first increases and then drops considerably when we increase the dimensions of *PseAAC*. However, an increasing trend is observed in case of classes 6 and 12; *Golgi Appar.* and *Cell Memb*. For the rest of the classes, the behavior is of mixed type. In case of *NN*, except for classes 1, 2 and 8, the general trend is a decline in *MCC* with increase in dimensionality. Similarly, in case of *PNN*, except for classes 2, 6, and 8, a decline is observed as we increase dimensions. In case of *SVM*, increase in *MCC* values is observed with increase in dimensions for classes 5, 6, and 12. While for class 2, an increase in *MCC* performance is first observed up to dimensions = 42 and then a decline occurs.

3.1.2. Prediction accuracy of IE

For jackknife test, it is observed from Table 4 that IE^{NN} predicts correctly 2984 out of 3799 sequences and 815 sequences are miss-predicted. Its overall accuracy comes out to be 78.55% for $CH03_{14}$ dataset. The results for IE^{NN} are compared with the results of IE^{CDC} reported by Chou and Shen (2006b) and Fayyaz et al. (2007) and improvement has been achieved in the accuracy by 8.35 and 7.4%, respectively. Similarly, the improvement of IE^{PNN} is also considerable; 7.95 and 7.0% respectively. In jackknife test, the most interesting result is the better prediction performance of IE^{SVM}. Therefore, if *SVM* learning mechanism is incorporated as a base classifier, then chances in the improved of *CE* enhances. The predicted accuracy of IE^{CDC} also validates the results reported by Chou and Shen (2006b) with slight difference in accuracy 0.7 and 0.15% in case of jackknife and independent dataset, respectively.

For independent dataset test, the overall prediction accuracies of the *IE* ensemble are validated on independent set containing 4498 protein sequences (see Table 4). The results, in this table, show IE^{NN} correctly predict 3774 sequences and only 724 sequences are incorrect to give an overall accuracy of 83.90%. This shows that IE^{NN} predicts 9.8 and 8.83% more accurately than IE^{CDC} (Chou and Shen, 2006b) and weighted IE^{CDC} (Fayyaz et al., 2007), respectively. Similarly, in IE^{PNN}, 9.2 and 8.23% increase in accuracy is obtained as compared to IE^{CDC} (Chou and Shen, 2006b) and weighted IE^{CDC} (Fayyaz et al., 2007), respectively. As explained earlier, for independent dataset test, the overall prediction of IE^{SVM} is not appreciable (43.35%).

3.1.3. Prediction accuracy of CE

The prediction accuracy of *CE* classifier is obtained using entire voting strategy. In jackknife test, Table 4 highlights the prediction accuracies of CE^{φ} and CE^{*} classifiers to be 82.44 and 83.99%, respectively. The performance of *CE* classifiers is higher than the best performing *IE* classifier, i.e. IE^{SVM} (80.86%). However, in independent test, CE^{φ} and CE^{*} classifiers correctly classifies 3918 and 3930 protein sequences out of 4498 to achieve overall accuracies of 87.11 and 87.33%, respectively. In independent test, due to the less accurate performance of IE^{SVM} with increasing dimension, the prediction of only the first three individual *SVM* classifiers are incorporated in the development of CE^{*}. This table highlights better prediction of CE^{*} as compared to CE^{φ}.

3.1.4. Performance of CE using different measures

In Table 5, the performance of *CE* is reported in terms of different quality measures, i.e. accuracy (*Acc*), Mathew's correlation coefficient (*MCC*), sensitivity (*Sn*), and specificity (*Sp*). It is observed that using jackknife test, *CE* yields relatively higher *Acc* and *MCC* values for $CH03_{14}$ dataset as compared to $CH00_{12}$. In case of $CH03_{14}$ dataset, most of the subcellular locations have *Acc* values in range of 0.8, except for endoplasmic reticulum and Golgi apparatus (>0.9). The *MCC* values of IE^{CDC}, IE^{NN}, IE^{PNN}, and IE^{SVM} classifiers for $CH03_{14}$ dataset are presented in Supplementary Tables 1–4. Table 5 also shows that in case of $CH03_{14}$ data, average specificity (0.987) is higher than that of average sensitivity (0.863).

Average *MCC* values of *IE* and *CE* ensembles are presented in Table 4. It is observed that average *MCC* values of *CE* (0.80 and 0.81) are higher than average *MCC* values of *IEs* for both $CH00_{12}$ and $CH03_{14}$ dataset, respectively using for jackknife test. In this table, average *Q*-statistics of *IEs* are in the range of 0.63–0.94 and 0.92–0.93 using jackknife test for $CH00_{12}$ and $CH03_{14}$ dataset, respectively. The *Q*-statistics of *CEs* for the same

Table 3
Accuracy rate obtained by individual classifiers for $CH03_{14}$ dataset using independent test.

Dim.	Classifier	*Acc* %	Classifier	*Acc* %	Classifier	*Acc* %	Classifier	*Acc* %
20	*CDC* (Φ_1)	59.93	*NN* (Φ_1)	83.17	*PNN* (Φ_1)	78.10	*SVM* (Φ_1)	81.52
22	*CDC* (Φ_2)	63.16	*NN* (Φ_2)	82.37	*PNN* (Φ_2)	76.83	*SVM* (Φ_2)	79.54
24	*CDC* (Φ_3)	66.03	*NN* (Φ_3)	81.88	*PNN* (Φ_3)	77.07	*SVM* (Φ_3)	76.85
26	*CDC* (Φ_4)	67.43	*NN* (Φ_4)	81.70	*PNN* (Φ_4)	76.91	*SVM* (Φ_4)	67.25
28	*CDC* (Φ_5)	68.87	*NN* (Φ_5)	81.48	*PNN* (Φ_5)	76.41	*SVM* (Φ_5)	61.72
30	*CDC* (Φ_6)	69.94	*NN* (Φ_6)	80.90	*PNN* (Φ_6)	75.73	*SVM* (Φ_6)	57.51
32	*CDC* (Φ_7)	70.47	*NN* (Φ_7)	80.46	*PNN* (Φ_7)	75.44	*SVM* (Φ_7)	53.49
34	*CDC* (Φ_8)	71.39	*NN* (Φ_8)	79.83	*PNN* (Φ_8)	75.28	*SVM* (Φ_8)	50.47
36	*CDC* (Φ_9)	71.76	*NN* (Φ_9)	79.99	*PNN* (Φ_9)	74.86	*SVM* (Φ_9)	48.11
38	*CDC* (Φ_{10})	71.87	*NN* (Φ_{10})	80.06	*PNN* (Φ_{10})	74.49	*SVM* (Φ_{10})	46.44
40	*CDC* (Φ_{11})	72.27	*NN* (Φ_{11})	79.39	*PNN* (Φ_{11})	74.68	*SVM* (Φ_{11})	44.51
42	*CDC* (Φ_{12})	72.94	*NN* (Φ_{12})	79.48	*PNN* (Φ_{12})	74.44	*SVM* (Φ_{12})	43.04
44	*CDC* (Φ_{13})	73.41	*NN* (Φ_{13})	79.52	*PNN* (Φ_{13})	74.04	*SVM* (Φ_{13})	42.02
46	*CDC* (Φ_{14})	73.45	*NN* (Φ_{14})	79.59	*PNN* (Φ_{14})	73.57	*SVM* (Φ_{14})	40.93
48	*CDC* (Φ_{15})	74.10	*NN* (Φ_{15})	79.66	*PNN* (Φ_{15})	73.12	*SVM* (Φ_{15})	39.91
50	*CDC* (Φ_{16})	73.92	*NN* (Φ_{16})	79.52	*PNN* (Φ_{16})	73.02	*SVM* (Φ_{16})	38.93
52	*CDC* (Φ_{17})	73.72	*NN* (Φ_{17})	79.10	*PNN* (Φ_{17})	72.99	*SVM* (Φ_{17})	38.61
54	*CDC* (Φ_{18})	73.74	*NN* (Φ_{18})	79.46	*PNN* (Φ_{18})	72.46	*SVM* (Φ_{18})	38.04
56	*CDC* (Φ_{19})	74.21	*NN* (Φ_{19})	79.08	*PNN* (Φ_{19})	72.39	*SVM* (Φ_{19})	37.46
58	*CDC* (Φ_{20})	73.92	*NN* (Φ_{20})	78.94	*PNN* (Φ_{20})	72.20	*SVM* (Φ_{20})	36.84
60	*CDC* (Φ_{21})	73.50	*NN* (Φ_{21})	78.52	*PNN* (Φ_{21})	72.15	*SVM* (Φ_{21})	36.59
62	*CDC* (Φ_{22})	73.255	*NN* (Φ_{22})	78.35	*PNN* (Φ_{22})	71.54	*SVM* (Φ_{22})	35.90

Note: Individual base classifiers $C(\Phi_i)$ ($I = 1, 2, \ldots, \xi$) are trained on feature spaces of dimensions; $\Phi_i = 20 + 2\,(I - 1)$ with ($I = 1, 2, \ldots, \xi$).

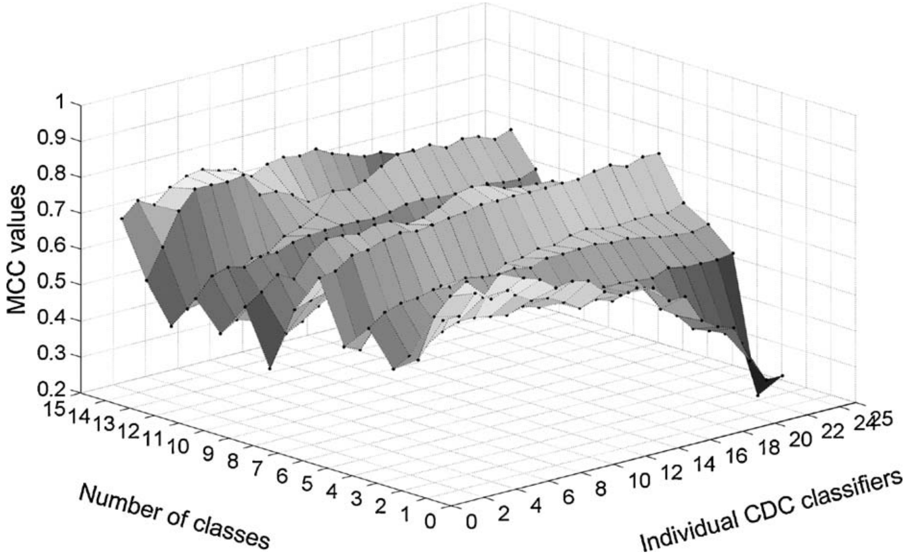

Fig. 2. *MCC* values for each of the 22 individual *CDC* classifiers using jackknife test for CH0314 dataset.

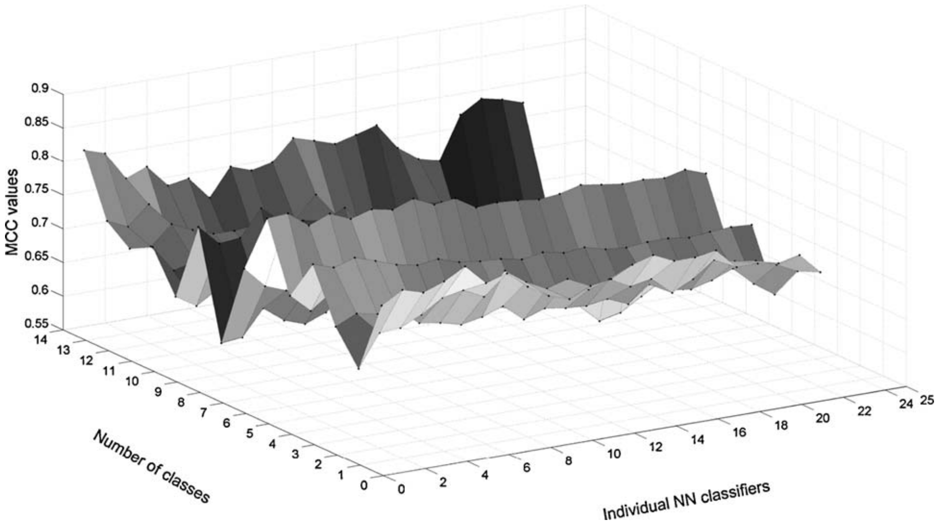

Fig. 3. *MCC* values for each of the 22 individual *NN* classifiers using jackknife test for $CH03_{14}$ dataset.

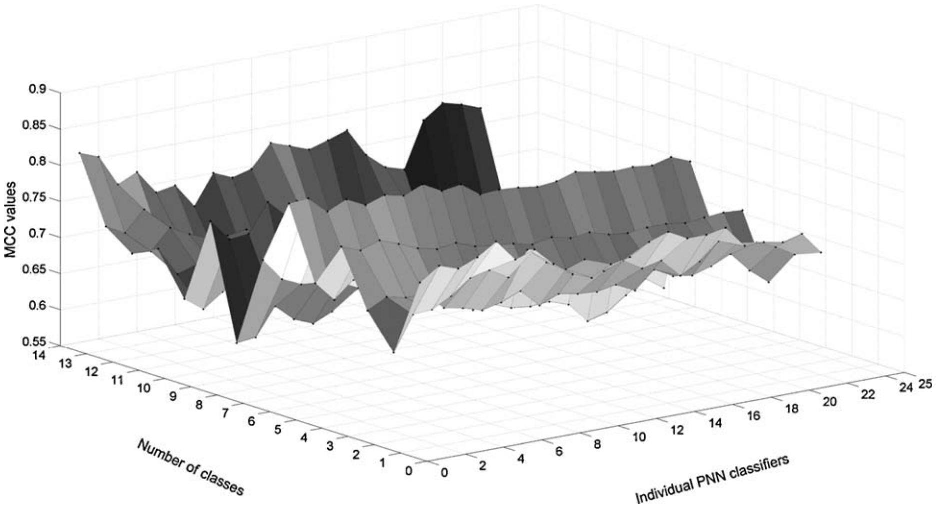

Fig. 4. *MCC* values for each of the 22 individual *PNN* classifiers using jackknife test for $CH03_{14}$ dataset.

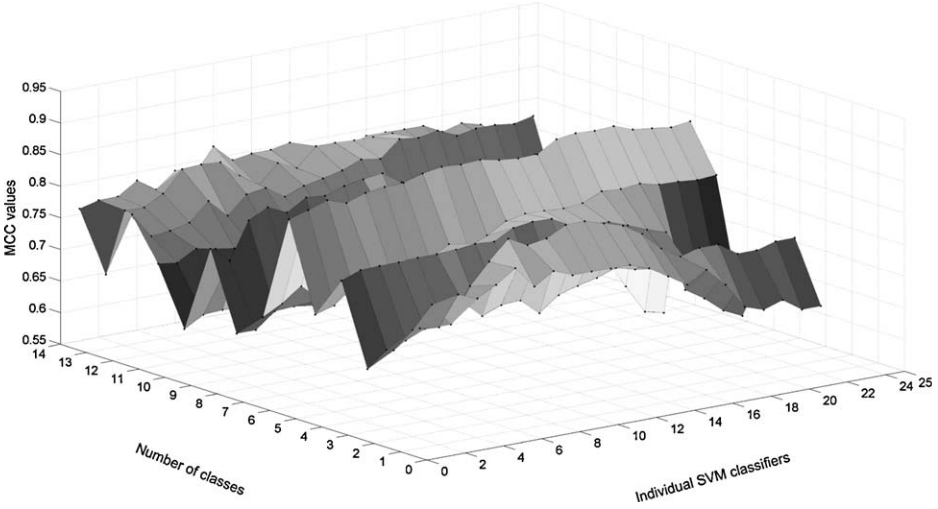

Fig. 5. *MCC* values for each of the 22 individual *SVM* classifiers using jackknife test for $CH03_{14}$ dataset.

case are found to be 0.83 and 0.82, respectively. In case of independent test using $CH00_{12}$ and $CH03_{14}$ dataset, the average *Q*-statistics of *IEs* are in the range of 0.71–0.96 and 0.93–0.98, respectively. However, *Q*-statistics of *CE*s for the same case are found to be 0.86 and 0.88, respectively. These observations show that the level of diversity in case of both *IE*s and *CE* classifiers for $CH00_{12}$ data are relatively higher compared to that of CH0314.

Table 4
Prediction performance of the proposed ensemble method for the two datasets using both independent dataset and jackknife tests.

Dataset	*IE*/*CE*	Jackknife test				Independent test		
		Correct predictions	*Acc*%	Avg. *Q*-statistics	Avg. *MCC*	Correct predictions	*Acc* %	Avg. *Q*-statistics
$CH00_{12}$ dataset	IE^{SVM}	1741	79.46	0.94	0.67	1734	69.52	0.96
	IE^{CDC}	1614	73.66	0.74	0.70	1962	78.67	0.71
	IE^{NN}	1665	75.99	0.92	0.74	2115	84.80	0.93
	IE^{PNN}	1667	76.08	0.85	0.73	2057	82.48	0.93
	Proposed *CE** using *PseAAC*	1771	80.83	0.63	0.80	2161	86.65	0.86
	CE using *PseAAC + DC*	1785	81.47	0.81	0.81	2171	87.04	0.84
$CH03_{14}$ dataset	IE^{SVM}	3072	80.86	0.93	0.45	1950	43.35	0.93
	IE^{CDC}	2694	70.91	0.92	0.63	3340	74.25	0.98
	IE^{NN}	2984	78.55	0.93	0.71	3774	83.90	0.98
	IE^{PNN}	2969	78.15	0.93	0.70	3747	83.30	0.98
	Proposed CE^{φ}	3132	82.44	0.80	0.81	3918	87.11	0.96
	Proposed *CE**	3191	83.99	0.82	0.81	3928	87.33	0.88

Note: CE^{φ} is developed by using *NN*, *PNN*, and *CDC* base classifiers in entire-pool voting scheme. However, *CE** is developed by adding *SVM* as well. However, in case of independent dataset, for best accuracy, only first three *SVMs* are included in *CE**. Parameters of SVM are optimized using grid search. Q-statistics of *IE*s are computed among the 22 individual classifiers, while Q-statistics of *CE* is computed among 66/88 individual classifiers.

Table 5
Prediction of *CE* ensemble for jackknife and independent dataset tests in terms of different evaluation measures for each of the 14 classes.

Dataset	*IE*/*CE*	Jackknife test				Independent test		
		Correct predictions	*Acc* %	Avg. *Q*-statistics	Avg. *MCC*	Correct predictions	*Acc* %	Avg. *Q*-statistics
$\mathrm{CH00}_{12}$ dataset	IE^{SVM}	1741	79.46	0.94	0.67	1734	69.52	0.96
	IE^{CDC}	1614	73.66	0.74	0.70	1962	78.67	0.71
	IE^{NN}	1665	75.99	0.92	0.74	2115	84.80	0.93
	IE^{PNN}	1667	76.08	0.85	0.73	2057	82.48	0.93
	Proposed CE^{*} using *PseAAC*	1771	80.83	0.63	0.80	2161	86.65	0.86
	CE using *PseAAC* + *DC*	1785	81.47	0.81	0.81	2171	87.04	0.84
$\mathrm{CH03}_{14}$ dataset	IE^{SVM}	3072	80.86	0.93	0.45	1950	43.35	0.93
	IE^{CDC}	2694	70.91	0.92	0.63	3340	74.25	0.98
	IE^{NN}	2984	78.55	0.93	0.71	3774	83.90	0.98
	IE^{PNN}	2969	78.15	0.93	0.70	3747	83.30	0.98
	Proposed CE^{φ}	3132	82.44	0.80	0.81	3918	87.11	0.96
	Proposed CE^{*}	3191	83.99	0.82	0.81	3928	87.33	0.88

3.2. *Performance using $\mathrm{CH00}_{12}$ dataset*

The overall accuracy of *CE-PLoc* system has also been evaluated for second $\mathrm{CH00}_{12}$ dataset. In Table 6, we summarize the results of existing approaches and compare with our proposed system. This table indicates, in jackknife test, the proposed *CE* correctly classifies 1771 protein sequences out of 2191 yielding an accuracy of 80.83%. However, in independent test, the proposed *CE* correctly classifies 2162 protein sequences out of 2494 giving an accuracy of 86.69%.

3.3. *Performance analysis using PseAAC and DC*

In the case of $\mathrm{CH03}_{12}$ dataset, the predicted values of individual classifiers are combined using both *PseAAC* and *DC* features. With additional *DC* features, overall accuracies of our method improve up to 81.47% and 87.04% for jackknife and independent data tests, respectively. The average accuracy across both types of the test mechanism is 84.255%, which is now slightly higher (83.82%) than that of Shi et al. (2007).

4. Discussion

In this work, we have observed that *CE-PLoc* provides better results using entire-pool voting as against sub-pool voting strategy. Therefore, results of proposed *CE* are reported using entire-pool voting strategy. Probably in sub-pool based voting, the selection of optimal weights is more important than the entire-pool voting.

The usefulness of individual classifiers generated through base learners is investigated. Their performance is evaluated and results are listed in Tables 2 and 3. The range of accuracy is found to be 60–80%, which is sufficient to include these individual classifiers (weak learners) in the pool for ensemble (Yeh and Mao, 2006). In Table 3, for independent test, predicted results of individual classifiers reveal the high performance of NN_{Φ} classifier (83.17%) compared to both PNN_{Φ} and SVM_{Φ} (78.10 and 81.52%, respectively). In this case, relatively better rising trend of CDC_{Φ} classifier is observed with the increase in feature dimensions. In case of independent test and for correctly predicting 4498 novel protein sequences, *SVM* classifier is unable to construct optimal decision boundary with increasing dimension. Therefore, the performance of *SVM* classifier degrades more sharply as compared to *NN* and *PNN* classifiers. This might be because *SVM* is inherently based on binary classification with increasing margin of separation. However, keeping in view both diversity based scheme and the multi-classification problem, it is found that *SVM* may not always provide accurate results (Khan et al., 2008). Under such circumstances, proximity based classifiers (*NN* and *PNN*) can better exploit feature spaces for complex multi-classification problems.

Q-values of *IE* and *CE* (Table 4) highlight the score of diversity in feature and decision space, respectively. This diversity score generated by base learners through varying feature spaces contribute in the improvement of the overall prediction performance of *CE*. The average specificity of *CE* is higher as compared to sensitivity (Table 5), i.e. the under predicted error is small as compared to over predicted. Therefore, *CE* is anticipated to predict protein sequences more precisely.

Table 4 highlights improved results of the proposed *CE-PLoc* for $\mathrm{CH03}_{14}$ dataset than that reported by Chou and Shen (2006b) and Fayyaz et al. (2007). This is because our scheme exploits diversity both in feature and decision spaces. Further, results in this table indicates better prediction performance of CE^{*} as compared to CE^{φ}. This is because by adding diverse types of base learners in the ensemble, it is expected that further improvement is possible (Yeh and Mao, 2006).

Performance of the proposed method is also investigated for $\mathrm{CH00}_{12}$ dataset. In Table 6, results are compared with the existing approaches (Chou and Cai, 2002; Gao et al., 2005; Chou, 2001; Shi et al., 2007, 2008). In these approaches, researchers have proposed different feature extraction strategies and learning algorithms using this dataset. Our approach yielded higher predictions accuracies as compared to the exiting approaches (Chou and Cai, 2002; Gao et al., 2005; Chou, 2001), except (Shi et al., 2007, 2008). Shi et al. (2008) have reported prediction accuracies of 82.15% and 85.49% using 5-fold jackknife and independent tests, respectively. In their previous work Shi et al. (2007, 2008) have shown prediction accuracies to be 80.3% and 87.0% for the same dataset. However, prediction accuracies of the proposed *CE* are 80.83% and 86.6%, for leave-one-out jackknife and independent tests, respectively. In case of our proposed *CE-PLoc*, the average accuracy across both types of the test mechanism is 83.45%, which is comparable (83.82%) to that of Shi et al. (2008). However, it is to be noted that Shi et al. (2008) have reported prediction accuracies on 5-fold jackknife test not leave-one-out jackknife. Leave-one-out jackknife test is considered to be more rigorous compared to 5-fold jackknife test. With the addition of base learners trained on *DC* features, the average accuracy of *CE-PLoc* across both types of the test mechanism is 84.255%, which is now slightly higher (83.82%) than that of Shi et al. (2008). The addition of *DC* based trained individual classifiers in the ensemble improves prediction performance of *CE* classifier but to a small extent. This is because sequence order effects are already exploited in *PseAAC* strategy. However, it is anticipated that the exploitation of other diverse types of feature extraction strategies such as transform domain (wavelet, Fourier, etc.) may enhance the overall prediction.

In brief, we can say that our proposed method is giving comparable prediction performance as reported by Shi et al. (2007, 2008) for 12 protein classes dataset. Considering the performance

Table 6
Comparative performance analysis on both the datasets with the existing approaches.

Dataset	Prediction methods		Jackknife test (leave-one-out)	Independent test	References
	Input features	Algorithm	*Acc %*	*Acc %*	
$CH00_{12}$ dataset	*PseAAC*	*CDC*	1600/2191 = 73.00	2017/2494 = 80.90	Chou (2001)
	Functional domain composition	*SVM*	1461/2191 = 66.70	2037/2494 = 81.70	Chou and Cai (2002)
	Using *PseAAC*	Aug. *CDC*	1532/2191 = 69.90	–	Gao et al. (2005)
	PseAA component	Aug. *CDC*	1590/2191 = 72.60	1865/2494 = 74.80	Xiao et al. (2006)
	Lempel-Ziv complexity	Aug. *CDC*	1612/2191 = 73.60	1990/2494 = 79.80	Xiao et al. (2005)
	PseAAC features with multi-scale energy	*SVM*	1759/2191 = 80.30	2170/2494 = 87.00	Shi et al. (2007)
	AAC distribution	*SVM*	1800/2191 = 82.15 (using 5-fold instead of leave-one-out jackknife)	2132/2494 = 85.49	Shi et al. (2008)
	PseAAC	Proposed *CE*	1771/2191 = 80.83	2162/2494 = 86.60	–
	PseAAC + *DC*	Proposed *CE*	1785/2191 = 81.47	2171/2494 = 87.04	–
$CH03_{14}$ dataset	*PseAAC*	Augmented *CDC*	2574/3799 = 67.80	3246/4498 = 72.20	Chou and Cai (2003)
	PseAAC	*CDC* ensemble	2666/3799 = 70.20	3331/4498 = 74.10	Chou and Shen (2006a,b)
	PseAAC	Weighted *CDC* ensemble	2704/3799 = 71.15	3377/4498 = 75.07	Fayyaz et al. (2007)
	PseAAC	Proposed *CE*	3191/3799 = 83.99	3930/4498 = 87.33	–

on benchmark datasets, the proposed *CE-PLoc* approach is more effective than the existing ensemble approaches as it has the ability of exploiting diversity in both feature and decision spaces.

5. Conclusion

In this work, we have proposed diversity based *CE-PLoc* system for classifying subcellular localizations of protein. Promising performance of the proposed system is observed for two benchmark datasets. This improvement is obtained by exploiting diversity in both feature and decision spaces. Using only *PseAAC* features of varying dimensions, *CE-PLoc* provides better prediction performance compared to existing approaches. Addition of base learners trained on dipeptide based features further improves the performance of *CE-PLoc*. However, the improvement is small in this case. It is also observed that by adding more learning mechanisms in the pool of ensemble enhance the prediction performance but at the cost of complexity. The proposed *CE-PLoc* approach not only has the attribute of exploiting different discrimination power of diverse feature extraction strategies, but also has the attribute of coping with the data variation by the exploitation of various learning mechanisms. The results indicate that our method could be used as a useful tool for the prediction of subcellular locations. However, user-friendly and publicly accessible web-servers provide the future direction for developing an effective, high throughput, and practically useful models, or predictors (Chou and Shen, 2009). We, therefore, intend to make efforts in our future work for providing a web-server for the method presented in this paper.

Acknowledgment

Authors are very grateful to Pakistan Institute of Engineering and Applied Sciences (PIEAS) for providing resources and environment for carrying out this research work.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compbiolchem.2011.05.003.

References

Afridi, T.H., Khan, A., Lee, Y.S., 2011. Mito-GSAAC: mitochondria prediction using genetic ensemble classifier and split amino acid composition. Amino Acids, doi:10.1007/s00726-011-0888-0.

Anand, A., Suganthan, P.N., 2009. Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates. Journal of Theoretical Biology 259, 533–540.

Bairoch, A., Apweiler, R., 1997. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. Nucleic Acids Research 25, 31–36.

Boden, M., Teasdale, R.D., 2008. Determining nucleolar association from sequence by leveraging protein-protein interactions. Journal of Computational Biology 15, 291–304.

Chang, C.C., Lin, C.J., 2008. LIBSVM: A Library for Support Vector Machines, Available from: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chou, K.C., 2000. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochemical and Biophysical Research Communications 278, 477–483.

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

Chou, K.C., 2005. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21, 10–19.

Chou, K.C., Cai, Y.D., 2002. Using functional domain composition and support vector machines for prediction of protein subcellular location. Journal of Biological Chemistry 277, 45765–45769.

Chou, K.C., Cai, Y.D., 2003. Prediction and classification of protein subcellular location – sequence-order effect and pseudo amino acid composition. Journal of Cellular Biochemistry 90, 1250–1260.

Chou, K.C., Elrod, D.W., 1999. Protein subcellular location prediction. Protein Engineering 12, 107–118.

Chou, K.C., Shen, H.B., 2006a. Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization. Biochemical and Biophysical Research Communications 347, 150–157.

Chou, K.C., Shen, H.B., 2006b. Predicting protein subcellular location by fusing multiple classifiers. Journal of Cellular Biochemistry 99, 517–527.

Chou, K.C., 2009. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Current Proteomics 6, 262–274.

Chou, K.C., Shen, H.B., 2009. Review: recent advances in developing web-servers for predicting protein attributes. Natural Science 2, 63–92.

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., Shen, H.B., 2010a. A new method for predicting the subcellular localization of eukaryotic proteins with both single and multiple sites Euk-mPLoc 2. 0. PLoS ONE 5, e9931.

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

Chou, K.C., Shen, H.B., 2010c. Cell-PLoc 2.0: an improved package of web-servers for predicting subcellular localization of proteins in various organisms. Natural Science 2, 1090–1103.

Chou, K.C., 2011. Some remarks on protein attribute prediction and pseudo amino acid composition. Journal of Theoretical Biology 273, 236–247 (50th Anniversary Year Review).

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

Chou, K.C., Zhang, C.T., 1995. Prediction of protein structural classes. Critical Reviews in Biochemistry and Molecular Biology 30, 275–349.

Daun, S., Rubin, J., Vodovotz, Y., Roy, A., Parker, R., Clermont, G., 2008. An ensemble of models of the acute inflammatory response to bacterial lipopolysaccharide in rats: results from parameter space reduction. Journal of Theoretical Biology 253, 843–853.

Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F., 2003. Prediction of protein function using protein–protein interaction data. Journal of Computational Biology 10, 947–960.

Denoeux, T., 1995. A *k*-nearest neighbor classification rule based on Dempster–Shafer theory. IEEE Transactions on Systems Man and Cybernetics 25, 804–813.

Du, P., Cao, S., Li, Y., 2009. SubChlo: predicting protein sub-chloroplast locations with pseudo-amino acid composition and the evidence-theoretic K-nearest neighbor (ET-KNN) algorithm. Journal of Theoretical Biology 261, 330–335.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. John Wiley & Sons, Inc., New York.

Fayyaz, M., Mujahid, A., Khan, A., Bangash, A., 2007. Prediction of protein sub-cellular localization through weighted combination of classifiers. In: 2007 International Conference on Electrical Engineering (ICEE, 2007) , April 11–12, 2007, Lahore, Pakistan.

Gao, Y., Shao, S., Xiao, X., Ding, Y., Huang, Y., Huang, Z., Chou, K.C., 2005. Using pseudo amino acid composition to predict protein subcellular location: approached with Lyapunov index, Bessel function, and Chebyshev filter. Amino Acids 28, 373–376.

Garg, A., Bhasin, M., Raghava, G.P.S., 2005. Support vector machine-based method for subcellular localization of human proteins using amino acid compositions. their order, and similarity search. Journal of Biological Chemistry 280, 14427–14432.

Guo, X., Gao, X., 2008. A novel hierarchical ensemble classifier for protein fold recognition. Protein Engineering Design and Selection 21, 659–664.

Hu, L., Huang, T., Shi, X., Lu, W.C., Cai, Y.D., 2011. Predicting functions of proteins in mouse based on weighted protein–protein interaction network and protein hybrid properties. PLoS ONE 6, e14556.

He, Z., Zhang, J., Shi, X.H., Hu, L.L., Kong, X., 2010. Predicting drug–target interaction networks based on functional groups and biological features. PLoS ONE 5, e9603.

Hayat, M., Khan, A., 2011. Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition. Journal of Theoretical Biology 271, 10–17.

Jaimovich, A., Elidan, G., Margalit, H., Friedman, N., 2006. Towards an integrated protein–protein interaction network: a relational markov network approach. Journal of Computational Biology 13, 145–164.

Jia, P., Qian, Z., Zeng, Z., Cai, Y., Li, Y., 2007. Prediction of subcellular protein localization based on functional domain composition. Biochemical and Biophysical Research Communications 357, 366–370.

Kandaswamy, K.K., Chou, K.C., Martinetz, T., Moller, S., Suganthan, P.N., 2011. AFP-Pred: a random forest approach for predicting antifreeze proteins from sequence-derived properties. Journal of Theoretical Biology 270, 56–62.

Kavousi, K., Moshiri, B., Sadeghi, M., Araabi, B.N., Moosavi-Movahedic, A.A., 2010. A protein folds classifier formed by fusing different modes of pseudo amino acid composition via PSSM. Journal of Computational Biology and Chemistry 35, 1–9.

Khan, A., Fayyaz, M., Choi, T.S., 2008. Proximity based GPCRs prediction in transform domain. Biochemical and Biophysical Research Communications 371, 411–415.

Khan, A., Majid, A., Mirza, A.M., 2005. Combination and optimization of classifiers in gender classification using genetic programming. International Journal of Knowledge-Based Intelligent Engineering Systems 9, 1–11.

Kuncheva, L.I., Whitaker, C.J., 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51, 181–207.

Li, S., Liu, B., Zeng, R., Cai, Y., Li, Y., 2006. Predicting O-glycosylation sites in mammalian proteins by using SVMs. Journal of Computational Biology and Chemistry 30, 203–208.

Li, F.M., Li, Q.Z., 2008. Predicting protein subcellular location using Chou's pseudo amino acid composition and improved hybrid approach. Protein & Peptide Letters 15, 612–616.

Lin, H., 2008. The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition. Journal of Theoretical Biology 252, 350–356.

Mohabatkar, H., 2010. Prediction of cyclin proteins using Chou's pseudo amino acid composition. Protein & Peptide Letters 17, 1207–1214.

Majid, A., Khan, A., Mirza, A.M., 2006. Combination of support vector machines using genetic programming I. International Journal of Hybrid Intelligent Systems 3, 109–125.

Matthews, B.W., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta 405, 442–451.

Mei, S., Fei, W., Zhou, S., 2011. Gene ontology based transfer learning for protein subcellular localization. BMC Bioinformatics 12, 44.

Naveed, M., Khan, A., 2011. GPCR-MPredictor: multi-level prediction of G protein-coupled receptors using genetic ensemble. Amino Acids.

Pan, Y.X., Zhang, Z.Z., Guo, Z.M., Feng, G.Y., Huang, Z.D., He, L., 2003. Application of pseudo amino acid composition for predicting protein subcellular location: stochastic signal processing approach. Journal of Protein Chemistry 22, 395–402.

Qiu, J.D., Huang, J.H., Shi, S.P., Liang, R.P., 2010. Using the concept of Chou's pseudo amino acid composition to predict enzyme family classes: an approach with support vector machine based on discrete wavelet transform. Protein & Peptide Letters 17, 715–722.

Rehman, Z.U., Khan, A., 2011. GPCR prediction using pseudo amino acid composition and multi-scale energy representation of different physiochemical properties. Analytical Biochemistry 412, 173–182.

Sahu, S.S., Panda, G., 2010. A novel feature representation method based on Chou's pseudo amino acid composition for protein structural class prediction. Journal of Computational Biology and Chemistry 34, 320–327.

Schölkopf, B., Smola, A.J., Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319.

Shen, Y.Q., Burger, G., 2010. TEST Loc: protein subcellular localization prediction from EST data. BMC Bioinformatics 11, 563.

Shen, H.B., Chou, K.C., 2010. Gneg-mPLoc: a top-down strategy to enhance the quality of predicting sub-cellular localization of Gram-negative bacterial proteins. Journal of Theoretical Biology 264, 326–333.

Shen, H.B., Yang, J., Chou, K.C., 2007. Euk-PLoc: an ensemble classifier for large-scale eukaryotic protein subcellular location prediction. Amino Acids 33, 57–67.

Shen, Y., Burger, G., 2007. 'Unite and conquer': enhanced prediction of protein subcellular localization by integrating multiple specialized tools. BMC Bioinformatics 8, 420–430.

Shi, J.Y., Zhang, S.W., Pan, Q., Cheng, Y.M., Xie, J., 2007. Prediction of protein subcellular localization by support vector machines using multi-scale energy and pseudo amino acid composition. Amino Acids 33, 69–74.

Shi, J.Y., Zhang, S.W., Pan, Q., Zhou, G.P., 2008. Using pseudo amino acid composition to predict protein subcellular location: approached with amino acid composition distribution. Amino Acids 35, 321–327.

Suykens, J.A.K., Van Gestel, T., Vandewalle, J., De Moor, B., 2003. A support vector machine formulation to PCA analysis and its Kernel version. IEEE Transactions on Neural Network 14, 447–450.

Tan, F., Feng, X., Fang, Z., Li, M., Guo, Y., Jiang, L., 2007. Prediction of mitochondrial proteins based on genetic algorithm – partial least squares and support vector machine. Amino Acids 33, 669–675.

Xiao, X., Shao, S., Ding, Y., Huang, Z., Huang, Y., Chou, K.C., 2005. Using complexity measure factor to predict protein subcellular location. Amino Acids 28, 57–61.

Xiao, X., Shao, S., Ding, Y., Huang, Z., Chou, K.C., 2006. Using cellular automata images and pseudo amino acid composition to predict protein subcellular location. Amino Acids 30, 49–54.

Xiao, X., Wang, P., Chou, K.C., 2011. GPCR-2L: predicting G protein-coupled receptors and their types by hybridizing two different modes of pseudo amino acid compositions. Molecular Biosystems 7, 911–919.

Xu, Y., Wang, X.B., Ding, J., Wu, L.Y., Deng, N.Y., 2010. Lysine acetylation sites prediction using an ensemble of support vector machine classifiers. Journal of Theoretical Biology 264 (1), 130–135.

Yeh, J.I., Mao, L., 2006. Prediction of membrane proteins in mycobacterium tuberculosis using a support vector machine algorithm. Journal of Computational Biology 13, 126–129.

Zhang, L., Liao, B., Li, D., Zhu, W., 2009. A novel representation for apoptosis protein subcellular localization prediction using support vector machine. Journal of Theoretical Biology 259, 361–365.

Zhang, S.W., Pan, Q., Zhang, H.C., Shao, Z.C., Shi, J.Y., 2006a. Prediction of protein homo-oligomer types by pseudo amino acid composition: approached with an improved feature extraction and Naive Bayes Feature Fusion. Amino Acids 30, 461–468.

Zhang, T.L., Ding, Y.S., Chou, K.C., 2008. Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern. Journal of Theoretical Biology 250, 186–193.

Zhang, T., Ding, Y., Chou, K.C., 2006b. Prediction of protein subcellular location using hydrophobic patterns of amino acid sequence. Journal of Computational Biology and Chemistry 30, 367–371.

Zhang, T., Ding, Y., Chou, K.C., 2006c. Prediction of protein subcellular location using hydrophobic patterns of amino acid sequence. Computational Biology and Chemistry 30, 367–371.

Zouhal, L.M., Denoeux, T., 1998. An evidence-theoretic K-NN rule with parameter optimization. IEEE Transactions on Systems Man and Cybernetics 28, 263–271.