Amino Acids (2013) 44:1365–1379
DOI 10.1007/s00726-013-1472-6

ORIGINAL ARTICLE

Learning protein multi-view features in complex space

Dong-Jun Yu · Jun Hu · Xiao-Wei Wu · Hong-Bin Shen · Jun Chen · Zhen-Min Tang · Jian Yang · Jing-Yu Yang

Received: 13 June 2012 / Accepted: 13 February 2013 / Published online: 28 February 2013
© Springer-Verlag Wien 2013

Abstract Protein attribute prediction from primary sequences is an important task and how to extract discriminative features is one of the most crucial aspects. Because single-view feature cannot reflect all the information of a protein, fusing multi-view features is considered as a promising route to improve prediction accuracy. In this paper, we propose a novel framework for protein multi-view feature fusion: first, features from different views are parallely combined to form complex feature vectors; Then, we extend the classic principal component analysis to the generalized principle component analysis for further feature extraction from the parallely combined complex features, which lie in a complex space. Finally, the extracted features are used for prediction. Experimental results on different benchmark datasets and machine learning algorithms demonstrate that parallel strategy outperforms the traditional serial approach and is particularly helpful for extracting the core information buried among multi-view feature sets. A web server for protein structural class prediction based on the proposed method (COMSPA) is freely available for academic use at: http://www.csbio.sjtu.edu.cn/bioinf/COMSPA/.

Keywords Protein attribute prediction · Feature extraction · Serial feature fusion · Parallel feature fusion · Complex space

D.-J. Yu · J. Hu · X.-W. Wu · Z.-M. Tang · J. Yang · J.-Y. Yang
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

D.-J. Yu
Changshu Institute, Nanjing University of Science and Technology, Changshu 215500, China

H.-B. Shen (✉) · J. Chen
Department of Automation, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: hbshen@sjtu.edu.cn

Introduction

Much work has been conducted to improve the accuracy of the protein attribute prediction from the primary sequence (Gao et al. 2010; Mizianty and Kurgan 2011; Kurgan and Disfani 2011; Smialowski et al. 2007a, b). From the perspective of pattern recognition, the corresponding task is a typical classification/prediction problem. Achieving satisfactory prediction accuracy when using multiple sets of information could be a challenging task. In general, there exists three popular schemes: (1) information level fusion (Dasigi et al. 2001): the information from individual information sets is fused together into a final decision to the problem at hand; (2) feature level fusion (Ulug and McCullough 1999): multiple feature sets are extracted from the multiple information sets; then, the obtained multiple feature sets are fused to perform a decision; (3) decision level fusion (Kuncheva 2004): the individual decisions are first made based on the different feature sets, and then they are recombined to obtain the final decision. These different fusion schemes have been exploited in many bioinformatics researches. In this study, we mainly focus on the feature level fusion. Our task is to investigate an efficient feature fusion approach to obtain better prediction accuracies. We do this by merging multiple views of protein sequence including amino acid composition and position-specific score matrix obtained by multiple sequence alignment.

As far as the protein attribute prediction is concerned, in the early stage, features extracted from the original amino acid sequence itself, i.e., amino acid composition (AAC), are widely used for inputting into a statistical learning machine for predicting the protein attributes. Subsequently, many variants have been presented to improve the quality of AAC feature. One of the representative ones is called pseudo amino acid composition (PseAAC) (Chou 2001) that can reflect the sequence-order information besides the amino acid composition, and has been broadly used. Other frequently used sequential features are di- and tri-peptide compositions (Smialowski et al. 2006, 2007b; Nanni and Lumini 2008; Shen and Chou 2008).

Recently, features derived from position-specific scoring matrix (PSSM), which contains evolutionary information obtained from multiple sequence alignment, are popular and they were shown to yield better prediction results when compared with AAC and their variants (Jeong et al. 2011; Pierleoni et al. 2011; Chen and Kurgan 2007; Yu et al. 2011). PsePSSM (Chou and Shen 2007), which encodes both the evolutionary information and sequence-order information of a protein, is a further variant and has been demonstrated effective.

Multiple previous studies have shown that different protein features have their own merits and shortcomings, where single-view feature can not reflect all the properties of a protein. For example, the AAC and PseAAC features reflect the physical composition characteristics of a protein, while PSSM and PsePSSM features describe the evolutionary characteristics of a protein. As the two types of features describe different aspects of a protein, an intuition is that the two types of features may be complementary and fusion of them may potentially improve the prediction accuracy.

As to feature level fusion, the existing techniques can be divided into three categories, i.e., feature-combination-based, feature-selection-based, and feature-extraction-based methods (Yang et al. 2003).

Feature-combination-based method is the most straightforward approach, by which different feature vectors are combined to form a super vector, and then the combined feature vector is used to perform classification/prediction. For example, a 100-dimensional quasi-sequence-order feature and a 21-dimensional physicochemical composition feature were combined to form a 121-dimensional feature for protein structural class prediction (Chen et al. 2008a); Jeong et al. (2011) combined two 400-dimensional and one 180-dimensional PSSM-based feature vectors and the obtained 980-dimensional feature vector was then used for protein function prediction; It is anticipated that the combination of features should improve the prediction accuracy if the features represent different discriminative information. However, the combination of features will simultaneously increase the information redundancy that could, in turn, deteriorate the prediction accuracy (Kohavi and John 1997). Previous studies have shown that directly combining different features will sometimes, but not definitely lead to the improvement of prediction accuracy when compared with a single view feature. For example, Chen et al. (2008a) investigated 10 different features and found that directly combining different features will, in most cases, lead to an "intermediate" prediction accuracy, i.e., the prediction accuracy of the combined feature lies between the worst and best prediction accuracies of the individual features. Jeong et al. (2011) also found that directly integrating different features does not always improve the results. Similar phenomenon was also observed in our experimental results of this study (see experimental results for details).

Importantly, from a conceptual point of view, this kind of feature-combination-based method is not a real feature fusion process. Feature fusion includes feature combination but more than it (Yang et al. 2003). In fact, feature fusion should be a process of reprocessing the combined features: retaining the favorable discriminatory information and reducing/eliminating unfavorable redundant or conflicting information. We believe it is preferable to perform the classification/prediction after the process of feature fusion than after the process of simple feature combination.

In feature-selection-based method, multiple feature sets are grouped together and then a suitable method facilitate selection of subset of relevant attributes. The hypothesis for the feature-selection-based method is that not all the feature components that can be calculated are informative. Many feature-selection-based fusion methods have been successfully applied in protein attribute prediction such as disulfide connectivity prediction (Zhu et al. 2010), protein–protein interaction prediction (Liu et al. 2009). Saeys et al. (2007) have presented a good review of feature selection techniques in bioinformatics.

Different from the feature selection methods that do not change the features themselves, feature-extraction-based approaches apply some mathematic transformations (e.g., principal component analysis (PCA), linear discriminate analysis (LDA), etc.) on the super-vector combined from the multiple feature sets. Some reports have shown that these feature-extraction-based algorithms can be successfully used to predict protein attributes (Dima and Thirumalai 2004).

It is easy to find that in feature level fusion, a common step is to combine multi-view feature vectors into one feature vector. The traditional method for combining multiple feature vectors is to group different sets of feature vectors into one super vector (serial combination). Recently, we proposed parallel combination strategy (Yang et al. 2003), which parallely combines two sets of feature

vectors by a complex vector for further feature extraction. For the convenience of subsequent description, here we define *serial feature fusion* and *parallel feature fusion* as follows:

Serial feature fusion

Features from protein different views are serially combined (refer to Eq. (15)) and then processed by appropriate feature extraction method.

Parallel feature fusion

Features from protein different views are parallely combined (refer to Eq. (16)) and then processed by appropriate feature extraction method.

Another important reason why we explore parallel feature fusion in bioinformatics problems is that parallel feature fusion is much more natural and convenient under some specific computational biology scenario. Taking disulfide bond prediction (Zhu et al. 2010) as an example, the task is to determine whether a given cysteine–cysteine pair is a disulfide bond. Obviously, we cannot make a decision based on the single cysteine's feature and have to simultaneously utilize both features of the two cysteines. As the features of the two cysteines are of the same size and well matched, it is thus natural to combine them into a complex vector, among which the real part and the imaginary part contain the corresponding feature components of the two cysteines, respectively.

Our experimental results from different classification algorithms on different benchmark datasets illustrate that the proposed parallel feature fusion method, which represents different sources of protein features separately with real and imaginary parts in the complex space, is helpful for extracting more discriminative classification features, and thus achieves better prediction success rates.

Materials

Benchmark datasets

Protein structural class datasets

Structural class knowledge of a specific protein provides useful information for the protein function understanding and plays important roles in the prediction of secondary structure and tertiary structure. Levitt and Chothia (1976) introduced the concept of protein structural class and classified proteins into the four structural classes: (1) all-α, (2) all-β, (3) α/β, and (4) α + β. The all-α and all-β proteins are essentially formed by α-helices and β-strands, respectively. The α/β class represents those proteins with both α-helices and β-strands that are largely interspersed in forming mainly parallel β-sheets, while the α + β class represents those also with both α-helices and β-strands, but they are largely segregated in forming mainly anti-parallel β-sheets. In this study, four widely used benchmark protein structural class datasets; namely, Z277 (Zhou 1998), Z498 (Zhou 1998), C204 (Chou 1999), and W1189 (Wang and Yuan 2000), are taken to investigate the effectiveness of the proposed method. Note that there are 277, 498, 204, and 1,189 protein sequences in benchmark datasets Z277, Z498, C204, and W1189, respectively, and each of the four benchmark datasets contains all the four types of structural classes.

Disulfide connectivity dataset

Disulfide bond is formed by the oxidation of the thiol (–SH) groups between two cysteine residues in the same or different protein polypeptide chains, which plays essential role in folding, stability and maturation of many proteins (Tsai et al. 2007). Owing to its important roles played on the structure and function of proteins, many methods are developed on the disulfide connectivity prediction, whose tasks are to identify the correct pairing of bound cysteine residues. In this study, the widely used dataset constructed in (Fariselli and Casadio 2001) was adopted as the benchmark. This dataset was prepared according to the following: (1) the proteins were from the release of Swiss-Prot database version 39 at www.ebi.ac.uk/swissprot/; (2) only protein sequences containing intra-chain disulfide bonds that were experimentally verified were included, whereas the interchain disulfide bonds were not considered and discarded; (3) protein sequences containing at least 2 and at most 5 disulfide bonds were selected; (4) to avoid the bias, a redundancy cutoff was operated to exclude the sequences which have 30 % pairwise sequence identity to any other in the dataset. Finally, there are 446 proteins and 1,371 disulfide bonds in the current dataset.

Extracting features from different views

PseAAC

Pseudo amino acid composition (PseAAC) proposed by Chou (2001) encodes both the amino acid composition information and the sequence-order information of amino acids in a protein. In this paper, we encode each protein into a $(20 + \xi \cdot \lambda)$-dimensional feature vector, where the first 20 components are the classical amino acid composition, while the remaining $\xi \cdot \lambda$ components are scalar quantities reflecting the sequence-order information of the protein. ξ is the number of the amino acid physiochemical

characteristics used, λ is the rank of correlation along the protein sequence. Six amino acid physiochemical characteristics ($\xi = 6$) are used in this study: (1) hydrophobicity (H^1), (2) hydrophilicity (H^2), (3) side chain mass (S), (4) pK of the α–COOH group (P^1), (5) pK of the α–NH_2 group (P^2), and (6) pI at 25 °C (I). Then, for a protein sequence with L amino acid residues, its sequence-order information can be reflected by a set of sequence order-correlated factors defined as:

$$\left\{\begin{array}{l} 1-\text{tier}\left|\begin{array}{l}\tau_1=\frac{1}{L-1}\sum_{i=1}^{L-1}H^1_{i,i+1},\tau_2=\frac{1}{L-1}\sum_{i=1}^{L-1}H^2_{i,i+1},\tau_3=\frac{1}{L-1}\sum_{i=1}^{L-1}S_{i,i+1},\\ \tau_4=\frac{1}{L-1}\sum_{i=1}^{L-1}P^1_{i,i+1},\tau_5=\frac{1}{L-1}\sum_{i=1}^{L-1}P^2_{i,i+1},\tau_6=\frac{1}{L-1}\sum_{i=1}^{L-1}I_{i,i+1},\end{array}\right.\\ 2-\text{tier}\left|\begin{array}{l}\tau_7=\frac{1}{L-2}\sum_{i=1}^{L-2}H^1_{i,i+2},\tau_8=\frac{1}{L-2}\sum_{i=1}^{L-2}H^2_{i,i+2},\tau_9=\frac{1}{L-2}\sum_{i=1}^{L-2}S_{i,i+2},\\ \tau_{10}=\frac{1}{L-2}\sum_{i=1}^{L-2}P^1_{i,i+2},\tau_{11}=\frac{1}{L-2}\sum_{i=1}^{L-2}P^2_{i,i+2},\tau_{12}=\frac{1}{L-2}\sum_{i=1}^{L-2}I_{i,i+2},\end{array}\right.\\ \cdots\\ \lambda-\text{tier}\left|\begin{array}{l}\tau_{6\lambda-5}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}H^1_{i,i+\lambda},\tau_{6\lambda-4}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}H^2_{i,i+\lambda},\tau_{6\lambda-3}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}S_{i,i+\lambda},\\ \tau_{6\lambda-2}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}P^1_{i,i+v},\tau_{6\lambda-1}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}P^2_{i,i+\lambda},\tau_{6\lambda}=\frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}I_{i,i+\lambda}\end{array}\right.\end{array}\right. \quad (1)$$

where λ is the rank of correlation and $\lambda < L$. $H^1_{i,j}$ is the hydrophobicity correlation function given by

$$H^1_{i,j} = H^1(R_i)\cdot H^1(R_j) \quad (2)$$

where $H^1(R_i)$ is the hydrophobicity value of the ith residue in the protein. Other five correlation functions, such as $H^2_{i,j}$,$S_{i,j}$,$P^1_{i,j}$,$P^2_{i,j}$, and $I_{i,j}$ are similarly defined as $H^1_{i,j}$ by replacing $H^1(\cdot)$ with corresponding amino acid physiochemical characteristic values. $\tau_1 \sim \tau_6$ are called the 1-tier correlation factors that reflect the sequence-order correlations between all the most contiguous residues along a protein chain through hydrophobicity, hydrophilicity, side chain mass, pK of the α–COOH group, pK of the α–NH3 + group, and pI at 25 °C, respectively; $\tau_7 \sim \tau_{12}$ are the corresponding 2-tier correlation factors that reflect the sequence-order correlation between all the second-most contiguous residues; and so forth.

Let $\mathbf{F}_{AAC} = (f_1, f_2, \cdots, f_{20})^T$ be the classical 20-D amino acid composition, in which $f_i (i = 1, 2, \cdots, 20)$ are the normalized occurrence frequencies of the 20 native amino acids in the protein. Then, the PseAAC vector is the weighted combination of $\mathbf{F}_{AAC}$ and $\{\tau_j\}_{j=1}^{6\lambda}$ of Eq. (1) as follows:

$$\mathbf{x}^{\lambda}_{PseAA} = (x_1, \cdots, x_{20}, x_{20+1}, \cdots, x_u, \cdots, x_{20+6\lambda})^T \quad (3)$$

where

$$x_u = \left\{\begin{array}{l}\frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{6\lambda}\tau_j}, \ (1 \le u \le 20)\\ \frac{w\tau_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{6\lambda}\tau_j}, \ (20+1 \le u \le 20+6\lambda)\end{array}\right. \quad (4)$$

where w is the weight factor. In all the experiments on the four protein structural class benchmark datasets, the weight factor w is set to be 0.1. We also empirically tested and found that when $\lambda = 15$, better results were obtained on Z277 (Zhou 1998), Z498 (Zhou 1998), C204 (Chou 1999). However, because the λ is the rank of correlation along a protein sequence, which should be smaller than the length of input protein sequence. When considering that the minimal protein length in W1189 dataset is 10, thus the maximal λ is set to be 9 in this dataset. PseAAC features were generated via the online calculator at http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/.

PsePSSM

Position-specific scoring matrix (PSSM) can partially provide the evolutionary information of a protein sequence, which is obtained from multiple sequence alignment. For a protein sequence $\mathbf{P}$ with L amino acid residues, we obtain its PSSM (L rows and 20 columns) by using the PSI-BLAST (Schaffer 2001) to search the Swiss-Prot database through three iterations with 0.001 as the E value cutoff for multiple sequence alignment against the sequence of the protein. The original PSSM matrix of a protein with

L amino acid residues generated by PSI-BLAST is in the form of:

$$\mathbf{P}_{pssm}^{original} = \begin{pmatrix} o_{1,1} & o_{1,2} & \cdots & o_{1,20} \\ o_{2,1} & o_{2,2} & \cdots & o_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ o_{k,1} & o_{k,2} & \cdots & o_{k,20} \\ \vdots & \vdots & \vdots & \vdots \\ o_{L,1} & o_{L,2} & \cdots & o_{L,20} \end{pmatrix}_{L\times 20} \tag{5}$$

where $o_{k,j}$ represents the score of the amino acid residue k in the protein sequence being mutated to amino acid type j during the evolution process. The positive score means that the corresponding mutation occurs more frequently in the alignment than expected by chance, while the negative score one means just the opposite. Note that here we use the numerical code 1, 2, $\cdots$, 20 to represent the 20 native amino acid types according to the alphabetical order of their single character codes.

To facilitate the latter computation, the original PSSM of a protein is further normalized row by row (Shen et al. 2007). Let u_k and σ_k be the mean and standard deviation of the 20 scores in row k of $\mathbf{P}_{pssm}^{original}$, respectively, i.e.

$$u_k = \frac{1}{20}\sum_{t=1}^{20} o_{k,t} \tag{6}$$

$$\sigma_k = \sqrt{\frac{1}{20}\sum_{t=1}^{20}\left(o_{k,t} - u_k\right)^2} \tag{7}$$

Then, the normalized PSSM is $\mathbf{P}_{pssm} = \left(p_{k,j}\right)_{L\times 20}$, where the element $p_{k,j}$ in $\mathbf{P}_{pssm}$ is obtained as follows:

$$p_{k,j} = \frac{o_{k,j} - u_k}{\sigma_k} \tag{8}$$

Let $\mathbf{P}_{pssm}$ be the normalized PSSM of a protein with L amino acid residues.

$$\mathbf{P}_{pssm} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,20} \\ \vdots & \vdots & \vdots & \vdots \\ p_{L,1} & p_{L,2} & \cdots & p_{L,20} \end{pmatrix}_{L\times 20} \tag{9}$$

Then, the PSSM composition is a 20-dimensional feature vector as defined

$$\mathbf{F}_{PSSM} = (p_1, p_2, \cdots, p_{20})^T \tag{10}$$

where

$$p_j = \frac{1}{L}\sum_{t=1}^{L} p_{t,j} \tag{11}$$

In $\mathbf{F}_{PSSM}$, p_j represents the average score of the amino acid residues in a protein being mutated to amino acid type j during the evolution process. Although the evolutionary information of a protein can be partially reflected by $\mathbf{F}_{PSSM}$, all the sequence-order information during the evolution process of the protein would be lost. To remedy the defection of losing sequence-order information, sequence-order information contained in PSSM is then extracted by calculating the correlation factor of each column of a PSSM as follows:

$$\boldsymbol{\theta}^g = \left(\theta_1^g, \theta_2^g, \cdots, \theta_{20}^g\right)^T \tag{12}$$

where $\theta_j^g = \frac{1}{L-g}\sum_{t=1}^{L-g}\left(p_{t,j} - p_{t+g,j}\right)^2$, $1 \le j \le 20$, $0 \le g \le G$, $G < L$. g is the rank of correlation along the protein sequence. The scalar quantity θ_j^g is the correlation factor by coupling the g-most contiguous PSSM scores along the protein sequence for the amino acid type j.

Then, the pseudo position-specific scoring matrix (PsePSSM) feature vector is the combination of $\mathbf{F}_{PSSM}$ (refer to Eq. (10)) and $\boldsymbol{\theta}^g$ (refer to Eq. (12)) as defined

$$\mathbf{x}_{PsePSSM}^g = \begin{pmatrix} \mathbf{F}_{PSSM} \\ \boldsymbol{\theta}^g \end{pmatrix} = \left(p_1, p_2, \cdots, p_{20}, \theta_1^g, \theta_2^g, \cdots, \theta_{20}^g\right)^T \tag{13}$$

According to Chou and Shen (2007), a protein can be represented by a set of PsePSSM feature vectors, denoted as $\{\mathbf{x}_{PsePSSM}^g\}_{g=0}^G$. Note that $\mathbf{x}_{PsePSSM}^g$ degenerates to $\mathbf{F}_{PSSM}$ when g (the rank of correlation along the protein sequence) $= 0$. Of course, one can concatenate the $G+1$ PsePSSM feature vectors into one $(20 + G \cdot 40)$-dimensional feature vector. However, there exist two disadvantages by doing so. First, the combined feature vector will contain redundant information as each PsePSSM encodes a $\mathbf{F}_{PSSM}$; second, the dimensionality of the combined feature vector is very high. To avoid these two disadvantages, in this study, a more compact PsePSSM feature vector is defined as follows:

$$\mathbf{x}_{PsePSSM}^G = \begin{pmatrix} \mathbf{F}_{PSSM} \\ \boldsymbol{\theta}^1 \\ \vdots \\ \boldsymbol{\theta}^G \end{pmatrix} \tag{14}$$

Then, the dimensionality of the redefined PsePSSM feature vector is $20 + G \cdot 20$. The next problem is how to choose an appropriate value of G. Because there are no theoretical justifications on determining the optimal value of G, we thus calculated overall accuracy for 10 values of G (1–10) using the benchmark datasets Z277 and W1189. It was found that the optimal value of G varies on different datasets. For example, the optimal value of G is 6 on dataset Z277, while the optimal value of G is 8 on dataset W1189. In fact, when $3 < G < 9$, the prediction accuracy slightly fluctuates and when $G > 9$, the overall accuracy tends to drop down on both datasets. To uniformly evaluate

the performance of the proposed method on different datasets G is set to be 6 in following experiments.

Methods

Parallel fusion of features from different views

Parallel combination

Let A and B be two feature spaces, e.g., PseAAC and PsePSSM feature spaces in this study, defined on training protein sample space Ω, the dimensionalities of A and B are n and m, respectively. For a given protein sample $\gamma \in \Omega$, its corresponding feature vectors are $\mathbf{x} \in A$ and $\mathbf{y} \in B$, respectively. In traditional serial combination, we will obtain a super vector of $\mathbf{z}$ by combining $\mathbf{x}$ and $\mathbf{y}$ as:

$$\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \tag{15}$$

Obviously, the dimensionality of the serially combined feature vector is $n + m$. As many previous studies have revealed that directly using the serial combination feature for classification/prediction is not satisfactory, whose performance is even worse than that of the single $\mathbf{x}$ or $\mathbf{y}$ in many cases (Chen et al. 2008a).

Rather than the serial combination method which combines two feature vectors into a super vector, parallel combination method combines two feature vectors to a complex vector (Yang et al. 2003) as follows:

$$\mathbf{z} = \mathbf{x} + i \cdot \mathbf{y} \tag{16}$$

where i is the imaginary unit. Note that when the dimensionalities of $\mathbf{x}$ and $\mathbf{y}$ are not equal, we fill the lower dimensional vector with zeros until dimensionality of both is equal. As an example, suppose that $\mathbf{x} = (x_1, x_2)^T$, and $\mathbf{y} = (y_1, y_2, y_3)^T$. Then, $\mathbf{x}$ is first turned into $(x_1, x_2, 0)^T$ and then combined with $\mathbf{y}$. The resulting complex vector is $\mathbf{z} = (x_1 + i \cdot y_1, x_2 + i \cdot y_2, 0 + i \cdot y_3)^T$.

Let us define the parallely combined feature space on Ω as $C = \{\mathbf{x} + i \cdot \mathbf{y} | \mathbf{x} \in A, \mathbf{y} \in B\}$. Thus, C is an n-dimensional complex vector space, where $n = \max(\dim A, \dim B)$. The inner product in the complex space is defined by

$$(\mathbf{a}, \mathbf{b}) = \mathbf{a}^H \mathbf{b} \tag{17}$$

where $\mathbf{a}, \mathbf{b} \in C$, and H is the denotation of conjugate transpose.

The complex vector space defined by the above inner product is usually called unitary space. In unitary space, the norm is defined as

$$\|\mathbf{z}\| = \sqrt{\mathbf{z}^H \mathbf{z}} = \sqrt{\sum_{j=1}^{n} \left(a_j^2 + b_j^2\right)} \tag{18}$$

where $\mathbf{z} = (a_1 + i \cdot b_1, \cdots, a_n + i \cdot b_n)^T$.

The unitary distance between two complex vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ is defined by

$$\|\mathbf{z}_1 - \mathbf{z}_2\| = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^H (\mathbf{z}_1 - \mathbf{z}_2)} \tag{19}$$

Parallel fusion by generalized principle component analysis (GPCA)

If samples are directly classified based on the serially combined feature $\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ or the parallely combined feature $\mathbf{z} = \mathbf{x} + i \cdot \mathbf{y}$, and the Euclidean distance is adopted in the serially combined feature space while the unitary distance is used in parallely combined feature space, they will result in the same classification accuracy. However, the combined feature vectors are always high dimensional and contains much redundant information and some conflicting information which are unfavorable for classification. Consequently, in general, we would rather perform the classification/prediction after further feature extraction process.

Principle component analysis (PCA) is a classical approach for achieving this goal (Pearson 1901). However, the classic PCA can only be performed in a real space and is not suitable for a complex space. To circumvent this problem, we have proposed the generalized principle component analysis (GPCA) technique, which can perform principal component analysis in a complex space (Yang et al. 2003). Here, we briefly summarize GPCA as follows:

Suppose that the feature vector $\mathbf{z}$ lies in an unitary space, L be the number of pattern classes, $P(\omega_i)$be the prior probability of pattern class i, $\bar{\mathbf{z}}_i = E\{\mathbf{z}|\omega_i\}$ be the mean feature vector of pattern class i, $\bar{\mathbf{z}} = E\{\mathbf{z}\} = \sum_{i=1}^{L} P(\omega_i) \cdot \bar{\mathbf{z}}_i$ be the mean vector of all the feature vectors, the between-class scatter matrix, within-class scatter matrix, and total-scatter matrix are, respectively, defined as follows:

$$S_b = \sum_{i=1}^{L} P(\omega_i)(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})^H \tag{20}$$

$$S_w = \sum_{i=1}^{L} P(\omega_i) E\left\{(\mathbf{z} - \bar{\mathbf{z}}_i)(\mathbf{z} - \bar{\mathbf{z}}_i)^H | \omega_i\right\} \tag{21}$$

$$S_t = S_b + S_w = E\left\{(\mathbf{z} - \bar{\mathbf{z}})(\mathbf{z} - \bar{\mathbf{z}})^H\right\} \tag{22}$$

where H is the denotation of conjugate transpose.

From Eqs. (20)–(22), it is obvious that S_w, S_b and S_t are all semi-positive definite Hermite matrices, together with the proved theorem that each eigenvalue of Hermite matrix is a real number (Ding and Cai 1995), we have the

following corollary: the eigenvalues of S_w, S_b and S_t in unitary space are all nonnegative real numbers. Based on the above corollary, the GPCA is thus can be described as follows:

Let $\mathbf{v}_1, \cdots, \mathbf{v}_n$ be the orthogonal eigenvectors of S_t, $\lambda_1, \cdots, \lambda_n$ be the associated eigenvalues, which satisfy. $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.. By choosing the first m-maximal eigenvectors as projection axes, a given feature vector $\mathbf{z}$ can be projected to a m-dimensional vector $\mathbf{f}$ as follows:

$$f = \Phi^H \mathbf{z} \quad \text{where } \Phi = (\mathbf{v}_1, \cdots, \mathbf{v}_m). \tag{23}$$

Because the above process is conducted in the complex space, we call it generalized principal component analysis (GPCA). The dimensionality-reduced vector f, rather than the original combined feature vector $\mathbf{z}$, is then used for classification/prediction. Note that the dimensionality-reduced vector f also lies in a unitary space.

When the complex feature space degenerates to a real space, the GPCA is in fact the classic PCA. In other words, the GPCA suites both the real space and the complex space, and the classic PCA is only a special case of the GPCA. It is also worth noting that the parameter m in GPCA, i.e., the dimensionality of the reduced feature vector, will also affect the prediction accuracy, thus the selection of parameter m will be further discussed in the subsequent section.

Framework for protein attribute prediction

Workflow

The workflow of the proposed framework is illustrated in Fig. 1. For a given protein, its features from two different views are first extracted. Then, the two features are parallely combined and further processed by GPCA. Thirdly, the dimensionality-reduced complex feature is classified by a prediction model. Note that the prediction model also needs to be generalized into a complex space, and we will discuss this point in the following section.

Prediction model in a complex space

It has been widely acknowledged that the overall prediction performance depends not only on the feature's discriminative ability but also on the classifier to be used. As we focus on evaluating the discriminative abilities of serial and parallel feature fusion methods, in this paper, we tested the two fusion methods with 3 popular classifier models: nearest neighbor (1NN) classifier, optimized evidence theoretic K nearest neighbor (OET-KNN) classifier (Zouhal and Denoeux 1998), and the Naive Bayesian classifier (Domingos and Pazzani 1997). In general, any prediction (classification) model in the real space can be extended to the complex space. The major difference is the similarity measure. For example, a commonly used similarity metric is the Euclidian distance in the real space, while the unitary distance should be used in the complex space. As for an illustration, here we briefly introduce the OET-KNN model in the complex space as follows:

the optimized evidence-theoretic K-nearest neighbor prediction algorithm (OET-KNN) (Zouhal and Denoeux 1998) is to perform prediction based on the evidence theory, which has been demonstrated successful in dealing with biological problems. However, as the dimensionality-reduced feature space is also a complex space in the proposed method, as shown in Fig. 1, the OET-KNN cannot be directly applied. Fortunately, as long as we appropriately replace the Euclidean distance with the unitary distance, OET-KNN can then be applied to the complex space. A critical step in OET-KNN classifier is to compute the evidence of any pattern $\mathbf{f}_i$ in the training dataset, i.e., the knowledge that pattern $\mathbf{f}_i$ belongs to class ω_j is a piece

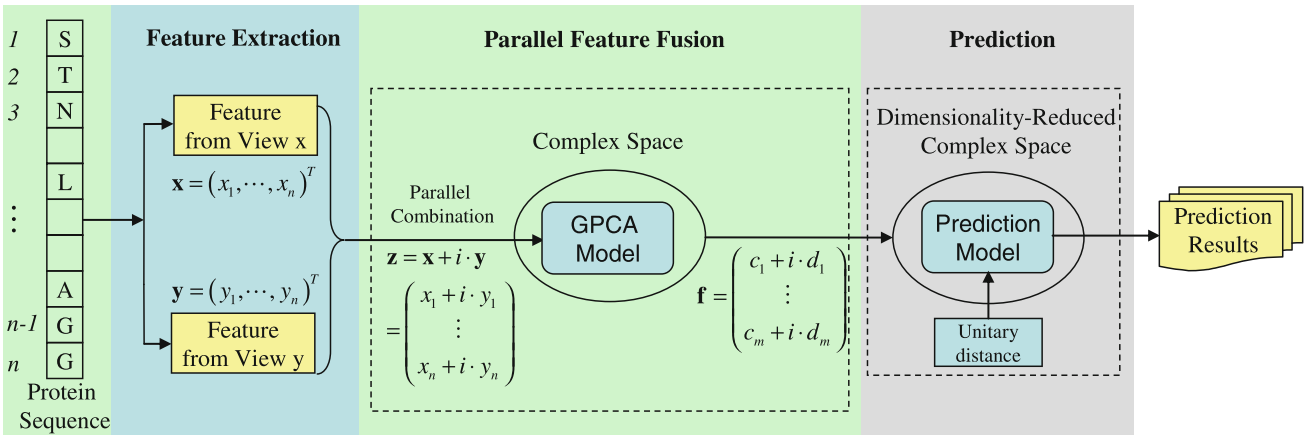

Fig. 1 Workflow of the proposed framework. Detailed description in the text

of evidence which increase our belief that the tested pattern $\mathbf{f}$ also belongs to class ω_j. This evidence is quantified by an evidence function as follows:

$$E(\mathbf{f}|\mathbf{f}_i,\omega_j) = \exp\left[-C_\omega \cdot d^2(\mathbf{f},\mathbf{f}_i)\right] \cdot \delta(\phi_i,\omega_j) \quad (24)$$

where $d(\mathbf{f},\mathbf{f}_i)$ is the Euclidean distance between $\mathbf{f}$ and $\mathbf{f}_i$, and for detailed information of other symbols, please refer to (Zouhal and Denoeux 1998). Based on Eq. (24), we can then compute this piece of evidence in the complex space by replacing the original Euclidean distance with the unitary distance as:

$$\begin{aligned} &\exp\left[-C_\omega \cdot d^2(\mathbf{f},\mathbf{f}_i)\right] \cdot \delta(\phi_i,\omega_j) \\ &\Rightarrow \exp\left[-C_\omega \cdot \left(\sqrt{(\mathbf{f}-\mathbf{f}_i)^H(\mathbf{f}-\mathbf{f}_i)}\right)^2\right] \cdot \delta(\phi_i,\omega_j) \end{aligned} \quad (25)$$

where $\sqrt{(\mathbf{f}-\mathbf{f}_i)^H(\mathbf{f}-\mathbf{f}_i)}$ is the unitary distance between two complex vectors as defined in Eq. (19).

Experimental results and discussions

The independent dataset test, sub-sampling or K-fold cross-validation test, and leave-one-out cross-validation test are three often used methods for evaluating the effectiveness of a predictor (Chou and Zhang 1995; Frishman 2010). These methods have been extensively discussed (Smialowski et al. 2010; Frishman 2010) and leave-one-out cross-validation test is considered as the most objective evaluation method (Huang et al. 2011). Accordingly, the leave-one-out test has been widely used by investigators to examine the quality of various predictors (Esmaeili et al. 2010; Hayat and Khan 2012; Mohammad Beigi et al. 2011; Chou and Shen 2010). In this study, the leave-one-out cross-validation test was also taken to evaluate the performance of the proposed method.

Results on 4 structural classes datasets

Influences of the reduced dimensionality

As described in above section, a serially or parallely combined feature vector will be firstly projected to a lower m-dimensional feature vector by applying PCA or GPCA (i.e., serial fusion or parallel fusion), and then the dimensionality-reduced feature vector is used for prediction. However, the optimal value of m is classifier-dependent. How to choose the parameter m is still a theoretically unresolved problem. In this section, the influences of the reduced dimensionality of the fused feature on overall prediction accuracy are empirically investigated on the benchmark dataset W1189 with different classifiers. More specifically, for each of the three chosen classifiers, we vary m from 2 to 60 with a step size of 2 and plot the overall prediction accuracy versus m. The performance comparison of parallel and serial fusion with classifiers 1NN, OET-KNN, and naïve Bayesian are shown in Figs. 2, 3 and 4, respectively. It is easy to find that the performance of parallel fusion is consistently superior to that of serial fusion from all the three tested classifiers on the benchmark dataset W1189 except for few occasions.

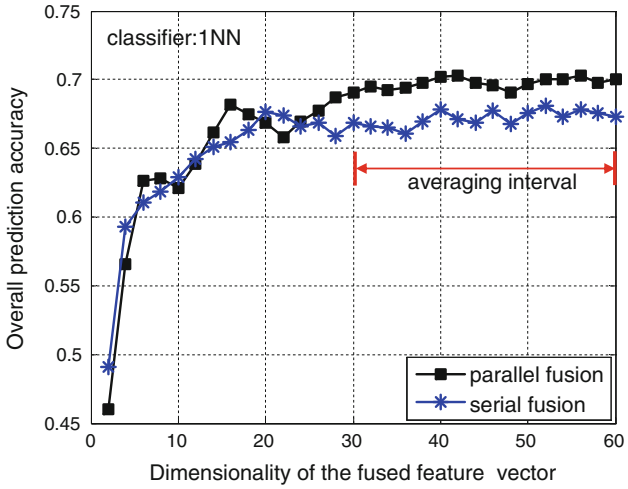

Fig. 2 Influences of the reduced dimensionality of the fused feature on prediction accuracy on benchmark dataset W1189 with one nearest neighbor (1NN) classifier

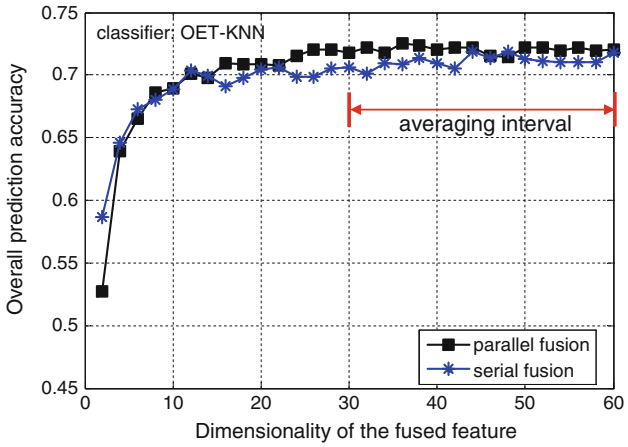

Fig. 3 Influences of the reduced dimensionality of the fused feature on prediction accuracy on benchmark dataset W1189 with optimized evidence theoretic K nearest neighbor (OET-KNN) classifier

Taking Fig. 2 as an example for detailed analysis, we can find that the overall prediction accuracies are consistently increasing when m varies from 2 to 16 for both serially and parallely fused features when applying 1NN classifier. When m is greater than 30, the prediction accuracy will not increase much but slightly fluctuate with the

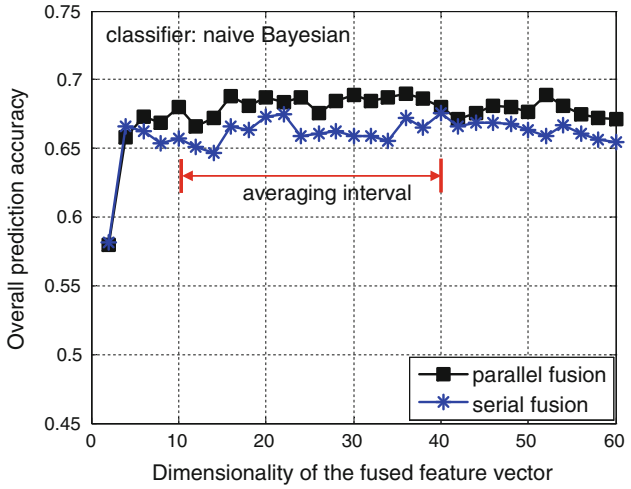

Fig. 4 Influences of the reduced dimensionality of the fused feature on prediction accuracy on benchmark dataset W1189 with naive Bayesian classifier

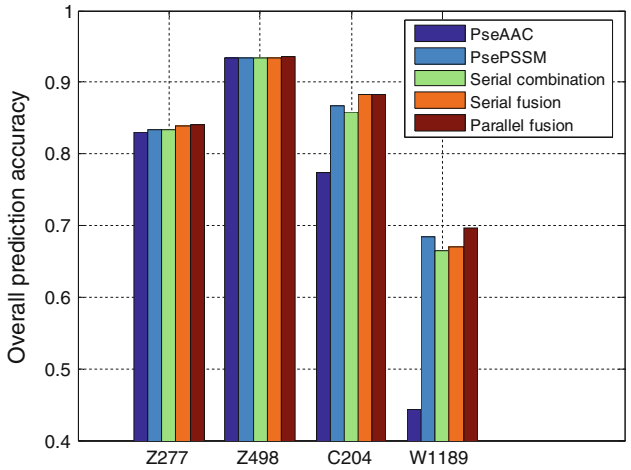

Fig. 5 Overall prediction accuracy comparisons of the five features on the four benchmark datasets with one nearest neighbor (1NN) classifier

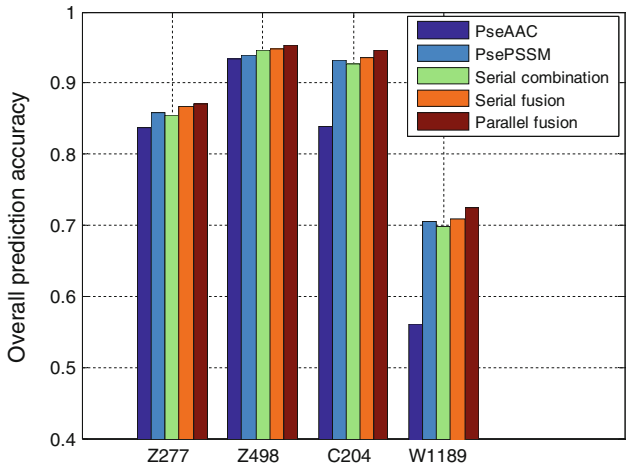

Fig. 6 Overall prediction accuracy comparison of the five feature groups across four datasets with optimized evidence theoretic *K* nearest neighbor (OET-KNN) classifier. PseAAC and PsePSSM stand for pseudo amino acid composition feature and pseudo position-specific score matrix feature, respectively

increment of m for both serial and parallel fusions. The prediction accuracy of parallel fusion is consistently high than that of the serial fusion when m is greater than 24. In addition, as can be seen from Fig. 2, the optimal value of m for parallel fusion is 42, corresponding to an overall accuracy of 70.33 %; while the optimal value for serial fusion is 52, corresponding to an overall accuracy of 68.13 %. Thus, reporting the performance comparison with a fixed value of m is unfair. Consequently, we would rather report the averaged performance comparison. Specifically, for 1NN classier, the averaged prediction accuracy for parallel or serial fusion is obtained by varying m from 30 to 60 and averaging the corresponding prediction accuracies. As aforementioned, the optimal value of m is classifier-dependent. Together with the empirically results shown in Figs. 3 and 4, we set the averaging intervals of m for classifiers OET-KNN and naïve Bayesian to be [30 60] and [10 40], respectively.

Comparisons between serial and the proposed parallel feature fusion

Figures 5, 6 and 7 intuitively illustrate the performance comparison of the five features (PseAAC, PsePSSM, serial combination of PseAAC and PsePSSM, serial fusion, and parallel fusion) on the four benchmark datasets when applying 1NN, OET-KNN, and naïve Bayesian, respectively. It is worth pointing out that only the performances of serial fusion and parallel fusion are shown in Fig. 7. The reason is that with Naive Bayesian classifier, the critical step is to compute the inverse of the covariance matrix. When the dimensionality of feature vector is high and the number of samples is small, the covariance matrix is singular thus the inverse of the covariance matrix is especially difficult to compute. In the presented benchmark datasets, the number of protein samples is small and the dimensionality of the PseAAC or PsePSSM feature vector is high (serial combination of PseAAC and PsePSSM will further increase the dimensionality), thus the naïve Bayesian cannot be applied. Fortunately, serial or parallel fusion can significantly reduce the dimensionality of the combined feature vector, while retaining its discriminative capability, thus the naïve Bayesian can be applied to compare these two methods.

Table 1 lists the detailed performance comparison of the five features on the four benchmark datasets with OET-KNN classifier. From Table 1, together with Figs. 5, 6 and 7, we can draw the following conclusions:

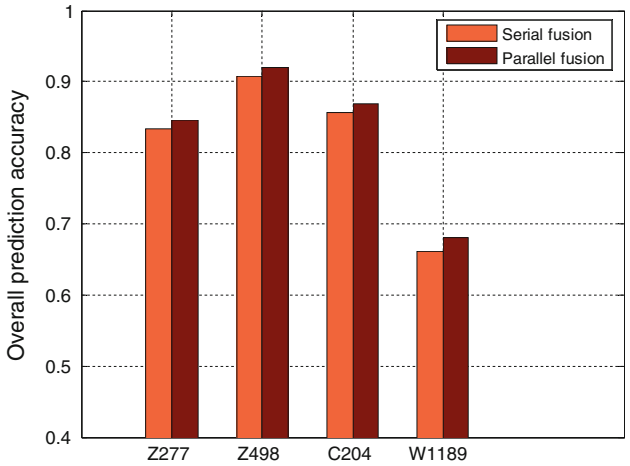

Fig. 7 Overall prediction accuracy comparison of the serial fusion and the parallel fusion on the four benchmark datasets with Naive Bayesian classifier

Table 1 Performance of comparison of the five features across the four benchmark datasets with optimized evidence theoretic *K* nearest neighbor (OET-KNN) classifier

Dataset	Feature	Prediction accuracy (%)				
		All-α	All-β	α/β	α + β	Overall
Z277	PseAAC	79.45	87.30	87.65	80.00	83.75
	PsePSSM	80.24	84.84	91.02	88.46	85.92
	Serial combination	83.78	84.61	88.61	84.75	85.56
	Serial fusion	85.93	85.94	90.24	83.33	86.64
	Parallel fusion	86.11	87.30	91.25	82.26	**87.00**
Z498	PseAAC	96.91	92.48	91.67	93.55	93.37
	PsePSSM	88.03	94.53	97.71	95.08	93.98
	Serial combination	93.52	96.80	95.56	92.31	94.58
	Serial fusion	93.46	95.35	96.99	93.02	94.78
	Parallel fusion	95.23	97.60	98.47	90.51	**95.38**
C204	PseAAC	95.92	93.10	72.92	71.43	83.82
	PsePSSM	98.00	96.83	89.13	86.67	93.14
	Serial combination	100.00	96.67	89.36	83.33	92.65
	Serial fusion	100.00	98.36	85.71	88.88	93.63
	Parallel fusion	100.00	98.39	87.50	91.11	**94.61**
W1189	PseAAC	56.08	59.71	58.93	33.00	56.14
	PsePSSM	70.33	75.16	75.43	52.81	70.51
	Serial combination	70.59	76.99	72.36	50.28	69.78
	Serial fusion	73.55	76.62	72.58	51.83	70.88
	Parallel fusion	73.25	77.26	76.34	54.91	**72.53**

PseAAC and PsePSSM stand for pseudo amino acid composition feature and pseudo position-specific score matrix feature, respectively. All values were calculated using leave-one-out cross-validation

The highest overall accuracy is highlighted in bold

(A) Serial combination of different features will occasionally, but not definitely lead to the improvement of prediction accuracy. For example, the prediction accuracy of the serially combined features lies between that of the individual features (PseAAC and PsePSSM) on all the four benchmark datasets (see Fig. 5) when 1NN is applied. The prediction accuracy of the serially combined feature lies between that of the individual features (PseAAC and PsePSSM) on all the four benchmark datasets except for dataset Z498 (see Fig. 6) when OET-KNN is applied. Taking the results on C204 with OET-KNN classifier (see Table 1) as an example, the prediction accuracies of the PseAAC and PsePSSM features are 83.82 and 93.14 %, respectively, while the prediction accuracy of the serially combined feature is 92.65 %, which is obviously inferior to that of PsePSSM feature.

(B) In most cases, performance of serial fusion is better than that of serial combination. By observing Figs. 5, 6, and Table 1, it is easy to find that performance of serial fusion is superior to that of serial combination only except on W1189 with 1NN classifier (see Fig. 5). Taking the results on W1189 with OET-KNN classifier (see Table 1) as an example, the prediction accuracies of the serial fusion is 70.88 %, while the prediction accuracy of the serially combined feature is 69.78 %, about 1 % improvement is achieved; while a 2 % improvement is achieved on W1189 with naïve Bayesian classifier (see Fig. 7).

(C) The overall performance of parallel fusion is better than that of serial fusion. On all the four benchmark datasets and three adopted classifiers, the overall prediction accuracies of parallel fusion is consistently higher than that of serial fusion. Taking the results on W1189 with OET-KNN (see Table 1) as an example, the parallel fusion achieves an overall accuracy of 72.53 %, which is about 1.65 % higher than that of serial fusion. The reason why parallel fusion outperforms serial fusion can be explained as follows: in serial fusion, the dimensionality of serial combined feature space equals to the sum of the dimensionalities of individual feature spaces; while in parallel fusion, the dimensionality of parallel combined feature space equals to the max of the dimensionalities of individual feature spaces. When the training samples are limited, accurately estimating the scatter matrix in a high dimensional feature space is more difficult than in a low one. This could be the main reason that why parallel fusion is superior to serial fusion in the presented experiments.

Performance comparison with existing predictors

Tables 2, 3, 4, and 5 illustrate the performance comparison of the proposed method (with OET-KNN classifier) with the most recently reported protein structural class prediction methods. It is found that the proposed method achieves satisfactory prediction accuracy on all the four benchmark datasets.

Table 2 Performance comparison of different methods on the Z277 dataset

Method	Prediction accuracy (%)				
	All-α	All-β	α/β	α + β	Overall
Component coupled (Zhou 1998)	84.3	82.0	81.5	67.7	79.1
Rough sets (Cao et al. 2006)	77.1	77.0	93.8	66.2	79.4
Information-theoretical approach (Zheng et al. 2010)	87.1	80.3	93.8	67.7	83.0
LogitBoost (Feng et al. 2005)	81.4	88.5	92.6	72.3	84.1
IGA-SVM (Li et al. 2008)	84.3	88.5	92.6	70.7	84.5
NN-CDM (Liu et al. 2010b)	80.0	86.4	91.6	81.8	85.2
CWT-PCA-SVM (Li et al. 2009)	85.7	90.2	87.7	80.1	85.9
SVM fusion (Chen et al. 2006b)	85.7	90.2	93.8	80.0	**87.7**
COMSPA of this paper	86.11	87.30	91.25	82.26	87.00

The results of COMSPA were calculated using leave-one-out cross-validation

The highest overall accuracy is highlighted in bold

Table 3 Performance comparison of different methods on the Z498 dataset

Method	Prediction accuracy (%)				
	All-α	All-β	α/β	α + β	Overall
Component coupled (Zhou 1998)	93.5	88.9	90.4	84.5	89.2
Rough sets (Cao et al. 2006)	87.9	91.3	97.1	86.0	90.8
SVM fusion (Chen et al. 2006b)	99.1	96.0	80.9	91.5	91.4
Information-theoretical approach (Zheng et al. 2010)	95.3	93.7	97.8	88.3	93.8
NN-CDM (Liu et al. 2010b)	96.3	93.7	95.6	89.9	93.8
IGA-SVM (Li et al. 2008)	96.3	93.6	97.8	89.2	94.2
LogitBoost (Feng et al. 2005)	92.6	96.0	97.1	93.0	94.8
CWT-PCA-SVM (Li et al. 2009)	94.4	96.8	97.0	92.3	95.2
MODAS (Mizianty and Kurgan 2009)	96.7	97.5	95.6	97.1	**96.8**
COMSPA of this paper	95.23	97.60	98.47	90.51	95.38

The results of COMSPA were calculated using leave-one-out cross-validation

The highest overall accuracy is highlighted in bold

Table 2 lists the performance comparison of different methods on the benchmark dataset Z277. The proposed method achieves the second-best prediction performance among the listed methods with an overall accuracy of 87.0 %, which is only 0.7 % lower than the first-best one (SVM fusion (Chen et al. 2006b)) and is 1.1 % better than the third-best one [CWT-PCA-SVM (Li et al. 2009)]. On the benchmark datasets Z498 and C204, the proposed method achieves the second-best and the best performances with an overall accuracy of 95.38 and 94.61 %, respectively, as shown in Tables 3 and 4. Taking the results listed in Table 4 as an example, the proposed method achieves the best prediction performance with an overall accuracy of 94.61 %, which is 3.41 % better than the second-best method NN-CDM (Liu et al. 2010b). Table 5 lists the performance comparison on the W1189 dataset. It is found that the proposed method achieves the fourth best prediction performance with an overall accuracy of 72.53 %. By careful analysis, we found that the highest three performers, i.e., SCPRED (Kurgan et al. 2008), MODAS (Mizianty and Kurgan 2009), and RKS-PPSC (Yang et al. 2010), all used the predicted secondary structure information into their protein structural prediction procedures. Because the residue secondary structure is directly related with the structural classification task, they thus achieved high performance. When compared with other predictors based only on the sequential features, the proposed method still performs the best (72.53 %) and is 1.83 % better than the second-best performer among all the listed methods that do not utilize sequence-derived structural information.

Results on disulfide connectivity dataset: an example of parallely fusing more than 2 different features

Disulfide bonds, formed by the cysteine pairs, play important roles in stabilizing the protein structures by forming long-range constraints (Fariselli and Casadio 2001). Correctly predicting the disulfide bonds from the

Table 4 Performance comparison of different methods on the C204 dataset

Method	Prediction accuracy (%)				
	All-α	All-β	α/β	α + β	Overall
Supervised fuzzy clustering (Shen et al. 2005)	73.1	90.2	62.2	63.1	73.5
LogitBoost (Cai et al. 2006)	90.4	88.5	80.0	73.9	83.8
Augmented covariant discriminant algorithm (Xiao et al. 2006)	82.7	90.2	100.0	87.0	89.7
SVM (Chen et al. 2006a)	88.5	96.7	77.8	73.9	85.3
WSVM (Qiu et al. 2009)	86.5	82.0	91.1	91.3	87.3
Multi-features fusion (Chen et al. 2008a)	92.3	93.4	95.6	78.3	90.2
Binary-tree SVM (Zhang and Ding 2007)	90.4	100.0	97.8	73.9	91.2
NN-CDM (Liu et al. 2010b)	88.5	100.0	97.8	76.1	91.2
COMSPA of this paper	100.00	98.39	87.50	91.11	**94.61**

The results of COMSPA were calculated using leave-one-out cross-validation

The highest overall accuracy is highlighted in bold

Table 5 Performance comparison of different methods on the W1189 dataset

Method	Prediction accuracy (%)				
	All-α	All-β	α/β	α + β	Overall
Logistic regression (Kurgan and Homaeian 2006)	57.0	62.9	64.7	25.3	53.9
FKNN classifier (Zhang et al. 2008)	48.9	59.5	81.7	26.6	56.9
WSVM (Qiu et al. 2009)	–	–	–	–	59.2
Specific tri-peptides (Costantini and Facchiano 2009)	–	–	–	–	59.9
SVM (Chen et al. 2008b)	75.8	75.2	82.6	31.8	67.6
AADP-PSSM (Liu et al. 2010a)	69.1	83.7	85.6	35.7	70.7
SCPRED (Kurgan et al. 2008)	89.1	86.7	89.6	53.8	80.6
RKS-PPSC (Yang et al. 2010)	89.2	86.7	82.6	65.6	81.3
MODAS (Mizianty and Kurgan 2009)	92.3	87.1	87.9	65.4	**83.5**
COMSPA of this paper	73.25	77.26	76.34	54.91	72.53

The results of COMSPA were calculated using leave-one-out cross-validation

The highest overall accuracy is highlighted in bold

amino acid sequences is considered one of the important tasks in the ab initio protein structure modeling. The usual steps for prediction of disulfide bridges from the primary sequences are encoding the cysteine residues with sequential features, then combining the two feature vectors of the two considered cysteines, and finally inputting the combined feature vector to a machine learning model for predictions. All the reported methods adopted the serial combination strategy for the two feature vectors of the two cysteines, e.g., simply concatenating one after the other. However, not all the features that can be calculated are useful for the prediction. For example, selecting features which are used by the machine learning algorithm was demonstrated to be helpful in improving the performance (Zhu et al. 2010). It is also important to remember that improper evaluation of feature selection leads to overfitting (Smialowski et al. 2010). When considering the equal roles of the two vectors of two considered cysteines, we thus compared the performances of serial and parallel feature fusion approaches in disulfide bridge predictions.

We take the dataset constructed in Fariselli and Casadio (2001) for a benchmark testing, which consists of 446 proteins and 1,371 disulfide bonds. Three different local features are used to encode the cysteine residues, i.e., predicted secondary structure of the residues (SS) by PSIPRED (Jones 1999), PSI-BLAST-determined evolutionary information encoded with position-specific scoring matrix (PSSM), and cysteine separation distance. The SS feature is the probabilities of residues being the structural states of helix, strand, and coil. We use a window of length 13 to encode the local SS and PSSM information, which is centered on the cysteine. Then we can get a vector of $13 \times (20 + 3) = 299$ components for each cysteine. The cysteine separation distance is calculated as the number of residues between the two considered cysteines along the sequence, which is 1 component. Thus, in the serial

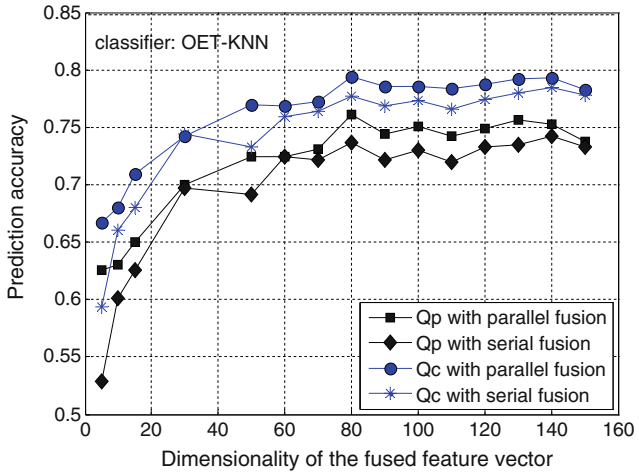

Fig. 8 Influences of the reduced dimensionality of the fused feature on prediction accuracy. Q_c is the percent of disulfide bonds which are correctly predicted, and Q_P is the percent of proteins whose disulfide connectivity patterns are all correctly predicted

combination case, we represent a cysteine–cysteine pair with a 599-dimensional vector; while in the parallel combination case, we will have a 300-dimensional complex vector. The feature vector is firstly projected to a lower dimensional feature vector by applying PCA or GPCA, respectively, on the two serial and parallel cases, and then the dimensionality-reduced feature vector is inputted to the OET-KNN predictor. Two widely used criteria Q_c and Q_P are used to evaluate the performances, where Q_c is the percent of disulfide bonds which are correctly predicted, and the Q_P is the percent of proteins whose disulfide connectivity patterns are all correctly predicted in the dataset. Figure 8 illustrates that the prediction accuracies evaluated by Q_C and Q_P for parallel fusion are better than those from serial fusion. Taking Q_P as an example, we can achieve the best performer of Q_P of 76.1 % at the reduced dimension of 80 for parallel fusion. These results demonstrate that parallel fusion of features of two cysteines is promising for enhancing the performance of disulfide bonds predictions.

Discussions and conclusions

In this study, we have developed a framework, in which parallel feature fusion is used for protein attribute prediction. Features from different views are first parallely combined to form a complex feature space, and then the generalized principal component analysis is applied in the obtained complex feature space to perform further feature extraction. The better performance of parallel feature fusion is derived from the merit that the dimensionality of the parallely combined feature space will not increase as that of the serially combined feature space, and thus the scatter matrix of training samples can be more accurately estimated. Experimental results of protein structural class predictions on four benchmark datasets and disulfide connectivity predictions all show that the proposed framework outperforms the traditional serial feature fusion. The proposed method enriches the content of protein attribute prediction and is flexible to suit for other problems in bioinformatics.

In the presented study, we first perform parallel feature fusion for two different sequential features that have already yielded promising results. When the number of features is more than two, a two-stage solution can be applied: dividing futures into two groups and features in each group are serially combined; the two serially combined features are then parallely combined and further processed by GPCA. The experiments on disulfide connectivity predictions have given such an example. All these results demonstrate that the proposed parallel fusion approaches are flexible for dealing with different real-world applications.

It is worth pointing out that in some cases, the feature level fusion and the decision level fusion are not totally independent of each other. They can be switched to each other. For example, instead of feature fusion, we can train single classifier on each type of feature, and then perform the decision fusion. Similarly, the decision fusion can thus be changed to a feature fusion. The performances of the two strategies are dependent on detailed applications. Although this is true, feature level fusion is necessary in analyzing many biological problems; especially, when considering the interactions between two biological components or molecules. For example, in the disulfide connectivity predictions, we have to consider the features from two cysteines, where a proper feature fusion strategy is important. Take ab initio protein structure prediction as another example, the bottleneck problem is correctly predicting the residue–residue interactions from primary sequence. Even the most accurate state-of-the-art contact prediction algorithms can only give $\sim$20–30 % accuracy, which significantly limit the structure resolution thus modeled (Wu et al. 2011; Zhang 2009). Other examples include protein–protein interaction predictions, membrane protein transmembrane helix–helix interactions, and etc. In considering of these, the parallel feature fusion algorithms proposed in this study will play important roles for improving two components contact prediction accuracies.

In fact, an automated prediction system's performance will be affected by many factors including the feature organization discussed in this paper. Other factors include classification algorithms and even the dataset itself. This study mainly focuses on investigating the difference between serial and parallel feature fusion methods, and the possibility of applying parallel feature fusion method to

bioinformatics problems. Currently, we have demonstrated the proposed parallel feature fusion method with several simple but popular classifiers such as nearest neighbor and Bayesian-based algorithm. The effectiveness of the proposed parallel feature fusion method for more sophisticated classifiers, such as SVM need to be further studied. We plan to investigate this point in near future.

A web server for protein structural class prediction based on the proposed method, called COMSPA (abbreviation of the first three characters of complex and space), has been implemented by Java Server Pages (JSP) programming language and is freely available at: http://www.csbio.sjtu.edu.cn/bioinf/COMSPA/ for academic use.

Acknowledgments The authors wish to thank the three anonymous reviewers whose constructive comments are very helpful for strengthening the presentation of this paper. This work was supported by the National Natural Science Foundation of China (61222306, 61233011, 91130033, 61175024), the Natural Science Foundation of Jiangsu (BK2011371), Jiangsu Postdoctoral Science Foundation, (No.1201027C), the National Science Fund for Distinguished Young Scholars (61125305), Shanghai Science and Technology Commission (11JC1404800), a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (201048), and Program for New Century Excellent Talents in University (NCET-11-0330).

Conflict of interest The authors declare that they have no conflict of interest.

References

Cai YD, Feng KY, Lu WC, Chou KC (2006) Using LogitBoost classifier to predict protein structural classes. J Theor Biol 238(1):172–176

Cao Y, Liu S, Zhang L, Qin J, Wang J, Tang K (2006) Prediction of protein structural class with rough sets. BMC Bioinformatics 7:20

Chen K, Kurgan L (2007) PFRES: protein fold classification by using evolutionary information and predicted secondary structure. Bioinformatics 23(21):2843–2850

Chen C, Tian YX, Zou XY, Cai PX, Mo JY (2006a) Using pseudo-amino acid composition and support vector machine to predict protein structural class. J Theor Biol 243(3):444–448

Chen C, Zhou X, Tian Y, Zou X, Cai P (2006b) Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network. Anal Biochem 357(1):116–121

Chen C, Chen LX, Zou XY, Cai PX (2008a) Predicting protein structural class based on multi-features fusion. J Theor Biol 253(2):388–392

Chen K, Kurgan L, Ruan J (2008b) Prediction of protein structural class using novel evolutionary collocation-based sequence representation. J Comput Chem 29(19):1596–1604

Chou KC (1999) A key driving force in determination of protein structural classes. Biochem Biophys Res Commun 264(1):216–224

Chou KC (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins 43(3):246–255

Chou KC, Shen HB (2007) MemType-2L: a Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem Biophys Res Comm 360:339–345

Chou KC, Shen HB (2010) Plant-mPLoc: a top-down strategy to augment the power for predicting plant protein subcellular localization. PLoS ONE 5(6):e11335

Chou KC, Zhang CT (1995) Prediction of protein structural classes. Crit Rev Biochem Mol Biol 30(4):275–349

Costantini S, Facchiano AM (2009) Prediction of the protein structural class by specific peptide frequencies. Biochimie 91(2):226–229

Dasigi V, Mann RC, Protopopescu VA (2001) Information fusion for text classification—an experimental comparison. Pattern Recogn 34(12):2413–2425

Dima RI, Thirumalai D (2004) Proteins associated with diseases show enhanced sequence correlation between charged residues. Bioinformatics 20(15):2345–2354

Ding XR, Cai MK (1995) Matrix theory in engineering. Tianjin University Press, Tianjin

Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29(2–3):103–130

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(2):203–209

Fariselli P, Casadio R (2001) Prediction of disulfide connectivity in proteins. Bioinformatics 17(10):957–964

Feng KY, Cai YD, Chou KC (2005) Boosting classifier for predicting protein domain structural class. Biochem Biophys Res Commun 334(1):213–217

Frishman D (2010) Structural bioinformatics of membrane proteins. Springer, New York

Gao J, Zhang T, Zhang H, Shen S, Ruan J, Kurgan L (2010) Accurate prediction of protein folding rates from sequence and sequence-derived residue flexibility and solvent accessibility. Proteins 78(9):2114–2130

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(4):411–421

Huang T, Chen L, Cai YD, Chou KC (2011) Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property. PLoS ONE 6(9):e25297

Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 292(2):195–202

Jeong JC, Lin X, Chen XW (2011) On position-specific scoring matrix for protein function prediction. IEEE/ACM Trans Comput Biol Bioinform 8(2):308–315

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324

Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley, Hoboken

Kurgan L, Disfani FM (2011) Structural protein descriptors in 1-dimension and their sequence-based predictions. Curr Protein Pept Sci 12(6):470–489

Kurgan LA, Homaeian L (2006) Prediction of structural classes for protein sequences and domains—impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. Pattern Recogn 39(12):2323–2343

Kurgan L, Cios K, Chen K (2008) SCPRED: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. BMC Bioinformatics 9:226

Levitt M, Chothia C (1976) Structural patterns in globular proteins. Nature 261(5561):552–558

Li ZC, Zhou XB, Lin YR, Zou XY (2008) Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. Amino Acids 35(3):581–590

Li ZC, Zhou XB, Dai Z, Zou XY (2009) Prediction of protein structural classes by Chou's pseudo amino acid composition:

approached using continuous wavelet transform and principal component analysis. Amino Acids 37(2):415–425
Liu L, Cai Y, Lu W, Feng K, Peng C, Niu B (2009) Prediction of protein–protein interactions based on PseAA composition and hybrid feature selection. Biochem Biophys Res Commun 380(2):318–322
Liu T, Zheng X, Wang J (2010a) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. Biochimie 92(10):1330–1334
Liu TG, Zheng XQ, Wang J (2010b) Prediction of protein structural class using a complexity-based distance measure. Amino Acids 38:721–728
Mizianty MJ, Kurgan L (2009) Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. BMC Bioinformatics 10:414
Mizianty MJ, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification and production propensity. Bioinformatics 27(13):i24–i33
Mohammad Beigi M, Behjati M, Mohabatkar H (2011) Prediction of metalloproteinase family based on the concept of Chou's pseudo amino acid composition using a machine learning approach. J Struct Funct Genomics 12(4):191–197
Nanni L, Lumini A (2008) Combing ontologies and dipeptide composition for predicting DNA-binding proteins. Amino Acids 34(4):635–641
Pearson K (1901) On lines and planes of closest fit to systems of points in space. Phil Mag 2(6):559–572
Pierleoni A, Martelli PL, Casadio R (2011) MemLoci: predicting subcellular localization of membrane proteins in eukaryotes. Bioinformatics 27(9):1224–1230
Qiu JD, Luo SH, Huang JH, Liang RP (2009) Using support vector machines for prediction of protein structural classes based on discrete wavelet transform. J Comput Chem 30(8):1344–1350
Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517
Schaffer AA (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res 29:2994–3005
Shen HB, Chou KC (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem 373(2):386–388
Shen HB, Yang J, Liu XJ, Chou KC (2005) Using supervised fuzzy clustering to predict protein structural classes. Biochem Biophys Res Commun 334(2):577–581
Shen HB, Yang J, Chou KC (2007) Euk-PLoc: an ensemble classifier for large-scale eukaryotic protein subcellular location prediction. Amino Acids 33(1):57–67
Smialowski P, Schmidt T, Cox J, Kirschner A, Frishman D (2006) Will my protein crystallize? A sequence-based predictor. Proteins 62(2):343–355
Smialowski P, Martin-Galiano AJ, Cox J, Frishman D (2007a) Predicting experimental properties of proteins from sequence by machine learning techniques. Curr Protein Pept Sci 8(2): 121–133
Smialowski P, Martin-Galiano AJ, Mikolajka A, Girschick T, Holak TA, Frishman D (2007b) Protein solubility: sequence based prediction and experimental verification. Bioinformatics 23(19):2536–2542
Smialowski P, Frishman D, Kramer S (2010) Pitfalls of supervised feature selection. Bioinformatics 26(3):440–443
Tsai CH, Chan CH, Chen BJ, Kao CY, Liu HL, Hsu JP (2007) Bioinformatics approaches for disulfide connectivity prediction. Curr Protein Pept Sci 8(3):243–260
Ulug ME, McCullough CL (1999) Feature and data-level fusion of infrared and visual images SPIE Conference on Sersor Fusion: architectures, algorithms and applications III vol. 3719:312–318
Wang ZX, Yuan Z (2000) How good is prediction of protein structural class by the component-coupled method? Proteins: Struct, Func, Bioinformatics 38(2):165–175
Wu S, Szilagyi A, Zhang Y (2011) Improving protein structure prediction using multiple sequence-based contact predictions. Structure 19(8):1182–1191
Xiao X, Shao SH, Huang ZD, Chou KC (2006) Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. J Comput Chem 27(4):478–482
Yang J, Yang JY, Zhang D, Lu JF (2003) Feature fusion: parallel strategy versus serial strategy. Pattern Recogn 36(6):1369–1381
Yang JY, Peng ZL, Chen X (2010) Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. BMC Bioinformatics 11(Suppl 1):S9
Yu DJ, Shen HB, Yang JY (2011) SOMRuler: a novel interpretable transmembrane helices predictor. IEEE Trans on Nanobiosci 10(2):119–121
Zhang Y (2009) Protein structure prediction: when is it useful? Curr Opin Struct Biol 19(19):145–155
Zhang TL, Ding YS (2007) Using pseudo amino acid composition and binary-tree support vector machines to predict protein structural classes. Amino Acids 33(4):623–629
Zhang TL, Ding YS, Chou KC (2008) Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern. J Theor Biol 250(1): 186–193
Zheng X, Li C, Wang J (2010) An information-theoretic approach to the prediction of protein structural class. J Comput Chem 31(6):1201–1206
Zhou GP (1998) An intriguing controversy over protein structural class prediction. J Protein Chem 17(8):729–738
Zhu L, Yang J, Song JN, Chou KC, Shen HB (2010) Improving the accuracy of predicting disulfide connectivity by feature selection. J Comput Chem 31(7):1478–1485
Zouhal LM, Denoeux T (1998) An evidence-theoretic K-NN rule with parameter optimization. IEEE Trans Syst Man Cybern 28: 263–271