Structural bioinformatics

Ensemble classifier for protein fold pattern recognition
Hong-Bin Shen¹,* and Kuo-Chen Chou¹,²
¹Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, China and ²Gordon Life Science Institute, San Diego, CA 92130, USA

Received on March 31, 2006; revised on April 26, 2006; accepted on April 27, 2006
Advance Access publication May 3, 2006
Associate Editor: Keith A Crandall

ABSTRACT
Motivation: Prediction of protein folding patterns is one level deeper than that of protein structural classes, and hence is much more complicated and difficult. To deal with such a challenging problem, the ensemble classifier was introduced. It was formed by a set of basic classifiers, with each trained in different parameter systems, such as predicted secondary structure, hydrophobicity, van der Waals volume, polarity, polarizability, as well as different dimensions of pseudo-amino acid composition, which were extracted from a training dataset. The operation engine for the constituent individual classifiers was OET-KNN (optimized evidence-theoretic k-nearest neighbors) rule. Their outcomes were combined through a weighted voting to give a final determination for classifying a query protein. The recognition was to find the true fold among the 27 possible patterns.

Results: The overall success rate thus obtained was 62% for a testing dataset where most of the proteins have <25% sequence identity with the proteins used in training the classifier. Such a rate is 6–21% higher than the corresponding rates obtained by various existing NN (neural networks) and SVM (support vector machines) approaches, implying that the ensemble classifier is very promising and might become a useful vehicle in protein science, as well as proteomics and bioinformatics.

Availability: The ensemble classifier, called PFP-Pred, is available as a web-server at http://202.120.37.186/bioinf/fold/PFP-Pred.htm for public usage.

Contact: lifesci-sjtu@san.rr.com

Supplementary information: Supplementary data are available on Bioinformatics online.

MATERIALS AND METHODS

INTRODUCTION

The avalanche of protein sequences generated in the post-genomic era has challenged us for developing computational methods by which the structural information can be timely extracted from sequence databases. Although the direct prediction of the three-dimensional (3D) structure of a protein from its sequence based on the least free energy principle is scientifically quite sound and some encouraging results already obtained in elucidating the handedness problems and packing arrangements in proteins (see e.g. Chou and Carlacci, 1991; Chou et al., 1982, 1984, 1990), it is very difficult to predict its overall fold owing to the notorious local minimum problem. Also, although it is quite successful to predict the 3D structure of a protein according to the homology modeling approach (Chou, 2004; Holm and Sander, 1999), a hurdle exists when the query protein does not have any structure-known homologous protein in the existing databases. Facing this kind of situation, can we find a different approach to predict the fold of a protein? In this paper, we shall resort to the taxonomic approach, whose underpinning is based on the assumption that the number of protein folds is limited (Chou and Zhang, 1995; Dubchak et al., 1999; Finkelstein and Ptitsyn, 1987; Murzin et al., 1995). Accordingly, predicting the 3D structure of a protein may be first converted to a problem of classification, i.e. identifying which fold pattern it belongs to. The present study was initiated in an attempt to introduce a novel approach, the ensemble classifier, to recognize the fold pattern for a query protein.

The working (training and testing) datasets studied here were taken from Ding and Dubchak (2001). The original training dataset and testing dataset contain 313 proteins and 385 proteins, respectively. Of these proteins, however, two (i.e. 2SCMC and 2GPS) in the training dataset and two (2YHX_1 and 2YHX_2) in the testing dataset do not have sequence records. These four proteins were excluded for further consideration due to lacking sequence information. Accordingly, we have 311 proteins for training dataset and 383 proteins for testing dataset. The names of the training and testing proteins and their sequences are given in Online Supplementary Materials A1 and AII, respectively. None of proteins in the training dataset has >35% sequence identity to those in the training dataset (Ding and Dubchak, 2001). According to the SCOP database (Andreeva et al., 2004; Murzin et al., 1995), the proteins in the training and testing datasets (Online Supplementary Materials A) were further classified into the following 27-fold types (Ding and Dubchak, 2001; Dubchak et al., 1995, 1999): (1) globin-like, (2) cytochrome c, (3) DNA-binding 3-helical bundle, (4) 4-helical up-and-down bundle, (5) 4-helical cytokines, (6) EF-hand, (7) immunoglobulin-like, (8) cupredoxins, (9) viral coat and capsid proteins, (10) conA-like lectin/glucanases, (11) SH3-like barrel, (12) OB-fold, (13) beta-trefoil, (14) trypsin-like serine proteases, (15) lipocalins, (16) (TIM)-barrel, (17) FAD (also NAD) binding motif, (18) flavodoxin-like, (19) NAD(P)-binding Rossmann-fold, (20) P-loop, (21) thioredoxin-like, (22) ribonuclease H-like motif, (23) hydrolases, (24) periplasmic binding protein-like, (25) beta-grasp, (26) ferredoxin-like and (27) small inhibitors, toxins, lectins, Of the above 27-fold types, types 1–6 belong to all a structural class, types 7–15 to all b class, types 16–24 to a/b class and type 25–27 to a+β class. Therefore, the classification of 27 folds is one level deeper than that of 4 structural classes (Cai, 2001; Chou and Zhang, 1995; Zhou, 1998; Zhou and Assa-Munt, 2001). Naturally, it is more challenging and difficult to conduct prediction among the 27-fold types than among the 4 structural classes (Chou, 1995; Chou and Maggiora, 1998).

To deal with the problem, Ding and Dubchak (2001) extracted the following six features from protein sequences: (1) amino acid composition, (2) predicted secondary structure, (3) hydrophobicity, (4) normalized van der Waals volume, (5) polarity and (6) polarizability. Of the above six features, only the amino acid composition contains 20 components, with each representing the occurrence frequency of one of the 20 native amino acids.
acids in a given protein (Chou and Zhang, 1994; Zhou and Doctor, 2003). For the other five features, each contains $3 + 3 + 5 \times 3 = 21$ components, as detailed in Ding and Dubchak (2001) and Dubchak et al. (1999). Based on these multiple parameter sets and majority voting rule trained by the proteins in the training dataset, an overall success rate of 56% was reported (Ding and Dubchak, 2001) in predicting the fold type for the proteins in the testing dataset.

In the present study, in order to avoid completely ignoring the sequence-order effects, the pseudo-amino acid composition (Chou, 2001) was used to replace the conventional amino acid composition (Chou and Zhang, 1993; Nakashima et al., 1986) as used in (Ding and Dubchak, 2001). However, rather than using a combined correlation function (Chou, 2001), here the alternate correlation function between hydrophobicity and hydrophilicity (Chou, 2005; Chou and Cai, 2005) is adopted to reflect the sequence-order effects. For reader’s convenience, a brief introduction about amphiphilic pseudo-amino acid composition (PseAA) is given below.

Suppose a protein $P$ with a sequence of $L$ amino acid residues:

$$R_1; R_2; R_3; R_4; \ldots; R_L,$$

where $R_1$ represents the residue at chain position 1, $R_2$ at position 2 and so forth. The hydrophobicity and hydrophilicity of the constituent amino acids in a protein play a very important role to its folding; e.g. many helices and sheets in proteins are amphiphilic that is formed by the hydrophobic and hydrophilic amino acids according to a special order along the helix chain, as illustrated by the ‘wenxiang’ diagram (Chou et al., 1997). Therefore, these two indices may be one of the optimal choices to reflect the sequence-order effects. In view of this, the sequence-order effects can be indirectly and partially, but quite effectively, reflected through the following equations (Fig. 1):

$$\begin{align*}
\tau_1 &= \frac{1}{L-1} \sum_{i=1}^{L-1} H^1_{i+1} \\
\tau_2 &= \frac{1}{L-1} \sum_{i=1}^{L-1} H^2_{i+1} \\
\tau_3 &= \frac{1}{L-2} \sum_{i=1}^{L-2} H^1_{i+2} \\
\tau_4 &= \frac{1}{L-2} \sum_{i=1}^{L-2} H^2_{i+2} \\
\tau_2L-1 &= \frac{1}{L-1} \sum_{i=1}^{L-1} H^1_{i+k} \\
\tau_2L &= \frac{1}{L-1} \sum_{i=1}^{L-1} H^2_{i+k}
\end{align*}$$

where $H^1_{i}$ and $H^2_{i}$ are the hydrophobicity and hydrophilicity correlation functions given by

$$\begin{align*}
H^1_i &= h^1(R_i) - \langle h^1 \rangle \\
H^2_i &= h^2(R_i) - \langle h^2 \rangle
\end{align*}$$

where $(h^i(R_i))$ are, respectively, the hydrophobicity and hydrophilicity values for the $i$th ($i = 1, 2, \ldots, L$) amino acid in Equation (1), and the dot $(\cdot)$ means the multiplication sign. In Equation (2) $\tau_1$ and $\tau_2$ are called the 1st-tier correlation factors that reflect the sequence-order correlation between all the most contiguous residues along a protein chain through hydrophobicity and hydrophilicity, respectively [Figure 1(a1), (a2)], $\tau_3$ and $\tau_4$ are the corresponding 2nd-tier correlation factors that reflect the sequence-order correlation between all the 2nd most contiguous residues [Figure 1(b1), (b2)], and so forth. Note that before substituting the values of hydrophobicity and hydrophilicity into Equation (3), they were all subjected to a standard conversion as described by the following equation:

$$\begin{align*}
h_1(R_i) &= \frac{h^1(R_i) - \langle h^1 \rangle}{\text{SD}(h^1)} \\
h_2(R_i) &= \frac{h^2(R_i) - \langle h^2 \rangle}{\text{SD}(h^2)}
\end{align*}$$

where the symbols $h^1_i(R_i)$ and $h^2_i(R_i)$ represent the original hydrophobicity value (Tanford, 1962) and hydrophilicity value (Hopp and Woods, 1981) for amino acid $R_i$, respectively (Table 1); $< h_i^1 >$ and $< h_i^2 >$ their mean value over 20 native amino acids; SD(h^1_i) and SD(h^2_i) their standard deviations. The converted hydrophobicity and hydrophilicity values obtained by Equation (4) will have a zero mean value over the 20 native amino acids and will remain unchanged if going through the same conversion procedure again. As we can see from Equations (1–4) as well as Figure 1, a considerable amount of sequence-order information has been incorporated into the 2A correlation factors through the hydrophobic and hydrophilic values of the amino acid residues along a protein chain. By fusing the 2A amphiphilic correlation factors into the classical amino acid composition, we have the following augmented discrete form to represent a protein sample $P$:

$$P = \begin{bmatrix}
P_1 \\
P_2 \\
P_{20} \\
P_{20+1} \\
P_{20+2} \\
P_{20+3} \\
P_{20+4} \\
P_{20+5} \\
P_{20+6} \
\end{bmatrix}$$
Recognition of protein fold patterns

Table 1. The amino acid parameters used for deriving the amphiphilic pseudo-amino acid components [cf. Equation (6)]

<table>
<thead>
<tr>
<th>Code</th>
<th>Hydrophobicity&lt;sup&gt;a&lt;/sup&gt; h&lt;sub&gt;i&lt;/sub&gt;</th>
<th>Hydrophilicity&lt;sup&gt;b&lt;/sup&gt; h&lt;sub&gt;j&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.62</td>
<td>-0.5</td>
</tr>
<tr>
<td>C</td>
<td>0.29</td>
<td>-1.0</td>
</tr>
<tr>
<td>D</td>
<td>-0.90</td>
<td>3.0</td>
</tr>
<tr>
<td>E</td>
<td>-0.74</td>
<td>3.0</td>
</tr>
<tr>
<td>F</td>
<td>1.19</td>
<td>-2.5</td>
</tr>
<tr>
<td>G</td>
<td>0.48</td>
<td>0.0</td>
</tr>
<tr>
<td>H</td>
<td>-0.40</td>
<td>-0.5</td>
</tr>
<tr>
<td>I</td>
<td>1.38</td>
<td>-1.8</td>
</tr>
<tr>
<td>K</td>
<td>-1.50</td>
<td>3.0</td>
</tr>
<tr>
<td>L</td>
<td>1.06</td>
<td>-1.8</td>
</tr>
<tr>
<td>M</td>
<td>0.64</td>
<td>-1.3</td>
</tr>
<tr>
<td>N</td>
<td>-0.78</td>
<td>2.0</td>
</tr>
<tr>
<td>P</td>
<td>0.12</td>
<td>0.0</td>
</tr>
<tr>
<td>Q</td>
<td>-0.85</td>
<td>0.2</td>
</tr>
<tr>
<td>R</td>
<td>-2.53</td>
<td>3.0</td>
</tr>
<tr>
<td>S</td>
<td>-0.18</td>
<td>0.3</td>
</tr>
<tr>
<td>T</td>
<td>-0.05</td>
<td>-0.4</td>
</tr>
<tr>
<td>V</td>
<td>1.08</td>
<td>-1.5</td>
</tr>
<tr>
<td>W</td>
<td>0.81</td>
<td>-3.4</td>
</tr>
<tr>
<td>Y</td>
<td>0.26</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

<sup>a</sup>The hydrophobicity values were taken from Tanford (1962).
<sup>b</sup>The hydrophilicity values were taken from Hopp and Woods (1981).

where

\[
p_u = \frac{\sum_{i=1}^{20} f_i}{\sum_{i=1}^{20} \sum_{j=1}^{u} \gamma_{ij}}, \quad (1 \leq u \leq 20)
\]

\[
p_u = \frac{\sum_{i=1}^{20} f_i}{\sum_{i=1}^{20} \sum_{j=1}^{u} \gamma_{ij}}, \quad (20 + 1 \leq u \leq 20 + 2a)
\]

where \(f_i (i = 1, 2, \ldots, 20)\) are the normalized occurrence frequencies of the 20 native amino acids in the protein \(P\), \(\gamma_j\) the sequence-correlation factor computed according to Equation (2) and \(\gamma\) the weight factor. In the current study, we chose \(\gamma = 0.5\) to make the results of Equation (6) within the range easier to be handled (\(\gamma\) can be of course assigned with other values, but this would not have a big different impact to the final results). Therefore, the first 20 numbers in Equation (5) represent the classic amino acid composition, and the next 2\(a\) discrete numbers reflect the amphiphilic sequence correlation along a protein chain. Such a protein representation is called ‘amphiphilic pseudo-amino acid composition’, which has the same form as the conventional amino acid composition, but contains much more information. It is through the 2\(a\) pseudo-amino acid components that the sequence order of a protein chain and the distribution of the hydrophobic and hydrophilic amino acids along the chain are indirectly and partially reflected. It should be pointed out that, according to the definition of the classical amino acid composition, all its components must be \(>0\); it is not always true, however, for the pseudo-amino acid composition: the components corresponding to the sequence correlation factors may also be \(<0\).

In this study, the OET-KNN (optimized evidence-theoretic \(k\)-nearest neighbors) algorithm is adopted as the operation engine of a classifier (Shen and Chou, 2005). For reader’s convenience, a brief introduction about OET-KNN classifier and its key equations are given in Appendix A. However, quite different from the case of (Shen and Chou, 2005), now we have many different input types, such as the (20+2\(a\))D PseAA, 21D predicted secondary structure, 21D hydrophobicity, 21D normalized van der Waals volume, 21D polarity and 21D polarizability (Ding and Dubchak, 2001). Since a basic classifier is defined by one operation engine and one input type, one way to use the information from the multiple input types is to combine the above 6 input types into one and use a \([(21+5)(20+2a)]D\) vector to represent it. However, doing so would introduce too many parameters into the input, thereby reducing the cluster-tolerant capacity (Chou, 1999) and cross-validation success rate. Furthermore, the PseAA with a different value of \(\lambda\) will become a different input type. In the present study, \(\lambda\) was assigned with 1, 4, 14 and 30. Therefore, we are actually facing \(5 + 4 = 9\) different input types (Table 2), and have 9 basic classifiers. To deal with this situation, we shall introduce an ensemble classifier, by which not only the other five features described in (Ding and Dubchak, 2001) but also the pseudo-amino acid compositions with a set of different \(\lambda\) values can be automatically fused into one prediction system.

The framework of ensemble classifier system was established by combining numerous basic classifiers together in order to reduce the variance caused by the peculiarities of a single training set and hence be able to learn a more expressive concept in classification than a single classifier. Illustrated in Figure 2 is the basic framework for an ensemble classifier that consists of

---

Table 2. List of nine features extracted from protein sequences for fold recognition

<table>
<thead>
<tr>
<th>Features</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-amino Acid Composition&lt;sup&gt;a&lt;/sup&gt;</td>
<td>22</td>
</tr>
<tr>
<td>Pseudo-amino Acid Composition&lt;sup&gt;b&lt;/sup&gt;</td>
<td>28</td>
</tr>
<tr>
<td>Pseudo-amino Acid Composition&lt;sup&gt;c&lt;/sup&gt;</td>
<td>48</td>
</tr>
<tr>
<td>Pseudo-amino Acid Composition&lt;sup&gt;d&lt;/sup&gt;</td>
<td>80</td>
</tr>
<tr>
<td>Predicted secondary structure</td>
<td>21</td>
</tr>
<tr>
<td>Hydrophobicity</td>
<td>21</td>
</tr>
<tr>
<td>Normalized van der Waals volume</td>
<td>21</td>
</tr>
<tr>
<td>Polarity</td>
<td>21</td>
</tr>
<tr>
<td>Polarizability</td>
<td>21</td>
</tr>
</tbody>
</table>

<sup>a</sup>The effects of the first rank of sequence-order correlation are incorporated [cf. Equation (5) with \(\lambda = 1\)].
<sup>b</sup>The effects of the first 4 ranks of sequence-order correlation are incorporated [cf. Equation (5) with \(\lambda = 4\)].
<sup>c</sup>The effects of the first 14 ranks of sequence-order correlation are incorporated [cf. Equation (5) with \(\lambda = 14\)].
<sup>d</sup>The effects of the first 30 ranks of sequence-order correlation are incorporated [cf. Equation (5) with \(\lambda = 30\)].
\[ \Omega = 9 \text{ basic classifiers. The final output of the ensemble is the weighted fusion of the outputs produced by the nine individual classifiers, as formulated below.} \]

Suppose the ensemble classifier \( C \) is expressed by

\[ C = C_1 \oplus C_2 \oplus C_3 \oplus \cdots \oplus C_8 \oplus C_9 \tag{7} \]

where \( C_1, C_2, \ldots, C_9 \) represent the nine basic OET-KNN classifiers (Appendix A) each operating on the input derived from one of the nine features listed in Table 2; i.e. classifier \( C_j \) operates on the 22D PseAA, \( C_2 \) on the 28D PseAA, \( C_3 \) on the 48D PseAA, \( C_4 \) on the 48D PseAA, \( C_5 \) on the 21D predicted secondary structure \( C_6 \) on the 21D hydrophobicity, \( C_7 \) on the 21D normalized van der Waals volume, \( C_8 \) on the 21D polarity, and \( C_9 \) on the 21D polarizability. In Equation (7) the symbol \( \oplus \) denotes the fusing operator. For reader’s convenience, the values of the nine input parameter systems (cf. Table 2) for each of the proteins in the training and testing datasets are given in the Online Supplementary Materials BI and BII, respectively.

Thus, the process of how the ensemble classifier \( C \) works by fusing the nine basic classifiers \( C(i) \) can be formulated as follows. Suppose

\[ Y_j = \sum_{i=1}^{9} w_i \mathcal{R}_i(P, S_i), \quad (j = 1, 2, \cdots, 27) \tag{8} \]

where \( S_j \) is the set only containing proteins of fold type 1, \( S_j \) the set of fold type 2, and so forth; \( \mathcal{R}_i(P, S_i) \) is the belief function or supporting degree for \( P \) belonging to \( S_j \) obtained by the \( i \)th basic classifier as defined by Equation (A5) in Appendix A; and \( w_i \) is the weighted factor, which was assigned in this study with the value of the success rate obtained by the \( i \)th single basic classifier \( C_i \), as will be further discussed below.

Thus the query protein \( P \) is predicted belonging to the fold type with which its score of Equation 8 is the highest; i.e. suppose

\[ Y_{\mu} = \text{Max}\{Y_1, Y_2, \cdots, Y_{27}\} \tag{9} \]

where the operator \( \text{Max} \) means taking the maximum one among those in the brackets, and the subscript \( \mu \) is the very fold type predicted for the query protein \( P \). If there is a tie, the query protein may not be uniquely determined and will be randomly assigned among those with a tie, but cases like that rarely occur.

RESULTS AND DISCUSSION

To demonstrate the power of the ensemble classifier, predictions were conducted based on the same training and testing datasets used by the previous investigators (Chung and Huang, 2003; Ding and Dubchak, 2001). None of proteins in these datasets has >35% sequence identity to any other, and most of proteins in the testing dataset have <25% sequence identity with those in the training dataset (Ding and Dubchak, 2001). The overall success rate in recognizing the fold among the 27 folding types by the ensemble classifier for the 383 proteins in the independent dataset is given in Table 3, where, for facilitating comparison, the success rates by the other approaches are also listed. As can be seen from Table 3, the ensemble classifier, which was formed by fusing nine basic classifiers, obviously outperformed the other approaches.

It is instructive to note that if using each of the nine basic classifiers \( C_1, C_2, \ldots, C_8, C_9 \) to do the same prediction, the success rates would be 0.40, 0.44, 0.40, 0.29, 0.42, 0.37, 0.32, 0.29, 0.24, respectively. All of them are significantly lower than the rate of 0.62 = 62% obtained by the ensemble classifier (Table 3), indicating that a strong classifier can be generated by fusing many weak classifiers. Actually, as mentioned above, these single classifier rates were assigned for the weights \( w_i \) \( (i = 1, 2, \ldots, 9) \) in Equation (9) to form the ensemble classifier.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Classifier & Success rate(\%) \\
\hline
MLP (Multi-Layer Perceptron) & 48.8 \\
(Chung and Huang, 2003) & \\
GRNN (General Regression Neural Networks) & 44.2 \\
(Chung and Huang, 2003) & \\
RBFN (Radial Basis Function Networks) & 49.4 \\
(Chung and Huang, 2003) & \\
NN (Neural Networks)* & 41.8 \\
(Ding and Dubchak, 2001) & \\
SVM (Support Vector Machines)* & 45.2 \\
(Ding and Dubchak, 2001) & \\
SVMm (Ding and Dubchak, 2001) & 51.1 \\
SVMm (Ding and Dubchak, 2001) & 56.0 \\
Ensemble Classifier* & 62.1 \\
\hline
\end{tabular}
\caption{Overall success rates by different approaches in recognizing the fold types for proteins in the independent testing dataset}
\end{table}

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Success rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP (Multi-Layer Perceptron)</td>
<td>48.8</td>
</tr>
<tr>
<td>(Chung and Huang, 2003)</td>
<td></td>
</tr>
<tr>
<td>GRNN (General Regression Neural Networks)</td>
<td>44.2</td>
</tr>
<tr>
<td>(Chung and Huang, 2003)</td>
<td></td>
</tr>
<tr>
<td>RBFN (Radial Basis Function Networks)</td>
<td>49.4</td>
</tr>
<tr>
<td>(Chung and Huang, 2003)</td>
<td></td>
</tr>
<tr>
<td>NN (Neural Networks)*</td>
<td>41.8</td>
</tr>
<tr>
<td>(Ding and Dubchak, 2001)</td>
<td></td>
</tr>
<tr>
<td>SVM (Support Vector Machines)*</td>
<td>45.2</td>
</tr>
<tr>
<td>(Ding and Dubchak, 2001)</td>
<td></td>
</tr>
<tr>
<td>SVMm (Ding and Dubchak, 2001)</td>
<td>51.1</td>
</tr>
<tr>
<td>SVMm (Ding and Dubchak, 2001)</td>
<td>56.0</td>
</tr>
<tr>
<td>Ensemble Classifier*</td>
<td>62.1</td>
</tr>
</tbody>
</table>

*The training method for NN is ‘one against others’.
*The training method for SVM is ‘one against others’.
*The training method for SVM is ‘unique one against others’.
*The training method for SVM is ‘all against all’.

CONCLUSIONS

An ensemble classifier is formed by a set of basic classifiers, whose individual outcomes are combined in some way, typically through a weighted voting, to give a final determination in classifying a query sample. The current ensemble classifier consists of nine basic individual classifiers. Their operation engine was OET-KNN algorithm, but they were each trained in nine different parameter systems extracted from the training dataset; i.e. 22D PseAA, 28D PseAA, 48D PseAA, 80D PseAA, 21D predicted secondary structure, 21D hydrophobicity, 21D normalized van der Waals volume, 21D polarity and 21D polarizability.

It is instructive to note that although the operation engine adopted here for the basic classifiers is the OET-KNN algorithm, others, such as the covariant discriminant algorithm and SVM algorithm, can also be used to replace the OET-KNN for forming different ensemble classifiers. Moreover, the constituent individual basic classifiers can be driven by completely different operation engines as well, and an ensemble classifier thus formed would become one with a mixture of operation engines. Similarly, we can also design an ensemble classifier by fusing both different input types and different operation engines. It is shown thru the present study that the ensemble classifier formed by fusing different input types, particularly different dimensions of pseudo-amino acid composition [cf. Equation (5)], is very promising for enhancing the success rate in recognizing the fold type of proteins.

APPENDIX A

The optimized evidence-theoretic k-nearest neighbors (OET-KNN) classifier

For reader’s convenience, a brief introduction of the OET-KNN classifier is given below. For further explanation, refer to (Shen and Chou, 2005). Let us consider a problem of classifying \( N \) entities into 27 classes (fold types), which can be formulated as
The available information is assumed to consist of a training dataset
\[ \mathbb{F} = \{ \Phi_1, \Phi_2, \ldots, \Phi_{m}, \ldots, \Phi_{27} \} \quad (A1) \]
The pattern (class) labels obtained for each of the masses are assigned to each subset concerned. Such masses are members of that pattern. Based on this evidence, basic belief (Cover and Hart, 1967), an unclassified entity \( \mathbb{P} \) is assigned to the class represented by a majority of its \( K \)-nearest neighbors of \( \mathbb{P} \). Owing to its good performance and simple-to-use feature, the KNN rule, also named as ‘voting KNN rule’, is quite popular in pattern recognition community.

The ET-KNN (evidence theoretic \( K \)-nearest neighbors) rule is a pattern classification method based on the Dempster–Shafer theory of belief functions (Denoeux, 1995). In the classification process, each neighbor of a pattern to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that pattern. Based on this evidence, basic belief masses are assigned to each subset concerned. Such masses are obtained for each of the \( K \)-nearest neighbors of the pattern under consideration and aggregated using the Dempster’s rule of combination (Shafer, 1976). A decision is made by assigning a pattern to the class with the maximum credibility.

Suppose \( \mathbb{P} \) is a query protein to be classified, and \( S^P_k \) is the set of its \( k \)-nearest neighbors in the training dataset \( \mathbb{N} \) of Equation (A2). Thus, for any \( \mathbb{P}_i \in S^P_k \), the knowledge that \( \mathbb{P}_i \) belongs to class \( \Phi_m \in \mathbb{F} \) can be considered as a piece of evidence that increases our belief that \( \mathbb{P} \) also belongs to \( \Phi_m \). According to the basic belief assignment mapping theory (Shafer, 1976), this item of evidence can be formulated by
\[ \mathbb{R}(\mathbb{P}_i, \Phi_m) = \alpha_0 \exp \left[ - \gamma^2_m D^2(\mathbb{P}_i, \mathbb{P}) \right] \quad (A3) \]
where \( \alpha_0 \) is a fixed parameter, \( \gamma_m \) is a parameter associated with class \( \Phi_m \), and \( D^2(\mathbb{P}_i, \mathbb{P}) \) is the square Euclidean distance between \( \mathbb{P} \) and \( \mathbb{P}_i \). In the ET-KNN rule, it was not addressed how to optimally select the parameters. In 1998 an optimization procedure to determine the optimal or near-optimal parameter values was proposed from the data by minimizing an error function (Zouhal and Denoeux, 1998). It was observed that the OET-KNN rule obtained thus an optimization treatment would lead to a substantial improvement in classification accuracy. The optimal parameter thus obtained for \( \alpha_0 \) of Equation A3 is 0.95, and those for \( \gamma_m \) are given in Table A1.

### Table A1. The optimal parameter \( \gamma_j (j = 1, 2, \ldots, 27) \) in Equation A3 obtained thru the optimized procedure (Zouhal & Denoeux, 1998) for the 9 basic individual classifiers in Equation 7

<table>
<thead>
<tr>
<th>( \gamma_1 )</th>
<th>( \gamma_2 )</th>
<th>( \gamma_3 )</th>
<th>( \gamma_4 )</th>
<th>( \gamma_5 )</th>
<th>( \gamma_6 )</th>
<th>( \gamma_7 )</th>
<th>( \gamma_8 )</th>
<th>( \gamma_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1028</td>
<td>0.2908</td>
<td>0.0656</td>
<td>0.0888</td>
<td>0.0798</td>
<td>0.0931</td>
<td>0.0954</td>
<td>0.1241</td>
<td>0.1476</td>
</tr>
<tr>
<td>0.0714</td>
<td>0.0727</td>
<td>0.0490</td>
<td>0.0480</td>
<td>0.0525</td>
<td>0.1176</td>
<td>0.0783</td>
<td>0.1013</td>
<td>0.1076</td>
</tr>
<tr>
<td>0.0398</td>
<td>0.0523</td>
<td>0.0240</td>
<td>0.0597</td>
<td>0.0274</td>
<td>0.0468</td>
<td>0.0520</td>
<td>0.0475</td>
<td>0.0700</td>
</tr>
<tr>
<td>0.0434</td>
<td>0.1324</td>
<td>0.0435</td>
<td>0.4306</td>
<td>0.0312</td>
<td>0.1298</td>
<td>0.0543</td>
<td>0.0462</td>
<td>0.0699</td>
</tr>
<tr>
<td>0.4848</td>
<td>0.0585</td>
<td>0.0469</td>
<td>0.1541</td>
<td>0.4146</td>
<td>0.2871</td>
<td>0.1890</td>
<td>0.1386</td>
<td>0.2142</td>
</tr>
<tr>
<td>0.1275</td>
<td>0.0784</td>
<td>0.0604</td>
<td>0.4092</td>
<td>0.1034</td>
<td>0.0973</td>
<td>0.3587</td>
<td>0.3606</td>
<td>0.5427</td>
</tr>
<tr>
<td>0.1977</td>
<td>0.3654</td>
<td>0.2482</td>
<td>0.4026</td>
<td>0.1011</td>
<td>0.3723</td>
<td>0.5057</td>
<td>0.1101</td>
<td>0.1234</td>
</tr>
<tr>
<td>0.1396</td>
<td>0.2218</td>
<td>0.2988</td>
<td>0.4026</td>
<td>0.0425</td>
<td>0.1200</td>
<td>0.1365</td>
<td>0.0795</td>
<td>0.1064</td>
</tr>
<tr>
<td>0.0682</td>
<td>0.0387</td>
<td>0.0445</td>
<td>0.0436</td>
<td>0.0731</td>
<td>0.0465</td>
<td>0.0645</td>
<td>0.0452</td>
<td>0.0645</td>
</tr>
<tr>
<td>0.1210</td>
<td>0.1002</td>
<td>0.1219</td>
<td>0.1331</td>
<td>0.1033</td>
<td>0.1108</td>
<td>0.1543</td>
<td>0.1665</td>
<td>0.6170</td>
</tr>
<tr>
<td>0.0787</td>
<td>0.0518</td>
<td>0.1014</td>
<td>0.0969</td>
<td>0.0899</td>
<td>0.0875</td>
<td>0.1146</td>
<td>0.1026</td>
<td>0.1548</td>
</tr>
<tr>
<td>0.0436</td>
<td>0.0265</td>
<td>0.0455</td>
<td>0.0449</td>
<td>0.0479</td>
<td>0.0523</td>
<td>0.0687</td>
<td>0.0706</td>
<td>0.1543</td>
</tr>
<tr>
<td>0.0448</td>
<td>0.0840</td>
<td>0.0594</td>
<td>0.0375</td>
<td>0.0484</td>
<td>0.0317</td>
<td>0.0578</td>
<td>0.1815</td>
<td>0.0502</td>
</tr>
<tr>
<td>0.4871</td>
<td>0.2862</td>
<td>0.3810</td>
<td>0.2096</td>
<td>0.0702</td>
<td>0.0961</td>
<td>0.1376</td>
<td>0.4662</td>
<td>0.0977</td>
</tr>
<tr>
<td>0.2142</td>
<td>0.2423</td>
<td>0.1911</td>
<td>0.1164</td>
<td>0.2026</td>
<td>0.1280</td>
<td>0.1298</td>
<td>0.1505</td>
<td>0.3061</td>
</tr>
<tr>
<td>0.5427</td>
<td>0.2530</td>
<td>0.3326</td>
<td>0.3723</td>
<td>0.0665</td>
<td>0.4249</td>
<td>0.1346</td>
<td>0.5627</td>
<td>0.1003</td>
</tr>
<tr>
<td>0.1234</td>
<td>0.3039</td>
<td>0.2352</td>
<td>0.3723</td>
<td>0.4069</td>
<td>0.1442</td>
<td>0.1472</td>
<td>0.1430</td>
<td>0.1334</td>
</tr>
<tr>
<td>0.0795</td>
<td>0.0436</td>
<td>0.0731</td>
<td>0.0645</td>
<td>0.1816</td>
<td>0.0407</td>
<td>0.0505</td>
<td>0.0419</td>
<td>0.0511</td>
</tr>
<tr>
<td>0.0419</td>
<td>0.0496</td>
<td>0.0409</td>
<td>0.0998</td>
<td>0.0662</td>
<td>0.0566</td>
<td>0.0178</td>
<td>0.0466</td>
<td>0.0419</td>
</tr>
<tr>
<td>0.1416</td>
<td>0.1119</td>
<td>0.4207</td>
<td>0.2484</td>
<td>0.1581</td>
<td>0.1430</td>
<td>0.1379</td>
<td>0.4289</td>
<td>0.0093</td>
</tr>
<tr>
<td>0.5118</td>
<td>0.4803</td>
<td>0.2018</td>
<td>0.1513</td>
<td>0.1527</td>
<td>0.1243</td>
<td>0.3159</td>
<td>0.3329</td>
<td>0.0403</td>
</tr>
<tr>
<td>0.1443</td>
<td>0.1240</td>
<td>0.2975</td>
<td>0.1517</td>
<td>0.1527</td>
<td>0.1379</td>
<td>0.1897</td>
<td>0.3937</td>
<td>0.2920</td>
</tr>
<tr>
<td>0.0419</td>
<td>0.0496</td>
<td>0.0409</td>
<td>0.0998</td>
<td>0.0662</td>
<td>0.0566</td>
<td>0.0178</td>
<td>0.0466</td>
<td>0.0419</td>
</tr>
<tr>
<td>0.0403</td>
<td>0.0425</td>
<td>0.0403</td>
<td>0.0492</td>
<td>0.0492</td>
<td>0.0403</td>
<td>0.0419</td>
<td>0.0466</td>
<td>0.0419</td>
</tr>
</tbody>
</table>
The belief function of \( P \) belonging to class \( \Phi_\mu \) is a combination of its \( k \)-nearest neighbors, and can be formulated as

\[
\mathbb{R}(P, \Phi_\mu) = (\cdots \mathbb{R}(P, \Phi_\mu) \oplus \mathbb{R}(P, \Phi_\mu) \oplus \cdots) \oplus \mathbb{R}(P, \Phi_\mu)
\]

where \( \oplus \) is called the orthogonal sum, which is commutative and associative. According to Dempster’s rule (Shafer, 1976), the belief function of Equation A4 can be expressed as

\[
\mathbb{R}(P, \Phi_\mu) = \frac{\sum_{i \in S_{K_i}} \mathbb{R}(P, \Phi_{i}) \mathbb{R}(P, \Phi_{S_{K_i}}) \mathbb{R}(P, \Phi_{S_{K_i}})}{1 - \sum_{i \in S_{K_i}} \mathbb{R}(P, \Phi_{i}) \mathbb{R}(P, \Phi_{S_{K_i}}) \mathbb{R}(P, \Phi_{S_{K_i}})}
\]

where \( S_{K_i} \) is the \( i \)-th possible subset of \( S_{K_i} \) and \( \cup, \cap \) and \( \emptyset \) are the symbols in set theory, representing 'contained in', 'intersection', and the empty set, respectively.

A decision is made by assigning the query protein \( P \) to the class with which the belief or credibility function of Equation A5 has the maximum value; i.e. if

\[
\mathbb{R}(P, \Phi_\mu) = \text{Max} \{ \mathbb{R}(P, \Phi_1), \mathbb{R}(P, \Phi_2), \cdots, \mathbb{R}(P, \Phi_27) \}
\]

where \( \mu = 1, 2, \ldots, 27 \) and the operator \( \text{Max} \) means taking the maximum one among those in the brackets, then the class \( \Phi_\mu \) is the class predicted for the query protein.

Conflict of Interest: none declared.

REFERENCES


