Improved DNA-binding protein identification by incorporating evolutionary information into the Chou’s PseAAC

XIANGZHENG FU1, WEN ZHU1,2, BO LIAO1,2, LIJUN CAI1, LIHONG PENG3 AND JIALIANG YANG2,4

1College of Information Science and Engineering, Hunan University, Changsha, Hunan, 410082, P.R. China
2School of Mathematics and Statistics, Hainan Normal University, Haikou, 570100, P.R. China
3School of Computer Science, Hunan University of Technology, Zhuzhou, Hunan, 412007, P.R. China
4Icahn Institute for Genomics and Multiscale Biology, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA

Corresponding author: LIJUN CAI (ljcai@hnu.edu.cn); BO LIAO (dragonbw@163.com); JIALIANG YANG (jialiang.yang@mssm.edu).

This study is supported by the Program for New Century Excellent Talents in university (Grant No.NCET-10-0365), National Nature Science Foundation of China (Grant Nos. 11171369, 61272395, 61300128, 61472127, 61572178, 61672214 and 61772192), and the Natural Science Foundation of Hunan, China (Grant Nos. 2018JJ2461, 2018JJ3570).

ABSTRACT DNA-binding proteins play critical roles in various cellular biological processes, such as gene expression and transcription. However, experimental methods to identify these proteins like ChIP-sequencing are expensive and time-consuming, which presents the need for in silico methods, especially machine learning-based methods. In recent years, the accuracy of machine learning-based DNA-binding protein prediction has been increasing significantly. However, there are still some critical problems to be solved like how to convert protein sequences into an appropriate discrete model or vector. In this study, we propose a novel feature construction method based on a position-specific scoring matrix (PSSM) named K-PSSM-Composition. The proposed features can efficiently capture the information about 20 amino acid residues and the local information of a given sequence during the evolutionary process. We perform a recursive feature elimination to extract the optimal set of features, which are used to train a support vector machine (SVM) model for predicting DNA-binding proteins. We evaluate and compare our proposed predictor with other advanced predictors via two standard benchmark datasets. The proposed method achieves the accuracy values of 89.77% and 88.71% for the jackknife test and independent test respectively, outperforming the compared methods. This finding demonstrates the efficacy and effectiveness of the proposed method in predicting DNA-binding proteins. The source code and data are available at https://github.com/Excelsior511/DNA-Binding-Proteins.

INDEX TERMS DNA-binding protein identification, Feature representation algorithm, Evolutionary information, Support vector machine.

I. INTRODUCTION

DNA-binding proteins play an important role in various biomolecular activities, including DNA damage and transcriptional regulation [1, 2]. Given the importance of these proteins, various methods have been developed to identify them. In previous studies, DNA-binding proteins are mainly detected through biological experimental methods, including filter binding analysis, gene analysis, chromatin immunoprecipitation on microarrays, and X-ray crystallography [3, 4], which are time-consuming and costly [1]. With the development and application of next-generation high-throughput DNA sequencing technology [5], the amount of available protein sequence data has been increasing rapidly. Since 1986, the Swiss-Prot [6] database has expanded its protein sequence library by more than 100 times. To handle this large-scale protein sequence data, fast and efficient computational methods are demanded, especially those based on machine learning (ML) algorithms and statistical models [7-9]. The prediction performance of ML-based methods mainly depends on the extraction of their feature information. Generally speaking, popular feature extraction methods used in ML-based predictors are mainly composed of structural information-based methods [1, 10-19] and sequence-based methods [18, 20-40].
Structure-based prediction methods rely on protein structural information. A three-dimensional (3-D) structure is the natural state of a protein, which consists of a secondary structure, a 3-D structure, and a side chain structure. There are a few methods taking the advantage of protein 3-D structural information to predict its ability to bind to DNAs. For example, Gao and Skolnick [11] combined protein structural comparisons and statistical analyses to determine a possible interaction between DNA base pairs and residues. They also [15] proposed a template based on the structure of DNA-binding protein complexes. Other methods integrate both protein structure and sequence information. For example, Szilágyi and Skolnick [12] implemented a logistic regression classifier whose eigenvectors include global amino acid composition, amino acid spatial distribution, and electric dipole moments. Although these structure-based prediction methods have good accuracy in predicting DNA-binding proteins, most of the proteins, especially a large number of newly discovered protein sequences in the postgenomic era, lack of structural information. Thus, such methods are not applicable.

Sequence-based prediction methods only require protein sequence information to predict DNA-binding proteins. For example, Fang et al. [30] used ACC transformation and dipeptide component combination to establish eigenvectors, based on which to predict protein binding affinity to DNAs. Langlois and Lu [1] proposed a new protein sequence representation method called LEAC that exploits protein secondary structural features and sequence characteristics. Zou et al. [3] utilized three different sequence feature transformation methods to construct eigenvectors. Other researchers also exploited the physicochemical properties of proteins to enhance predictions. For example, Cai et al. [41] studied three protein structural factors, namely helix-turn-helix, helix-hairpin-helix, and helix-loop-helix. Kumar et al. [20] extracted protein characteristics in terms of their physicochemical properties, such as hydrophobicity (H), polarity (P1), polarizability (P2), and volumes of side chains of amino acids (VSC). Lin et al. [42] established a gray model to express protein sequences as pseudo-amino acid composition (PseAAC). In some sequence-based prediction methods, such as position-specific score matrix (PSSM), evolutionary information is extracted from the frequency spectrum output by PSI-BLAST [43]. Kumar et al. [44] proposed DNAbiner to extract features from the PSSM matrix and used support vector machines to make predictions. To obtain better predictive results, some researchers utilized methods that combine frequency spectra with other sequence features. For example, Dey et al. [45] put forward four types of features, namely evolutionary conservative residues, spatial clustering, hydrogen bond donor function, and residue propensity. Lou et al. [4] used primary sequence, secondary structure, predictable relative solvent accessibility, and PSSM matrix feature extraction. They also utilized decision trees, logistic regression, and other tools as classifiers. Liu et al. [22] presented a predictor called iDNAPro-PseAAC that contains evolutionary information and PseAAC. Waris et al. [25] trained multiple classifiers by using features extracted from dipeptide components, split amino acid composition, and PSSM and found a classifier that obtains the best predictive performance.

With the explosive growth of biological sequences in the post-genomic era, one of the most important but also most difficult problems in computational biology is how to express a biological sequence with a discrete model or a vector, yet still keep considerable sequence-order information or key pattern characteristic. This is because all the existing machine-learning algorithms can only handle vector but not sequence samples, as elucidated in a comprehensive review [46]. However, a vector defined in a discrete model may completely lose all the sequence-pattern information. To avoid completely losing the sequence-pattern information for proteins, the pseudo amino acid composition [47] or PseAAC [48] was proposed. Ever since then, it has been widely used in nearly all the areas of computational proteomics (see, e.g., [49-51] as well as a long list of references cited in [52]). Because it has been widely and increasingly used, recently three powerful open access soft-ware, called ‘PseAAC-Builder’, ‘propy’, and ‘PseAAC-General’, were established: the former two are for generating various modes of Chou’s special PseAAC [53]; while the 3rd one for those of Chou’s general PseAAC [54], including not only all the special modes of feature vectors for proteins but also the higher level feature vectors such as “Functional Domain” mode, “Gene Ontology” mode, and “Sequential Evolution” or “PSSM” mode. Encouraged by the successes of using PseAAC to deal with protein/peptide sequences, the idea of PseAAC was extended to PseKNC (Pseudo K-tuple Nucleotide Composition) to generate various feature vectors for DNA/RNA sequences that have proved very successful as well [55]. Particularly, recently a very powerful web-server called ‘Pse-in-One’ [56] and its updated version ‘Pse-in-One2.0’ [57] have been established that can be used to generate any desired feature vectors for protein/peptide and DNA/RNA sequences according to the users’ need or their own definition. In summary, feature-based methods to predict DNA-bind proteins are promising. However, it is challenging to retrieve effective feature representations for all query protein sequences.

As a series of recent publications[58-71] consistent with Chou’s 5-step rules, many researchers follow five guiding principles in order to create a truly useful sequence-based biosystem statistical predictor: (a) construct or select a valid benchmark dataset to train and test the predictor; (b) formulate the biological sequence samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target to be predicted; (c) introduce or develop a powerful algorithm (or engine) to operate the prediction; (d) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor;
(e) establish a user-friendly web-server for the predictor that is accessible to the public. In this study, we propose K-PSSM-composition, a promising feature representation algorithm that efficiently extracts features from profiles (PSSM). The proposed features can efficiently capture information about the 20 amino acid residues and local information regarding a particular sequence during an evolutionary process. In the framework of the proposed algorithm, the protein sequence information is obtained by dividing the original PSSM matrix into K sub-PSSM matrices of the same size, and each PSSM submatrix is normalized. The sum of the row vectors of the same amino acid residue characters of each normalized submatrix is then calculated to construct PSSM submatrix features. Finally, the features of each submatrix are combined to construct the K-PSSM-composition features, and the proposed features are fed into the training SVM model to predict DNA-binding proteins. Our method is subsequently evaluated and compared with popular methods on two stringent benchmark datasets: PDB1075 [42] used in a jackknife test and PDB186 [4] utilized as an independent test.

II. Materials and methods

A. Framework of the proposed method

Fig. 1 illustrates the overall framework of our method, which consists of two phases including model training and DNA-binding protein prediction.

![FIGURE 1. Overall framework of the proposed method for predicting DNA-binding proteins.](image)

In the model training phase, training samples are first encoded into K-PSSM-Composition feature vectors by the proposed k-PSSM-Composition feature extraction algorithm. The support vector machine recursive feature elimination (SVM-RFE) algorithm and the Correlation Bias Reduction (CBR) [72] algorithm are then used to extract an optimal set of features, which are fed into a support vector machine (SVM) classifier to generate a training model. In the prediction phase, the same feature representation process is applied to testing protein sequences and the trained SVM classifier is employed to predict their DNA binding affinities.

B. Datasets

In order to establish an effective and accurate predictor, a strict and reliable data set is needed. The DNA binding protein data is usually selected from the Protein Data Bank (http://www.rcsb.org/pdb/home/home.do). The user can obtain the amino acid sequence information of the DNA binding protein by querying the keyword in the PDB database. In order to construct high-quality non-redundant data, we screened the DNA binding protein sequences obtained from the PDB database by the following two criteria: (1) we removed sequences which are short (less than 50 amino acids) or contain the character "X"; (2) we eliminated redundancy and homogeneity bias that may lead to overestimation of performance. The PISCES tool was used to remove fragments with a similarity below 25%. The benchmark data set was constructed using the protein sequences passing the two filtering steps.

In this study, the accuracy of the prediction was major tested on two benchmark datasets. The first benchmark dataset was PDB1075 [42] consisting of 525 DNA binding proteins (positive samples) and 550 non-DNA binding proteins (negative samples) selected from the PDB (released December 2013); the other benchmark dataset, called PDB186, was constructed by Lou et al. [4] containing 93 DNA binding proteins and 93 non-DNA binding proteins collected from PDB. The PDB186 data set provides independent testing for verifying predictors.

Recently, Mishra et al. [40] selected 518 DNA binding proteins and 545 non-DNA binding proteins from the PDB1075 to construct a benchmark dataset to predict whether it is a DNA binding protein. This study names the dataset constructed by Mishra et al. [40] as PDB1063.

C. Classification algorithm

We adopted SVM as classifiers, a widely used machine learning algorithm in classification. We used publicly available support vector machine library (LIBSVM) to implement our SVM classifier.

The LIBSVM toolkit can be downloaded freely at http://www.csie.ntu.edu.tw/~cjlin/libsvm. We integrated this toolbox in the Matrix Laboratory (MATLAB) workspace to build the prediction system. Here, a radial basis function was selected as a kernel function, and a grid search based on 10-fold cross validation was used to optimize the SVM parameter γ and the penalty parameter C.

D. Feature extraction
Evolutionary information is essential for protein function annotation in biological analysis and has been widely used in many studies. In this study, evolutionary information is retrieved in the form of the PSSM matrix of each protein sequence, which is generated by running the PSI-BLAST [43] program to search the nrdb50 [73] database through three iterations with 0.001 as an E-value cutoff of multiple sequence alignment. This study proposes a new protein sequence feature representation algorithm called K-PSSM-composition based on the PSSM matrix. In the following subsections, we briefly introduce PSSM and describe the procedure involving K-PSSM-composition.

1) POSITON-SPECIFIC SCORING MATRIX (PSSM).

PSSM stores PSI-BLAST-generated evolutionary information of protein sequences [43]. A given protein sequence S is expressed as $S_1S_2...S_L$, where $S_i$ (1 ≤ i ≤ L) is the amino acid (residue) that appears at the ith position of the sequence S, and L is the length of S. The PSSM size of the sequence S is L × 20 (L rows and 20 columns) formulated as follows:

$$PSSM_{original} = \begin{bmatrix}
    p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\
    p_{2,1} & p_{2,2} & \cdots & p_{2,20} \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{L,1} & p_{L,2} & \cdots & p_{L,20}
\end{bmatrix}_{L \times 20}$$ (1)

Where L is the length of the primary protein sequence, and $p_{i,j}$ (i = 1,2,...,L; j = 1,2,...,20) is the score that the amino acid residue at the ith location of the protein sequence changes to amino acid j during the evolutionary process.

2) K-PSSM-COMPOSITION FEATURES.

The K-PSSM-composition method is based on the PSSM-composition feature extraction method. PSSM-composition was proposed by Ronesh Sharma et al. [74], which was targeted at subcellular localization prediction. PSSM-composition features can sufficiently explore evolutionary information. However, the information between 20 amino acid residues and the local information of the sequence are lost during the evolutionary process if sequence features are extracted through the PSSM-composition method. To address these problems, we modified the PSSM-Composition features through the following steps.

Step 1. Segment the PSSM matrix (Fig. 2).

The PSSM matrix is divided into k equal submatrix (sub-PSSMs) by row; the obtained sub-PSSM matrix is expressed in Equation 2:

$$SubPSSM(\lambda) = \begin{bmatrix}
    p'_{d+1,1} & p'_{d+1,2} & \cdots & p'_{d+1,20} \\
    p'_{d+2,1} & p'_{d+2,2} & \cdots & p'_{d+2,20} \\
    \vdots & \vdots & \ddots & \vdots \\
    p'_{d+\lambda,1} & p'_{d+\lambda,2} & \cdots & p'_{d+\lambda,20}
\end{bmatrix}_{\lambda \times 20}$$ (2)

Where d = (\lambda - 1)\star(U(\lambda)), and U(\lambda) represents the number of rows of each submatrix for $\lambda = 1, 2, \ldots, k$ with

$$U(\lambda) = \begin{bmatrix}
    \left\lfloor \frac{L}{\lambda} \right\rfloor, & \lambda = 1, 2, \ldots, k-1 \\
    L - \left\lfloor \frac{L}{\lambda} \right\rfloor, & \lambda = k
\end{bmatrix}$$ (3)

Here, \([ \star ]\) denotes rounding down. The number of rows of the first k-1 matrix is $\left\lfloor \frac{L}{\lambda} \right\rfloor$, and the number rows of the kth submatrix is $L - \left\lfloor \frac{L}{\lambda} \right\rfloor$.

Step 2. Normalize sub-PSSM matrices.

Each sub-PSSM matrix (Equation 2) is normalized as follows:

$$SubPSSM(\lambda)_{normalized} = \begin{bmatrix}
    f_{d+1,1} & f_{d+1,2} & \cdots & f_{d+1,20} \\
    f_{d+2,1} & f_{d+2,2} & \cdots & f_{d+2,20} \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{d+i,1} & f_{d+i,2} & \cdots & f_{d+i,20} \\
    f_{d+U(\lambda),1} & f_{d+U(\lambda),2} & \cdots & f_{d+U(\lambda),20}
\end{bmatrix}_{U(\lambda) \times 20}$$ (4)

where
\[ f_{i,j} = \frac{1}{1 + e^{-p_{i,j}}} \] (5)
and \( p_{i,j} \) (see Eq. 2).

**Step 3. Calculate each of the sub-PSSM matrix features and combine them to obtain k-PSSM-composition features.**

For each normalized sub-PSSM matrix, the sum of the row vectors of the same amino acid residue character is calculated. For example, the sum of the rows of the amino acid residue “K” in the sub-PSSM matrix is identified to obtain a new row (20-D vector; Table 1), and other amino acid residues are subjected to the same procedure. These new lines are then merged to have a 400-D vector. The amino acids {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y} are subjected to the same procedure. These new lines are then merged to have a 400-D vector.

Finally, the features of all the sub-PSSM matrices are combined to obtain the K-PSSM-composition features and formulated as follows:

\[
K\_PSSM\_composition = \left[ PSSM\_com(1), \ldots, PSSM\_com(\lambda) \right]_{1\times(400* k)}
\] (6)

where \( \lambda = 1, \ldots, K \); \( PSSM\_com(\lambda) \) is the sub-PSSM matrix feature as follows:

\[
PSSM\_com(\lambda) = \left[ F^A, F^R, \ldots, F^\phi \right]_{1\times400}
\] (7)

Where \( \phi \) is the corresponding residue type of the 20 amino acids \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}, \( F^\phi \) is the sum of the rows of all \( \phi \) in the sub-PSSM matrix.

Here, the parameter \( k = 8 \) is selected as the default parameter. Parameter optimization is described in detail in Subsection 3.3. The protein sequence is finally represented as a 3200D feature vector.

**E. Feature selection**

After the features are extracted, all of the samples in the benchmark datasets are converted into numerical feature vectors with the same dimensions. The feature space of each protein sequence consists of K-PSSM-composition features. To remove the noisy and redundant features in the original feature space (K-PSSM-composition), reduce overfitting, and improve performance, we used the SVM-RFE and CBR [72] algorithms to select an optimal feature subset. The SVM-RFE algorithm, proposed by Guyon et al. [75], has been successfully applied in many systems biology problems [76-78]. The CBR algorithm has been utilized to reduce potential bias in linear and nonlinear SVM-RFE. In this study, feature selection by the SVM-RFE + CBR[72] algorithm is performed as follows. First, all of the feature vectors are ranked using SVM-RFE + CBR, and a group of top-ranked features is selected. Second, the selected feature vectors are reorganized into new and ordered feature vectors. Finally, these new feature vectors are fed into an SVM classifier to generate a training model.

**F. Measurements**

In this paper, in order to evaluate the prediction performance of K-PSSM-composition, two cross-validation methods namely jackknife and independent dataset tests were applied for the examination of predictor quality and its effectiveness. Receiver operating characteristic (ROC) was plotted based on specificity (Sp) and sensitivity (Sn), and areas under ROC curves (AUC) were calculated based on the trapezoidal approximation. The AUC provides a measure of the classifier performance with a high value indicating good classification performance. In addition, compared to other methods, independent tests were performed. Matthew correlation coefficient (MCC) and

<table>
<thead>
<tr>
<th>( A )</th>
<th>( R )</th>
<th>( N )</th>
<th>( D )</th>
<th>( C )</th>
<th>( Q )</th>
<th>( T )</th>
<th>( W )</th>
<th>( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>0.12</td>
<td>0.99</td>
<td>0.27</td>
<td>0.12</td>
<td>0.02</td>
<td>0.88</td>
<td>\ldots</td>
<td>0.12</td>
</tr>
<tr>
<td>K</td>
<td>0.5</td>
<td>0.27</td>
<td>0.27</td>
<td>0.05</td>
<td>0.5</td>
<td>\ldots</td>
<td>0.99</td>
<td>0.02</td>
</tr>
<tr>
<td>E</td>
<td>0.5</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.05</td>
<td>0.73</td>
<td>\ldots</td>
<td>0.95</td>
</tr>
<tr>
<td>K</td>
<td>0.5</td>
<td>0.12</td>
<td>0.05</td>
<td>0.12</td>
<td>0.05</td>
<td>0.05</td>
<td>\ldots</td>
<td>0.27</td>
</tr>
<tr>
<td>S</td>
<td>0.88</td>
<td>0.27</td>
<td>0.27</td>
<td>0.27</td>
<td>0.05</td>
<td>0.5</td>
<td>\ldots</td>
<td>0.5</td>
</tr>
<tr>
<td>P</td>
<td>0.5</td>
<td>0.05</td>
<td>0.05</td>
<td>0.02</td>
<td>0.02</td>
<td>0.73</td>
<td>\ldots</td>
<td>0.27</td>
</tr>
<tr>
<td>K</td>
<td>0.12</td>
<td>0.01</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
<td>0.12</td>
<td>\ldots</td>
<td>0.27</td>
</tr>
<tr>
<td>G</td>
<td>0.95</td>
<td>0.98</td>
<td>0.27</td>
<td>0.12</td>
<td>0.02</td>
<td>0.73</td>
<td>\ldots</td>
<td>0.12</td>
</tr>
<tr>
<td>K</td>
<td>0.5</td>
<td>0.27</td>
<td>0.27</td>
<td>0.12</td>
<td>0.05</td>
<td>0.88</td>
<td>\ldots</td>
<td>0.73</td>
</tr>
<tr>
<td>I</td>
<td>0.98</td>
<td>0.05</td>
<td>0.12</td>
<td>0.12</td>
<td>0.27</td>
<td>0.12</td>
<td>\ldots</td>
<td>0.5</td>
</tr>
<tr>
<td>S</td>
<td>0.73</td>
<td>0.73</td>
<td>0.95</td>
<td>0.5</td>
<td>0.12</td>
<td>0.5</td>
<td>\ldots</td>
<td>0.88</td>
</tr>
<tr>
<td>P</td>
<td>0.88</td>
<td>0.02</td>
<td>0.02</td>
<td>0.05</td>
<td>0.05</td>
<td>0.02</td>
<td>\ldots</td>
<td>0.5</td>
</tr>
<tr>
<td>Q</td>
<td>0.88</td>
<td>0.27</td>
<td>0.88</td>
<td>0.95</td>
<td>0.02</td>
<td>0.5</td>
<td>\ldots</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Summed up

| K | 1.74 | 1.66 | 0.91 | 0.68 | 0.17 | 2.43 | \ldots | 2.38 | 0.11 | 0.72 | 2.1 |
AUC were used to evaluate the prediction performance. The MCC accounts for true and false positives and negatives and are usually regarded as a balanced measure that can be used even if the classes are of different sizes. The sensitivity (SE), specificity (SP), accuracy (ACC) and MCC were defined as follows:

\[
\begin{align*}
SE &= \frac{TP}{TP + FN} \\
SP &= \frac{TN}{TN + FP} \\
ACC &= \frac{TP + TN}{TP + FN + TN + FP} \\
MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}
\end{align*}
\] (8)

Where TP, TN, FP and FN denote the number of true positives, true negatives, false positives and false negatives, respectively.

However, in the form of equation (8), these metrics are not intuitive and are not easy to understand for most biologists. In particular, although MCC is very important in measuring the stability of prediction methods, its interpretation of this form is not intuitive. Therefore, we use the following formula proposed by Chen et al. [79], which based on the Chou’s symbols [80]. Let \( N^+ \) (\( N^- \)) be the total number of positive (negative) samples in the data set. Let \( N^*_+ \) (\( N^*_- \)) be the number of positive (negative) samples that were incorrectly predicted. The relationship between the symbols used in equation (8) and Chou’s can be given by the following equation:

\[
\begin{align*}
TP &= N^*_- - N^*_+ \\
TN &= N^*_- - N^*_+ \\
FP &= N^*_+ \\
FN &= N^*_-
\end{align*}
\] (9)

Therefore, the performance metrics in Equation 8 can be redefined as:

\[
\begin{align*}
\text{Accuracy} &= 1 - \frac{N^*_- - N^*_+}{N^+ + N^-} \\
\text{Sensitivity} &= 1 - \frac{N^*_+}{N^+} \\
\text{Specificity} &= 1 - \frac{N^*_-}{N^-} \\
MCC &= \frac{1 - \left( \frac{N^*_- - N^*_+}{N^+} \right) \left( \frac{1}{1 + \frac{N^*_+}{N^-}} \right)}{\sqrt{1 + \frac{N^*_- - N^*_+}{N^+} \left( \frac{1}{1 + \frac{N^*_+}{N^-}} \right) + \frac{1}{1 + \frac{N^*_+}{N^-}}}}
\end{align*}
\] (10)

The interpretation of each performance metrics is more intuitive and easier to understand, as defined in equation (10). For example, when all instances of the positive (negative) class are correctly predicted, we have \( N^- = 0 \) (\( N^+ = 0 \)), so the sensitivity (specificity) of the classifier is 1. Conversely, if all of the positive (negative) instances are incorrectly predicted, then \( N^+ = N^- \) (\( N^*_+ = N^*_- \)). Therefore, the sensitivity (specificity) becomes 0. For a perfect classifier, we have \( N^+ = N^- \), in which case both the precision and the MCC become 1. On the other hand, when all samples are misclassified (i.e., \( N^- = N^*_- \) and \( N^+ = N^*_+ \)), the precision and MCC become 0 and -1, respectively. For a random predictor, we can expect \( N^*_+ = N^*_- = \frac{N^+}{2} \), which results in an accuracy of 0.5 and an MCC of 0.

A series of recently published studies have concurred and applauded the advantages of Chou’s intuitive metrics (see, e.g., [55, 60, 62, 65, 79, 81-84]). However, it is instructive to point out that the metrics described above are only valid for single label systems where each protein belongs to only one functional category. And for multi-label systems where proteins may belong to several functional categories, whose existence has become more frequent in system biology [61, 85-89], system medicine [90, 91] and biomedicine [92], a completely different set of metrics as defined in [93] is absolutely needed.

III. Results and Discussions

In this study, two cross-validation methods, namely, jackknife and independent dataset tests, were used to examine the quality and effectiveness of a predictor. In the jackknife test, our method was applied to the PDB1075 dataset to analyze the effectiveness of parameter optimization and feature selection, and the performance of our method was compared with other methods. In the independent test, our prediction model was tested on the independent PDB186 dataset and compared with the results of other methods.

A. Comparisons with state-of-the-art predictors

In this section, we compare the performance of our method with several state-of-the-art methods on the benchmark and independent test dataset. We compared the ACC, SE, SP, and MCC achieved by our predictor with the following methods: Local-DPP [36], iDNA-Protidis [42], iDNAPro-PseAAC [22], PSSM-DT [21], DNA-Prot [20], DNA Binder [44], PseDNA-Pro [37], iDNA-Prot [42], Kmer1+ACC [23], HMMBinder [35], StackDPPred [40], DPP-PseAAC [18], and Wang’s method [29]. The performance of different methods via the jackknife and independent test were displayed in Table 2 and Table 3, respectively. Here, we used the same model trained with the proposed method on the benchmark dataset and tested with the independent dataset. For a fair comparison, the performances of these methods were taken from other studies with best tuned parameters [29, 36].

VOLUME XX, 2018
Finally, it is of note that the results of applying feature selection are better than those without it.

Table 4 compares our method with other state-of-the-art predictors on the benchmark dataset PDB1063 using the Jackknife test. The respective ACC and MCC of our method (at k = 12) are 2.04% and 0.04 higher than those of the previously known best-performing predictor StackDPPred [40] (ACC = 89.96% and MCC = 0.80). In summary, the proposed method outperforms existing methods in terms of DNA-binding protein prediction.

### B. Effect of Feature Selection

We evaluated the effect of feature selection algorithm in this section. Specifically, a 5-fold cross validation was applied to the PDB1075 dataset to find the optimal set of features by utilizing SVM-RFE + CBR technique.
Table 4. Results of the proposed method and state-of-the-art predictors on the benchmark dataset PDB1063 (Jackknife test evaluation).
The best values are in boldface.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC(%)</th>
<th>SE(%)</th>
<th>SP(%)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSM-DT</td>
<td>79.96%</td>
<td>81.91%</td>
<td>78.00%</td>
<td>0.622</td>
</tr>
<tr>
<td>iDNA-Prot</td>
<td>75.40%</td>
<td>83.81%</td>
<td>64.73%</td>
<td>0.500</td>
</tr>
<tr>
<td>DNA-binder</td>
<td>73.58%</td>
<td>66.47%</td>
<td>80.36%</td>
<td>0.470</td>
</tr>
<tr>
<td>DNA-Port</td>
<td>72.55%</td>
<td>82.67%</td>
<td>59.76%</td>
<td>0.440</td>
</tr>
<tr>
<td>StackDPPred</td>
<td>89.96%</td>
<td>91.12%</td>
<td>88.80%</td>
<td>0.799</td>
</tr>
<tr>
<td>Our method (k=1)</td>
<td>78.83%</td>
<td>82.63%</td>
<td>75.23%</td>
<td>0.580</td>
</tr>
<tr>
<td>Our method (k=2)</td>
<td>80.62%</td>
<td>84.75%</td>
<td>76.70%</td>
<td>0.620</td>
</tr>
<tr>
<td>Our method (k=8)</td>
<td>89.37%</td>
<td>89.19%</td>
<td>89.54%</td>
<td>0.790</td>
</tr>
<tr>
<td>Our method (k=12)</td>
<td>92.00%</td>
<td>92.08%</td>
<td>91.93%</td>
<td>0.840</td>
</tr>
</tbody>
</table>

When $k = 1$ (the parameters of K-PSSM-Composition), we varied the number of features from 10 to 100 by using the SVM-RFE + CBR technique. The highest accuracy was found when the number of the reduced features was set to 76. Fig. 3 shows the plot of the accuracy against the number of reduced features by utilizing this recursive feature selection algorithm with the SVM classifier.

FIGURE 3. The accuracy of different dimension features on PDB1075 dataset (Jackknife test evaluation).

We then compared the performance of the following feature selection techniques: SVM-RFE, SVM-RFE + CBR, and no feature elimination. We performed a 5-fold cross validation for these experiments and applied different feature elimination techniques to the benchmark PDB1075 dataset (Table 5).

Table 5. Comparison of performance of different feature selection methods on the PDB1075 dataset using Jackknife test evaluation (K=1). The best values are in boldface.

<table>
<thead>
<tr>
<th>Methods</th>
<th>ACC(%)</th>
<th>SE(%)</th>
<th>SP(%)</th>
<th>MCC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Feature Selection</td>
<td>75.55%</td>
<td>78.1%</td>
<td>71.1%</td>
<td>0.491</td>
<td>0.82</td>
</tr>
<tr>
<td>SVM-RFE</td>
<td>77.4%</td>
<td>82.67%</td>
<td>72.36%</td>
<td>0.552</td>
<td>0.84</td>
</tr>
<tr>
<td>SVM-RFE+CBR</td>
<td>79.57%</td>
<td>95.7%</td>
<td>63.44%</td>
<td>0.625</td>
<td>0.85</td>
</tr>
</tbody>
</table>

As can be seen, the SVM-RFE + CBR technique is the best among the feature elimination techniques used. For a better view, we also presented the receiver operating curve (ROC) of each method for the benchmark dataset in Figure 4. As can be seen at k=1, the method SVM-RFE + CBR achieves an auROC of 0.85, better than that of SVM-RFE with auROC 0.84 and No Feature Selection with auROC 0.82. As the value of k increases, the dimension of the feature also increases, and more redundant features may be captured. Therefore, we adopted SVM-RFE + CBR throughout this study.

FIGURE 4. ROC curve of different feature selection methods on PDB1075 dataset (Jackknife test evaluation).

C. Parameter optimization

In this section, we optimized the parameters $k$ of the proposed feature representation algorithm K-PSSM-composition (Section 2.4). To optimize these parameters, we implemented the proposed method on the benchmark datasets PDB1075 and PDB1063, varied $k$ from 1 to 18, and evaluated the predictive performance by conducting jackknife tests.
Table 6 presents the predictive results of our method for different k values. For PDB1063, the maximum ACC and MCC were 92.00% (k = 12) and 0.840 (k = 12), respectively, and for PDB1075, the maximum ACC and MCC were 90.51% (k = 18) and 0.810 (k = 18), respectively. We also plotted in Fig. 5 the accuracies (ACCs) and Matthew correlation coefficient (MCCs) for different k values. As can be seen, when k is between 8 and 18, the ACC and MCC are relatively stable. Therefore, k = 8 was set as the default parameter to generate the proposed features. In particular, the default parameter (k = 8) yields a 3200D (8 × 400) feature vector for a query protein.

![Graph of ACC and MCC of different K values on PDB1075 and PDB1063.](image)

**FIGURE 5.** The ACC and MCC of different K values on PDB1075 and PDB1063 (Jackknife test evaluation).

### Table 6. Results of different K in K-PSSM-Composition feature selection (evaluated on the benchmark dataset PDB1075 and PDB1063). The best values are in boldface.

<table>
<thead>
<tr>
<th>PDB1075</th>
<th>PDB1063</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC(%)</td>
</tr>
<tr>
<td>K=1</td>
<td>78.6%</td>
</tr>
<tr>
<td>K=2</td>
<td>79.63%</td>
</tr>
<tr>
<td>K=3</td>
<td>80.56%</td>
</tr>
<tr>
<td>K=4</td>
<td>82.33%</td>
</tr>
<tr>
<td>K=5</td>
<td>83.91%</td>
</tr>
<tr>
<td>K=6</td>
<td>85.86%</td>
</tr>
<tr>
<td>K=7</td>
<td>86.14%</td>
</tr>
<tr>
<td>K=8</td>
<td>89.77%</td>
</tr>
<tr>
<td>K=9</td>
<td>86.88%</td>
</tr>
<tr>
<td>K=10</td>
<td>88.65%</td>
</tr>
<tr>
<td>K=11</td>
<td>89.86%</td>
</tr>
<tr>
<td>K=12</td>
<td>88.47%</td>
</tr>
<tr>
<td>K=13</td>
<td>89.49%</td>
</tr>
<tr>
<td>K=14</td>
<td>88.65%</td>
</tr>
<tr>
<td>K=15</td>
<td>86.56%</td>
</tr>
<tr>
<td>K=16</td>
<td>89.49%</td>
</tr>
<tr>
<td>K=17</td>
<td>89.30%</td>
</tr>
<tr>
<td>K=18</td>
<td>90.51%</td>
</tr>
</tbody>
</table>

### D. Discussion

Machine learning-based DNA binding protein prediction methods usually predict DNA binding proteins based on their sequences. The prediction is often formulated as a two-class classification problem, in which a critical issue is how to vectorize the protein sequence. Different vectorization methods will greatly affect the performance of the machine learning model. Currently, widely accepted sequence vectorization methods are based on the extracted feature vector of the global information of the sequence. However, studies have shown that local regions of proteins are conservative, which are also important for the prediction. In this paper, a new protein sequence vectorization method K-PSSM-compositions was proposed and applied to predict DNA binding proteins. In order to extract local features, the average segmentation method was used to divide the PSSM matrix into multiple sub-matrices. The improved PSSM-Composition method was applied to the sub-matrix to vectorize the protein sequences, and the obtained feature vectors capture locally conserved information, evolutionary information, and the information between different amino acid residues of a protein sequence. Using SVM as a classifier, we demonstrated that our
method outperforms other state-of-the-art methods in prediction accuracy on multiple datasets and validation methods. Our method also has better scalability and robustness, as well as high interpretability.

IV. Conclusions

DNA-binding proteins play crucial roles in many biological activities. In this study, we have proposed a novel feature representation algorithm named K-PSSM-Composition that addresses the challenging problem of differentiating DNA- and non-DNA-binding proteins. The proposed feature representation algorithm can efficiently capture information about 20 amino acid residues and local information regarding a particular sequence during an evolutionary process by segmenting an originally large PSSM matrix into several submatrices of equal sizes. A rigorous jackknife test and an independent test show that our method has an enhanced performance of DNA-binding protein identification and provides an effective basis for future studies on DNA-binding proteins.

As pointed out in [94] and demonstrated in a series of recent publications (see, e.g., [58, 59, 62-64, 85, 95-98]), user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful prediction methods and computational tools. Actually, many practically useful web-servers have significantly increased the impacts of bioinformatics on medical science [46], driving medicinal chemistry into an unprecedented revolution [52], we shall make efforts in our future work to provide a web-server for the prediction method presented in this paper. The source code and data of our method can be found at https://github.com/Excelsior511/DNA-Binding-Proteins.

REFERENCES


[38] B. Liu et al., "sDNA-Protid: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition," *Plos One*, vol. 9, no. 9, p. e106691, 2014.


