This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TargetDBP: Accurate DNA-Binding Protein Prediction via Sequence-based Multi-View Feature Learning

Jun Hu, Xiao-Gen Zhou, Yi-Heng Zhu, Dong-Jun Yu, and Gui-Jun Zhang

Abstract—Accurately identifying DNA-binding proteins (DBPs) from protein sequence information is an important but challenging task for protein function annotations. In this paper, we establish a novel computational method, named TargetDBP, for accurately targeting DBPs from primary sequences. In TargetDBP, four single-view features, i.e., AAC (Amino Acid Composition), PsePSSM (Pseudo Position-Specific Scoring Matrix), PsePRSA (Pseudo Predicted Relative Solvent Accessibility), and PsePPDBS (Pseudo Predicted Probabilities of DNA-Binding Sites), are first extracted to represent different base features, respectively. Secondly, differential evolution algorithm is employed to learn the weights of four base features. Using the learned weights, we weightedly combine these base features to form the original super feature. An excellent subset of the super feature is then selected by using a suitable feature selection algorithm SVM-REF+CBR (Support Vector Machine Recursive Feature Elimination with Correlation Bias Reduction). Finally, the prediction model is learned via using support vector machine on the selected feature subset. We also construct a new gold-standard and non-redundant benchmark dataset from PDB database to evaluate and compare the proposed TargetDBP with other existing predictors. On this new dataset, TargetDBP can achieve higher performance than other state-of-the-art predictors. The TargetDBP web server and datasets are freely available at http://csbio.njust.edu.cn/bioinf/targetdbp/ for academic use.

Index Terms—DNA-binding protein prediction, Sequence-based, Differential evolution, Feature selection, Support vector machine

1 Introduction

INTERACTIONS between proteins and DNA are indispensable for biological activities and play vital roles in a wide variety of biological processes, such as gene regulation, DNA replication and repair [1]. Hence, accurately targeting DNA-binding proteins (DBPs) is of significant importance for the annotation of protein functions. Tremendous wet-lab efforts have been made to uncover the intrinsic mechanism of protein-DNA interactions. However, identification of DBPs via wet-lab experimental technologies is often cost-intensive and time-consuming. Facing the difficulty in experimentally identifying DBPs and the avalanche of new protein sequences generated in the post-genomic age [2], it is highly desired to develop an automatic computational method for rapidly and accurately targeting DBPs.

During the past decades, many computational methods have been emerged for targeting DBPs [3-5]. These methods can be roughly grouped into two categories according to the features they used: structure-based methods and sequence-based methods. Note that the structure-based methods, e.g., DBD-Hunter [5] and iDBPs [6], generally make use of both the structural and sequential information of target proteins, while the sequence-based methods solely employ the protein sequence information. For the structure-based methods, although they can show promising predictive performance, their application is limited, since the structural information of proteins is not always available. In contrast, the sequence-based methods can overcome this shortcoming by only using the sequence information as input for the DBP prediction. Since the gap between protein sequences and structures fast continues to widen in the post-genomic age, the development of sequence-based DBP predictors has become a hot topic in bioinformatics.

In the recent, many researchers have proposed a series of sequence-based methods to predict DBPs. To name a few: iDNA-Prot [7], PseDNA-Pro [8], iDNAPro-PseAAC [9], iDNA-Prot|dis [10], Local-DPP [11], PSFM-DBT [12], HMMBinder [13], IKP-DBPPred [3], iDNAProt-ES [14], DPP-PseAAC [15], and the methods proposed in references [16-18]. These methods often use only sequence information and recognize DBPs with one or more machine-learning algorithms, such as support vector machine (SVM) [14] or random forest (RF) [7, 11]. For example, in iDNA-Prot [7], the authors first employed a grey system theory [19] to extract the novel pseudo amino acid composition for representing the feature of each protein data and then adopted the algorithm of RF to train the final prediction model. In PseDNA-Pro [8], three kinds of sequence-based information, i.e., amino acid composition, pseudo amino acid composition (PseAAC) [20, 21], and physicochemical distance transformation, are used to rep-

- *J. Hu, Z.G. Zhou, and G.J. Zhang are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023, China. E-mail: hujunum@zjut.edu.cn, zxg@zjut.edu.cn, and zgj@zjut.edu.cn.*
- *Y.H. Zhu and D.J. Yu are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P. R. China. E-mail: zhy930505@njust.edu.cn and njyudj@126.com.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

resent the feature vector of each protein data and the SVM algorithm is then employed to generate the prediction model. In iDNAPro-PseAAC [9], the information of pseudo amino acid composition and profile-based protein representation [22] were integrated to represent the numerical feature of each protein for predicting DBPs based on the SVM algorithm. In HMMBinder [13], the monogram and bigram features were first extracted from the HMM profiles, which were generated by HHblits [23] based on the sequence information, and then were fed into the SVM algorithm for predicting DBPs. In iDNAProt-ES [14], the PSSM-based evolutionary information and sequence-driven structural information, generated by PSI-BLAST and SPIDER2, respectively, were extracted to represent each protein; then, the recursive feature elimination was utilized to select the optimal subset of features; finally, the SVM algorithm were used to learn the final model. In DPP-PseAAC [15], the authors first used Chou's general PseAAC [20, 21] to represent protein data; then, the RF and SVM-RFE (support vector machine recursive feature elimination) algorithms were employed to rank features; finally, the SVM algorithm was utilized to generate the final prediction model. Despite the promising results of these methods, there remains room for further improvements in accurately predicting DBPs from sequence information.

To further improve the accuracy of DBP prediction, in this study, we propose a new sequence-based predictor, named TargetDBP. To make the proposed TargetDBP to be a useful sequence-based DBP predictor, we should observe the Chou's 5-step rule [20] used in a series of recent publications [24-26], i.e., making the following five steps clear: (1) how to construct or select a valid benchmark dataset to train and test the predictor; (2) how to formulate the biological sequence samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target to be predicted; (3) how to introduce or develop a powerful algorithm (or engine) to operate the prediction; (4) how to properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (5) how to establish a user-friendly web-server for the predictor that is accessible to the public.

Following the Chou's 5-step rule, a new gold-standard and non-redundant benchmark data set consisting of 1,200 DBPs and 1,200 non-DBPs is first collected from Protein Data Bank (PDB) [27] and used to train and test the prediction model of TargetDBP. Secondly, we generate the feature representation of each protein by using the following steps: (1) four sequence-based single-view features, i.e., AAC, PsePSSM, PsePRSA, and PsePPDBS, are extracted to represent different base features, respectively; (2) differential evolution algorithm [28] is employed to learn the weights of four base features; (3) using these learned weights, we can weightedly combine the four base features to form the original super feature; (4) a suitable feature selection algorithm, SVM-REF+CBR, is employed to select an excellent subset of the super feature for further quantifying the difference between DBPs and non-DBPs. Thirdly, the prediction model is learned via using the algorithm of support vector machine on the selected feature subset. Then, the leave-one-out cross-validation and independent dataset tests are used to systematically examine the strengths and weaknesses of TargetDBP. Finally, we have established an online web server of TargetDBP, which can be freely accessible for academic use at http://csbio.njust.edu.cn/bioinf/targetdbp/.

2 Materials and Methods

2.1 Benchmark datasets

The first important step of the statistical predictor development is to establish a comprehensive, reliable, and stringent benchmark dataset [10]. In this paper, the benchmark dataset $\mathbf{S}$ for DBP prediction can be formally denoted as following:

$$\mathbf{S} = \mathbf{S}_{posi} \cup \mathbf{S}_{nega} \quad (1)$$

where $\mathbf{S}_{posi}$ means the positive subset that only contains DBPs, $\mathbf{S}_{nega}$ means the negative subset that only includes non-DBPs, and the symbol $\cup$ represents the "union" in the set theory.

In order to construct $\mathbf{S}_{posi}$, we first extract all DBP chains from PDB [27] (as of May 12, 2018). Here, each DBP chain is classified into the positive class (i.e., DNA-binding protein) in PDB [27] or contains one DNA-binding residue at least, which includes at least one of its non-hydrogen atoms whose distance to at least one non-hydrogen atom of the DNA molecule is less than the sum of the Van Der Waals radii of the two corresponding atoms plus a tolerance of 0.5 Å. Then, the CD-hit software [29] is used to remove the redundant protein chains such that no two chains have more than 25 percent sequence identity. To ensure no fragment in the final dataset, any chain with less than 50 residues in length is removed. Besides, we also remove the proteins containing the residue 'X' since they include unknown residue. Finally, a total of 1,200 non-redundant protein chains are obtained to form $\mathbf{S}_{posi}$. The $\mathbf{S}_{posi}$ is divided into a positive training subset ($\mathbf{S}^{tr}_{posi}$) and a positive independent validation subset ($\mathbf{S}^{tst}_{posi}$). $\mathbf{S}^{tr}_{posi}$ consists of 1,052 protein chains, which were all deposited into the PDB before May 12, 2014, while $\mathbf{S}^{tst}_{posi}$ includes 148 protein chains, which were all deposited into the PDB after May 12, 2014.

According to the same steps as described above, we can obtain 16,058 non-DBP chains from PDB, where the sequence identity of each two chains is less than 25%. Here, each non-DBP chain is not annotated to the class of DBP in PDB [27] and contains no one DNA-binding residue. To construct $\mathbf{S}_{nega}$, we first select 1,052 chains, which were all deposited into the PDB before May 12, 2014, from these non-DBP chains to form a negative training subset $\mathbf{S}^{tr}_{nega}$. We then choose 148 chains, which were all deposited into the PDB after May 12, 2014, to form a negative independent validation subset $\mathbf{S}^{tst}_{nega}$. Finally, $\mathbf{S}_{nega} = \mathbf{S}^{tr}_{nega} \cup \mathbf{S}^{tst}_{nega}$.

In this paper, the benchmark dataset $\mathbf{S}$ can be also presented as following:

$$\mathbf{S} = \mathbf{S}^{tr} \cup \mathbf{S}^{tst} \quad (2)$$

where $\mathbf{S}^{tr} = \mathbf{S}^{tr}_{posi} \cup \mathbf{S}^{tr}_{nega}$ means the training dataset, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

is used to train the DBP prediction model and test the TargetDBP performance by the leave-one-out cross-validation test (i.e., jackknife test), and $\mathbf{S}^{tst} = \mathbf{S}^{tst}_{posi} \cup \mathbf{S}^{tst}_{nega}$ represents the independent validation dataset, which is employed to test the prediction performance by independent test. The protein name list of the benchmark dataset $\mathbf{S}$ can be found in Text S1 in the Supporting Information (SI). Moreover, the corresponding protein sequence information can be easily downloaded from the online web server of TargetDBP, whose address is http://csbio.njust.edu.cn/bioinf/targetdbp/ .

2.2 Feature Extraction

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 (such as SVM [30], RF [31]) can only handle vectors as elaborated in a comprehensive review [32]. However, a simple vector (such as AAC [33, 34]) defined in a discrete model may completely lose all the sequence-pattern information. To avoid completely losing the sequence-pattern information for proteins, lots of methods, such as DPC (dipeptide composition) [34], TPC (tripeptide composition) [35], PseAAC [20, 36], and PsePSSM [37], have been proposed to represent the feature of each protein sequence. Recently, two powerful web-servers, i.e., Pse-in-One [38] and BioSeq-Analysis [39], have been established to make us to generate the feature of protein sequence easily.

In this study, we employed four effective single-view features, i.e., AAC, PsePSSM, PsePRSA, and PsePPDBS, to represent four different base features of each protein, respectively. The details of extracting these single-view features are described as follows:

2.2.1 Amino Acid Composition

Amino acid composition (AAC) is a classical sequence-based feature view that has been widely employed in many protein attribute prediction tasks including DBP prediction [4]. Let AA_1, AA_2, …, AA_{19}, and AA_{20} be the 20 ordered native amino acid types (i.e., *A*, *C*, …, *W*, and *Y*), n_i be the number of occurrence of AA_i in the given protein sequence, and L be the protein length. The AAC feature ($\mathbf{f}_{AAC}$) of the protein is then formulated to be a vector with 20 elements, as follows:

$$\mathbf{f}_{AAC} = \left(\frac{n_1}{L}, \frac{n_2}{L}, \cdots, \frac{n_{20}}{L}\right)^{\mathrm{T}} \tag{3}$$

where T means the transpose of the vector. The dimensionality of the AAC-view feature is 20.

2.2.2 Pseudo Position-Specific Scoring Matrix

Position-specific scoring matrix (PSSM) is widely used to extract the evolutionary information of a given protein sequence, which has been proved to be highly effective in a variety of prediction tasks in the field of bioinformatics, including the DBP prediction [11], protein-ligand binding site prediction [40, 41], protein contact map prediction [42], and so on. However, when developing a machine-learning-based DBP predictor, such as TargetDBP in this study, PSSM, whose length is equal to that of the given protein sequence, cannot directly be employed to represent the feature vector. This is because, on the one hand, the lengths of different protein sequences are generally different and these lengths vary widely; but on the other hand, all existing machine-learning algorithms, such as SVM [30], RF [31], and K-nearest neighbor (KNN) [43], cannot handle the length-different information, such as PSSM.

To deal with the above dilemma, Chou and Shen [37] proposed the PsePSSM approach, which has been demonstrated to be very useful for the most areas of computational proteomics, such as protein crystallization prediction [33] and protein structural class prediction [44]. The details of generating PsePSSM feature are described as follows:

For one given protein sequence consisting of L amino acids, its PSSM profile can be generated by using the PSI-BLAST software [45] to search against the non-redundant NCBI database through three iterations, with 0.001 as the *e-value* cutoff for multiple sequence alignment. The logistic function, i.e., $f(x)=1/(1+e^{-x})$, is then used to normalize the score of each element, denoted as x, in a PSSM profile. Let $\{p_{i,j}\}_{L\times 20}$ denote the normalization PSSM ($\mathrm{PSSM}_{\mathrm{norm}}$) of the given protein sequence. The PsePSSM can be constructed with the following two steps [44]:

Step 1: Calculating the PSSM Composition

The PSSM composition ($\mathbf{u}^{PSSM}$) is the vector with 20 elements as defined:

$$\mathbf{u}^{PSSM} = \left(u_1^{PSSM}, u_2^{PSSM}, \cdots, u_{20}^{PSSM}\right)^{\mathrm{T}} \tag{4}$$

where $u_j^{PSSM} = \sum_{k=1}^{L} p_{k,j} / L$, $1 \le j \le 20$.

Step 2: Calculating Correlation Factors

Although the evolutionary information of the given protein sequence can be partially reflected by $\mathbf{u}^{PSSM}$, all the sequence-order information during the evolution process of the protein are fully lost. In order to remedy this defect and extract the sequence-order information, we calculate the g-tier correlation factor of each column of the $\mathrm{PSSM}_{\mathrm{norm}}$ matrix as follows:

$$\boldsymbol{\xi}^g = (\xi_1^g, \xi_2^g, \cdots, \xi_{20}^g)^{\mathrm{T}} \tag{5}$$

where $\xi_j^g = \sum_{k=1}^{L-g} (p_{k,j} - p_{k+g,j})^2 / (L-g)$, $1 \le j \le 20$, $0 \le g \le G$, $G \le L$. The scalar quantity ξ_j^g is the correlation factor by coupling the g-most contiguous PSSM scores along the protein sequence for the amino acid type j.

Finally, the PsePSSM of the given protein sequence, denoted as $\mathbf{f}_{PsePSSM}$, is the combination of its PSSM composition ($\mathbf{u}^{PSSM}$) and the G correlation factor vectors ($\{\boldsymbol{\xi}^g\}_{g=1}^{G}$) as follows:

$$\mathbf{f}_{PsePSSM} = \begin{pmatrix} \mathbf{u}^{PSSM} \\ \boldsymbol{\xi}^1 \\ \boldsymbol{\xi}^2 \\ \vdots \\ \boldsymbol{\xi}^G \end{pmatrix} \tag{6}$$

Considering the fact that there is no theoretical justifi-

cation on determining the optimal value *G*, we empirically tested different values of *G* from 1 to 9 with a step of 1 on the training dataset (i.e., $\mathbf{S}^{tr}$) and found that the optimal value of *G* is 6 (see details in Text S2 in SI). Accordingly, the dimensionality of the final PsePSSM ($\mathbf{f}_{PsePSSM}$) is $20+6\times 20=140$.

2.2.3 Pseudo Predicted Relative Solvent Accessibility

The solvent accessibility [46] has close relevance to the spatial arrangement of a protein, the characteristics of residues packing, and the protein-DNA interactions. Many researches [4, 47] have experimentally demonstrated that the solvent accessibility has a positive impact to the DBP prediction.

In this study, the solvent accessibility of each residue is evaluated by the predicted relative solvent accessibility (PRSA) via feeding the corresponding sequence to the standalone SANN program [48], downloaded from http://lee.kias.re.kr/~newton/sann/. For each given sequence, the SANN program accurately predicts its PRSA profile (*L* rows and 3 columns, where *L* is the length of the given sequence), which includes the probabilities of three solvent accessibility classes (i.e., buried (B), intermediate (I), and exposed (E)) of each residue. Similar to the generation procedure of PsePSSM, we also extract the pseudo predicted relative solvent accessibility (PsePRSA) from the PRSA profile as follows:

Let $\{a_{i,j}\}_{L\times 3}$ be the PRSA profile. The PsePRSA feature vector, denoted as $\mathbf{f}_{PsePRSA}$, can be formulated as follows:

$$\mathbf{f}_{PsePRSA}=\begin{pmatrix}\mathbf{u}^{PRSA}\\ \mathbf{o}^{1}\\ \mathbf{o}^{2}\\ \vdots\\ \mathbf{o}^{H}\end{pmatrix} \tag{7}$$

where $\mathbf{u}^{PRSA}=(u_1^{PRSA},u_2^{PRSA},u_3^{PRSA})^T$, $u_j^{PRSA}=\sum_{k=1}^{L}a_{k,j}/L$, $\mathbf{o}^h=(o_1^h,o_2^h,o_3^h)^{\mathrm{T}}$, $o_j^g=\sum_{k=1}^{L-h}(a_{k,j}-a_{k+h,j})^2/(L-h)$, $1\le j\le 3$, $1\le h\le H$, $H\le L$. Then, we tested different values of *H* from 1 to 9 with a step of 1 on the training dataset (i.e., $\mathbf{S}^{tr}$) and found that the optimal value of *H* is 4 (see details in Text S3 in SI). Finally, the dimensionality of the PsePRSA is $3+4\times 3=15$.

2.2.4 Pseudo Predicted Probabilities of DNA-Binding Sites

Theoretically, we can correctly target all DBPs, when the DNA-binding sites (DBSs) of these proteins can be predicted with the accuracy of 100 percent. However, most of the state-of-the-art DBS predictors can only achieve around 80% accuracy. Directly using the prediction results of DBS predictors to target the DBPs is not the best way to help us to improve the accuracy of DBP prediction. In this study, we employ the prediction probability results of DBS predictor to act as a new feature view and extract the useful feature vector from it to enhance the accuracy of DBP prediction.

To avoid the obviously high accuracy of DBS prediction caused by the potential high sequence identity between the testing proteins and the training proteins of the existing DBS predictors, we use our training dataset to train a new DBS prediction model rather than chose one of the existing DBS predictors. Inspired by our previous work, i.e., TargetDNA [49], we first extract the discriminative feature of each residue of each training protein from the views of PSSM and PRSA with a sliding window of size 9. Then, we employ the SVM algorithm [30, 50] to train the DBS prediction model. Finally, we use this model to gain the probability of each residue of each testing protein being classified into the class of DBS. It is importantly noted that we obtain the predicted probabilities of DBSs (PPDBS) for the training proteins via using the leave-one-out cross-validation test.

For one given protein sequence with *L* amino acids, we can obtain the corresponding PPDBS, denoted as $\{s_j\}_{j=1}^{L}$, using the above procedures, where s_j means the probability of the *j*-th residue belonging to DBS. We then extract the feature of pseudo PPDBS (PsePPDBS) by the following steps:

Step 1: Calculating the PPDBS Composition

The PPDBS composition ($\mathbf{u}^{PPDBS}$) is the vector with 20 elements as defined:

$$\mathbf{u}^{PPDBS}=\left(u_1^{PPDBS},u_2^{PPDBS},\cdots,u_{20}^{PPDBS}\right)^T \tag{8}$$

where $u_i^{PPDBS}=\sum_{k=1}^{L}s_k\cdot T_k(AA_i)/L$, AA_i means *i*-th order amino acid type (see details in Section 2.2.1), $T_k(AA_i)$=1 when the amino acid type of the *k*-th residue of the given protein is equal to AA_i, otherwise $T_k(AA_i)$=0, and $1\le i\le 20$.

Step 2: Calculating Correlation Factors

To save some parts of the sequence-order information of the given protein, we calculate the *m*-tier correlation factor of the PPDBS as follows:

$$\eta^m=\sum_{k=1}^{L-m}(s_k-s_{k+m})^2/(L-m) \tag{9}$$

where $0\le m\le M$ and $M\le L$.

Finally, the PsePPDBS of the given sequence, denoted as $\mathbf{f}_{PsePPDBS}$, is the combination of $\mathbf{u}^{PPDBS}$ and the *M* correlation factors ($\{\eta^m\}_{m=1}^{M}$) as follows:

$$\mathbf{f}_{PsePPDBS}=\begin{pmatrix}\mathbf{u}^{PPDBS}\\ \eta^{1}\\ \eta^{2}\\ \vdots\\ \eta^{M}\end{pmatrix} \tag{10}$$

To ensure the optimal value of the parameter *M*, we empirically tested different values of *M* from 1 to 9 with a step of 1 on the training dataset (i.e., $\mathbf{S}^{tr}$) and found that the value optimal *M* is 4 (see details in Text S4 in SI). Accordingly, the dimensionality of the final PsePPDBS is $20+4\times 1=24$.

2.3 Learning Weights for Combining Multi-View Features

It is believed that the above four single-view base features potentially contain the complementary discriminative information for predicting DBPs. How to combine the four base features is one of the most crucial steps in generating a machine-learning-based DBP prediction model. The most straightforward and convenient method is to serially and directly combine these base features to gain a super feature (i.e., AAC+PsePSSM+PsePRSA+PsePPDBS)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

to be employed for training a DBP prediction model. Here, '+' means simply serial combination. However, the simple combination method cannot guarantee to obtain an optimal discriminative capability, since it neglects the relative importance of these base features. Hence, learning the relative importance of these base features would be especially useful for improving the accuracy of DBP prediction.

In this study, we employ the differential evolution (DE) algorithm [28, 51], which is one of the most competitive variants of evolution algorithms, to learn the optimal/suboptimal weights of these base features, since it is easy to be implemented and achieve high performance [52, 53]. Furthermore, DE has been used to help numerous research fields, such as protein structure prediction [54] and sensor network localization [55], to achieve a positive impact. The procedure of using DE to learn the suitable weights of these single-view features in this study is briefly described as follows:

Let $f(\mathbf{w})$ be the problem that is to search the maximum *MCC* value of the leave-one-out cross-validation test on $\mathbf{S}^{tr}$, $\mathbf{w}=(w_1,w_2,w_3,w_4)^T \in [-2,2]^4$ is one candidate solution, where w_1, w_2, w_3, and w_4 mean the weights of AAC, PsePSSM, PsePRSA, and PsePPDBS feature views, respectively. Then, the steps of DE can be described as follows:

Step 1: Initialization

Randomly generate the initial population $P^g=\{\mathbf{w}_1^g,\mathbf{w}_2^g,\cdots\mathbf{w}_N^g\}$, where $\mathbf{w}_i^g=(w_{i,1}^g,w_{i,2}^g,w_{i,3}^g,w_{i,4}^g)^{\mathrm{T}}$ means the *i*-th solution in the *g*-th generation population, *N* is the population size. Set the values of scaling factor (*F*), crossover rate (*CR*), and maximum generation (G_{max}) to be 0.5, 0.5, and 1000, respectively. Set the generation count, i.e., *g*, to be 1.

Step 2: Mutation

For each solution $\mathbf{w}_i^g$ in the population P^g, a mutant vector ($\mathbf{m}_i^{g+1}$) is generated according to

$$\mathbf{m}_i^{g+1}=\mathbf{w}_{r_1}^g+F\cdot(\mathbf{w}_{r_2}^g-\mathbf{w}_{r_3}^g) \tag{11}$$

where $\mathbf{w}_{r_1}^g$, $\mathbf{w}_{r_2}^g$, and $\mathbf{w}_{r_3}^g$ are three different solutions randomly selected from the set of $P^g-\{\mathbf{w}_i^g\}$.

Step 3: Crossover

To increase the diversity of the solutions in the next generation P^{g+1}, the step of crossover is introduced in DE algorithm. For each solution $\mathbf{w}_i^g$ in P^g, a trial vector $\mathbf{t}_i^{g+1}=(t_{i,1}^{g+1},t_{i,2}^{g+1},t_{i,3}^{g+1},t_{i,4}^{g+1})$ is formed as follows:

$$t_{i,k}^{g+1}=\begin{cases} m_{i,k}^{g+1}, & if\ R_j<CR\ or\ k=k_{rand} \\ w_{i,k}^g, & otherwise \end{cases} \tag{12}$$

where R_j is a uniform random number generator with outcome $\in[0,1]$ for the *j*-th element. $k_{rand}\in\{1,2,3,4\}$ means a randomly selected index, which ensures that $\mathbf{t}_i^{g+1}$ gets at least one element from $\mathbf{m}_i^{g+1}$.

Step 4: Selection

For each solution $\mathbf{w}_i^g$ in P^g, to decide whether or not it should be remained to become a member, i.e., $\mathbf{w}_i^{g+1}$, of the next generation population, i.e., P^{g+1}, the corresponding trial vector $\mathbf{t}_i^{g+1}$ is compared to the solution $\mathbf{w}_i^g$ based on the output of $f(\mathbf{w})$. If $f(\mathbf{w}_i^g)$ is larger than $f(\mathbf{t}_i^{g+1})$, $\mathbf{w}_i^g$ is retained to be $\mathbf{w}_i^{g+1}$; otherwise, the trial vector $\mathbf{t}_i^{g+1}$ is set to $\mathbf{w}_i^{g+1}$.

Step 5: Loop or Termination

If *g* is larger than G_{max}, the procedure of DE algorithm is terminated and the best solution $\mathbf{w}_{best}^g$ in P^g should be outputted; otherwise, *g*=*g*+1 and repeat *Steps 2-4*.

After the DE procedure is terminated, we can obtain the final solution $\mathbf{w}_{best}^{G_{max}}$. In this study, the $\mathbf{w}_{best}^{G_{max}}$ is (0.275, 0.168, 0.164, 1.489)$^{\mathrm{T}}$. That is, the weights of AAC, PsePSSM, PsePRSA, and PsePPDBS are 0.275, 0.168, 0.164, and 1.489, respectively. Based on $\mathbf{w}_{best}^{G_{max}}$, we can generated a new super feature, denoted as *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS, by weightedly and serially combining the features of AAC, PsePSSM, PsePRSA, and PsePPDBS.

2.4 Feature Selection Using SVM-RFE+CBR

Feature selection is a widely-used technique in the fields of machine learning and pattern recognition, since it can enhance the performance of the prediction model by removing irrelevant, noisy, and redundant information from the original feature space. In order to further improve the performance of DBP prediction, we employ the feature selection algorithm named support vector machine recursive feature elimination with correlation bias reduction (SVM-RFE+CBR) [56], which has been successfully applied in DBP prediction [57], to select one excellent feature subset from the weighted feature vector, i.e., *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS. As described in [57], SVM-REF+CBR is the enhanced version of SVM-RFE algorithm [58]. SVM-RFE+CBR has both linear and nonlinear versions. The nonlinear SVM-RFE+CBR employs a special kernel function to transform the nonlinear learning problem in the original feature space to the linear one in the feature space of higher dimensions. In this study, we can gain the output of the nonlinear SVM-RFE+CBR algorithm with a ranked feature list on the training dataset. We then select an optimal feature subset based on the ranked features (see Section 3.3).

2.5 Implementation of Support Vector Machine

Support vector machine (SVM) algorithm [30, 50], a machine learning approach based on the structural risk minimization principle of statistics learning theory, has been used in a wide variety of bioinformatics fields, including DBP prediction [4]. In this study, we also utilize SVM to construct the prediction model. We use the LIBSVM software (version libsvm-3.18) [59], which can be freely downloaded at http://www.csie.ntu.edu.tw/~cjlin/libsvm/, to implement the SVM algorithm and train the model of DBP prediction. Here, a radial basis function is chosen as the kernel function. The kernel width parameter σ and the regularization parameter γ, which are two most important parameters, are optimized over a five-fold cross-validation using a grid search tool in the LIBSVM software.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

2.6 Architecture of TargetDBP

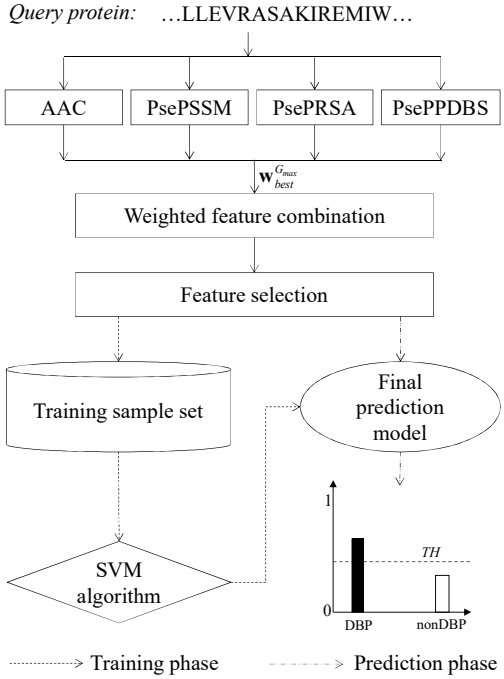


Fig. 1. Architecture of TargetDBP.

Fig. 1 demonstrates the architecture of the proposed TargetDBP. For a given protein, TargetDBP can extract the four different single-view features, i.e., AAC, PsePSSM, PsePRSA, and PsePPDBS, by calling the corresponding programs. Based on the weights $\mathbf{w}_{best}^{G_{max}}$, which is learned by using DE algorithm on the training dataset, the procedure of weighted feature combination can be performed to generate a discriminative feature vector. Then, the SVM-RFE+CBR algorithm is used to choose an optimal feature subset to be the final feature representation of each protein. In training phase, after generating the final features of all proteins in the training dataset $\mathbf{S}^{tr}$, we can obtain the training sample set. We then employ the SVM algorithm to train the final prediction model for targeting DBPs. In prediction phase, for each protein to be predicted, after generating the corresponding final feature vector, the final prediction model can be used to give the probability of classifying it to be a DBP. Finally, the decision is performed based on the probability and the prescribed threshold *TH*: a protein with probability above *TH* is marked as DBP. How to choose the threshold *TH* will be described in Section 2.7.

2.7 Evaluation indexes

In this study, the performance of DBP prediction is assessed by using the following six classical evaluation indexes of binary classification, i.e., Recall (*Rec*), Specificity (*Spe*), Accuracy (*Acc*), Precision (*Pre*), Mathew's Correlation Coefficient (*MCC*) [60], and F1-score (F_1):

$$Rec = \left(1 - \frac{N_-^+}{N^+}\right) \times 100 \tag{13}$$

$$Spe = \left(1 - \frac{N_+^-}{N^-}\right) \times 100 \tag{14}$$

$$Acc = \left(1 - \frac{N_-^+ + N_+^-}{N^+ + N^-}\right) \times 100 \tag{15}$$

$$Pre = \left(1 - \frac{N_+^-}{N^+ - N_-^+ + N_+^-}\right) \times 100 \tag{16}$$

$$MCC = \frac{1 - \left(\frac{N_-^+}{N^+} + \frac{N_+^-}{N^-}\right)}{\sqrt{\left(1 + \frac{N_+^- - N_-^+}{N^+}\right)\left(1 + \frac{N_-^+ - N_+^-}{N^-}\right)}} \tag{17}$$

$$F_1 = 2 \cdot \frac{Pre \cdot Rec}{Pre + Rec} \tag{18}$$

where N^+ is the total number of DBPs, N_-^+ is the number of DBPs incorrectly predicted as non-DBP, N^- is the total number of non-DBPs, and N_+^- is the number of non-DBPs incorrectly predicted as DBP. The *MCC* measure ranges from -1 to 1, and the other five evaluation measures range between 0 to 1. The higher values of these evaluation indexes mean the better performance of DBP prediction. The reported threshold (*TH*), which can maximize the *MCC* value over the leave-one-out cross-validation test on the training dataset $\mathbf{S}^{tr}$, is then chosen to calculate the values of *Rec*, *Spe*, *Acc*, *Pre*, *MCC*, and F_1 in this study. The value of *TH* is 0.48 in this study. Moreover, the area under the receiver operating characteristic (ROC) curve (denoted as AUC), which is a threshold-independent evaluation index, is also used to evaluate the overall ability of DBP prediction.

3 Results and Discussions

3.1 Performance Comparison of Four Single-View Features

In this section, we will investigate the discriminative performances of the four single-view features, i.e., AAC, PsePSSM, PsePRSA, and PsePPDBS. Each feature is evaluated by performing leave-one-out cross-validation test on the training dataset $\mathbf{S}^{tr}$ with the SVM algorithm. Table 1 summarizes the discriminative performance comparison of these single-view features.

TABLE 1
Performance Comparison of Four Single-View Features over Leave-One-Out Cross-Validation Test on the Training Dataset

Feature	*Rec*	*Spe*	*Acc*	*Pre*	*MCC*	F_1
ACC	67.21	68.16	67.68	67.85	0.354	0.675
PsePSSM	72.34	72.24	72.29	72.27	0.446	0.723
PsePRSA	64.26	65.21	64.73	64.88	0.295	0.646
PsePPDBS	77.00	77.09	77.04	77.07	0.541	0.770

From Table 1, we can easily find that PsePPDBS consistently outperforms other three single-view features concerning the six evaluation indexes. Concretely, the *MCC* and F_1 of PsePPDBS are 0.541 and 0.770, which are 52.82% and 14.07% higher than AAC, 21.30% and 6.50% higher than PsePSSM, and 83.39% and 19.20% higher than PsePRSA, respectively.

Fig. 2 also shows the ROC curves of AAC, PsePSSM, PsePRSA, and PsePPDBS. In Fig. 2, it is easy to find that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

the AUC value of PsePPDBS is 0.831, which is higher than that of the other three single-view features. The underlying reason for PsePPDBS to outperform other three single-view features is that PsePPDBS contains the direct related information for DBP prediction.

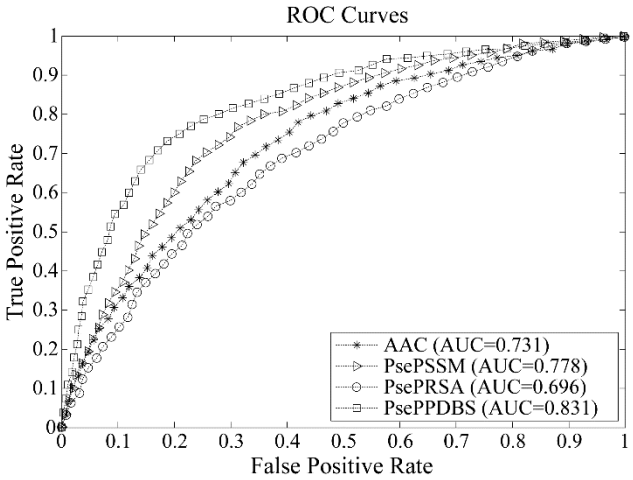


Fig. 2. ROC curves of AAC, PsePSSM, PsePRSA, and PsePPDBS.

3.2 Weighted Feature Combination can Enhance Prediction Performance

In this section, to verify the ability of weighted feature combination, we will compare the discriminative performances of AAC+PsePSSM+PsePRSA+PsePPDBS and *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS. Each feature is tested by performing leave-one-out cross-validation test on $\mathbf{S}^{tr}$ with SVM algorithm. Table 2 summarizes the compared results.

TABLE 2
Performance Comparison between AAC+PsePSSM+PsePRSA+PsePPDBS and *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS over Leave-One-Out Cross-Validation Test on the Training Dataset

Feature	*Rec*	*Spe*	*Acc*	*Pre*	*MCC*	F_1
D-Feature *	76.81	77.19	77.00	77.10	0.540	0.770
W-Feature #	78.52	79.18	78.85	79.04	0.577	0.788

** D-Feature means the AAC+PsePSSM+PsePRSA+PsePPDBS feature*
 # W-Feature means the wAAC+wPsePSSM+wPsePRSA+wPsePPDBS feature

From Table 2, we can observe that *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS is consistently superior to AAC+PsePSSM+PsePRSA+PsePPDBS concerning the six evaluation indexes. The *Rec*, *Spe*, *Acc*, *Pre*, *MCC*, and F_1 of *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS are 78.52, 79.18, 78.85, 79.04, 0.577, and 0.788, which are 2.23%, 2.58%, 2.40%, 2.52%, 6.85%, and 2.34% higher than AAC+PsePSSM+PsePRSA+PsePPDBS, respectively. Fig. 3 also demonstrates the ROC curves of the two combination features. From Fig. 3, we can find that the AUC of *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS is 0.855, which is slightly higher than that of AAC+PsePSSM+PsePRSA+PsePPDBS.

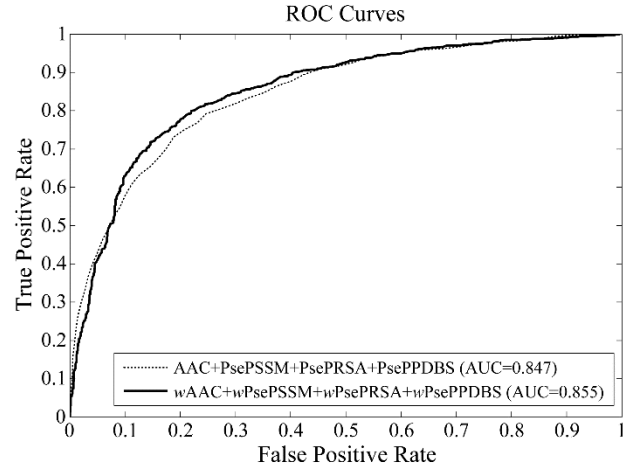


Fig. 3. ROC curves of AAC+PsePSSM+PsePRSA+PsePPDBS and *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS.

By revisiting Table 1, we can find an interesting phenomenon that is the *MCC* value of PsePPDBS is slightly higher than that of the directly serial combination feature, i.e., AAC+PsePSSM+PsePRSA+PsePPDBS. This phenomenon can be understood that directly combining these four single-view features, i.e., AAC, PsePSSM, PsePRSA, and PsePPDBS, cannot further improve the performance of DBP prediction. While, comparing to PsePPDBS, *w*AAC+*w*PsePSSM+*w*PsePRSA+*w*PsePPDBS can obtain a higher prediction performance. The above comparison results can demonstrate that the impact of weighted feature combination should be positive.

3.3 Improving Performance by Feature Selection

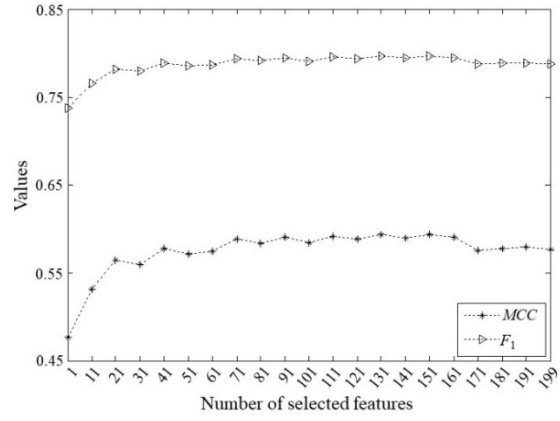


Fig. 4. The variation curves of MCC and F1 values versus the number of selected features based on the SVM-RFE+CBR-ranked features.

Since SVM-RFE+CBR cannot automatically ensure the optimal number (*ON*) of the selected features, we will empirically select the value of *ON* based on the SVM-RFE-CBR ranked features in this section on the training dataset $\mathbf{S}^{tr}$. Concretely, after obtaining the ranked features by using SVM-RFE+CBR algorithm, we evaluate the performance variations of *MCC* and F_1 on the training dataset (i.e., $\mathbf{S}^{tr}$) over leave-one-out cross-validation tests

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

by gradually varying the selected number of the top-ranked features from 1 to 199 with a step size of 10. Fig. 4 shows the variation curves of *MCC* and F_1 values along with the increasing number of selected features over leave-one-out cross-validation test.

From Fig. 4, it can be observed that when the number of selected features (*NSF*) is 131, the corresponding *MCC* and F_1 values based on leave-one-out cross-validation test both achieve the highest values; when $1 \leq NSF < 131$, *MCC* and F_1 values tend to increase with a little fluctuation; when $NSF > 131$, the *MCC* and F_1 values slightly fluctuate and enter a slowly descending state. Thus, the final optimal feature subset determined by feature selection is composed of the 131 top-ranked features, i.e., *ON*=131. In the 131 top-ranked features, there are 18 features selected from the AAC feature view, 84 features selected from PsePSSM feature view, 14 features selected from PsePRSA feature view, and 15 features selected from PsePPDBS feature view. That is, the 90.00%, 60.00%, 93.33%, and 62.50% features in AAC, PsePSSM, PsePRSA, and PsePPDBS feature views are selected to compose the final optimal feature subset, respectively. The main reason for this phenomenon should be that the four feature views are all important for DBP prediction.

TABLE 3
Performance Comparison between the Features of With and Without Feature Selection over Leave-One-Out Cross-Validation Test on the Training Dataset

With/without feature selection	*Rec*	*Spe*	*Acc*	*Pre*	*MCC*	F_1
Without	78.52	79.18	78.85	79.04	0.577	0.788
With	79.56	79.85	79.71	79.79	0.594	0.797

To further demonstrate the efficacy of feature selection, Table 3 and Fig. 5 list the performance comparisons between with and without feature selection on $\mathbf{S}^{tr}$ over leave-one-out cross-validation test. As shown in Table 3, values obtained with feature selection are consistently better than those obtained without feature selection in terms of all six evaluation indexes. Moreover, as demonstrated in Fig. 5, the AUC value of with feature selection is 0.865, which is also 1.17% higher than that of without feature selection. The results shown in Table 3 and Fig. 5 indicate that the performance is indeed enhanced after applying feature selection.

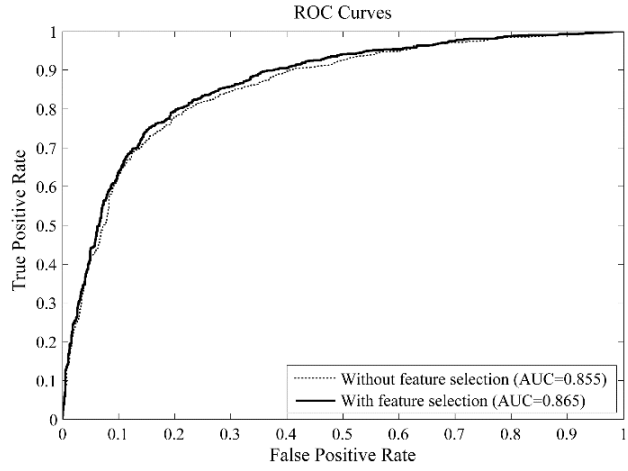


Fig. 5. ROC curves of the features of with and without feature selection over leave-one-out cross-validation test on the training dataset.

TABLE 4
Performance Comparisons between the Proposed TargetDBP and Other Predictors of DNA-Binding Proteins on the Independent Validation Dataset $\mathbf{S}^{tst}$

Predictor	*Rec*	*Spe*	*Acc*	*Pre*	*MCC*	F_1
iDNA-Prot [a]	63.51	60.81	62.16	61.84	0.243	0.627
PseDNA-Pro [b]	78.38	56.08	67.23	64.09	0.354	0.705
iDNAPro-PseAAC [c]	78.38	54.05	66.22	63.04	0.334	0.699
iDNA-Prot|dis [d]	72.30	64.19	68.24	66.88	0.366	0.695
Local-DPP [e]	3.38	93.92	48.65	35.71	-0.06	0.062
PSFM-DBT [f]	71.62	65.54	68.58	67.52	0.372	0.695
HMMBinder [g]	99.32	1.35	50.34	50.17	0.034	0.667
IKP-DBPPred [h]	52.70	63.51	58.11	59.09	0.163	0.557
iDNAProt-ES (on PDB1075) [i]	91.89	51.35	71.62	65.38	0.473	0.764
iDNAProt-ES (on $\mathbf{S}^{tr}$) [j]	95.95	41.22	68.58	62.01	0.444	0.753
DPP-PseAAC [k]	55.41	66.89	61.15	62.60	0.225	0.588
TargetDBP	76.35	77.03	76.69	76.87	0.534	0.766

[a] Results computed using the iDNA-Prot server at http://www.jci-bioinfo.cn/iDNA-Prot/.
[b] Results computed using the PseDNA-Pro server at http://bioinformatics.hitsz.edu.cn/PseDNA-Pro/.
[c] Results computed using the iDNAPro-PseAAC server at http://bioinformatics.hitsz.edu.cn/ iDNAPro-PseAAC/.
[d] Results computed using the iDNA-Prot|dis server at http://bioinformatics.hitsz.edu.cn/iDNA-Prot_dis/.
[e] Results computed using the Local-DPP server at http://server.malab.cn/Local-DPP/Index.html.
[f] Results computed using the PSFM-DBT server at http://bioinformatics.hitsz.edu.cn/PSFM-DBT/.
[g] Results computed using the standalone version of HMMBinder.
[h] Results computed using the IKP-DBPPred server at http://server.malab.cn/IKP-DBPPred/index.jsp.
[i] Results computed using the re-implementation of iDNAProt-ES on PDB1075.
[j] Results computed using the re-implementation of iDNAProt-ES on $\mathbf{S}^{tr}$.
[k] Results computed using the DPP-PseAAC server at http://77.68.43.135:8080/DPP-PseAAC/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2893634, IEEE/ACM Transactions on Computational Biology and Bioinformatics

3.4 Comparing TargetDBP with existing DBP predictors

In this section, we will experimentally demonstrate the efficacy of the proposed TargetDBP by comparing it with other existing DBP predictors, including iDNA-Prot [7], PseDNA-Pro [8], iDNAPro-PseAAC [9], iDNA-Prot|dis [10], Local-DPP [11], PSFM-DBT [12], HMMBinder [13], IKP-DBPPred [3], iDNAProt-ES [14], and DPP-PseAAC [15], on the independent validation dataset, i.e., $\mathbf{S}^{tst}$.

The prediction results of iDNA-Prot [7], PseDNA-Pro [8], iDNAPro-PseAAC [9], iDNA-Prot|dis [10], Local-DPP [11], PSFM-DBT [12], IKP-DBPPred [3], and DPP-PseAAC [15] are obtained by feeding the 296 proteins in $\mathbf{S}^{tst}$ to their corresponding web servers. Since the web servers of HMMBinder and iDNAProt-ES are not working, the results of HMMBinder and iDNAProt-ES are computed using the standalone version of HMMBinder and the re-implementations of iDNAProt-ES, respectively. Here, for an objective and fair comparison, we re-implemented two versions of iDNAProt-ES: one is learned on PDB1075, which is constructed in [10], after removing 12 false data, i.e., 3THWD, 4FCYC, 4GNXK, 4GNXL, 1AOII, 4JJNJ, 4JJNI, 4ESVA, 2G8FA, 4EUWA, 3SWMA, and 4FF1A, and some redundant proteins, e.g., 2AY0B and 2AY0C (see details in Text S5 in SI); the other is learned on $\mathbf{S}^{tr}$. Table 4 demonstrates the performance comparisons between TargetDBP with other predictors on $\mathbf{S}^{tst}$.

According to the *MCC* and F_1, which are two overall measurements of the quality of the binary predictions, listed in Table 4, we can find that the TargetDBP acts as the best performer followed by iDNA-Prot [7], PseDNA-Pro [8], iDNAPro-PseAAC [9], iDNA-Prot|dis [10], Local-DPP [11], PSFM-DBT [12], HMMBinder [13], IKP-DBPPred [3], iDNAProt-ES [14], and DPP-PseAAC [15]. Compared with the second-best predictor, i.e., iDNAProt-ES (trained on PDB1075), TargetDBP achieves the improvements of 50.01, 7.08, 17.57, and 12.90 percent on *Spe*, *Acc*, *Pre*, and *MCC*, respectively, while still possesses almost the equal performance on F_1 as iDNAProt-ES at the same time.

It has not escaped from our notice that HMMBinder [13] and iDNAProt-ES (trained on $\mathbf{S}^{tr}$) [14] achieves the highest and second-highest *Rec* values, i.e., 99.32 and 95.95, respectively. However, the corresponding *Spe* values are much lower, i.e., 1.35 and 41.22, denoting too many false positives are incurred during prediction. On the other hand, Local-DPP achieves the best performance on *Spe* (93.92) while with much lower *Rec* value implying too many false negatives are produced during prediction. Those are why the *Pre* values of HMMBinder [13], iDNAProt-ES (trained on $\mathbf{S}^{tr}$) [14], and Local-DPP [11] are both lower than that of TargetDBP.

3.5 Online implementation of TargetDBP

We have implemented an online TargetDBP server, which is freely available for academic use at http://csbio.njust.edu.cn/bioinf/targetdbp/. To use the server, one or more protein sequences and an available email address should be inputted. Then, the server will evaluate the running time and send it to user by email. After the server finished the submitted prediction task, the result email will automatically be sent to user with instruction to access the result page, which will be kept on the TargetDBP web server for 3 months.

If only one given protein sequence is inputted, the TargetDBP prediction generally takes about 3-70 minutes depending on the length of the given sequence. The relatively long computational time stems from the fact that TargetDBP must perform PSI-BLAST, SANN, and the procedure of DBS prediction to gain discriminative features, and run LIBSVM to predict whether the given protein is DBP or not.

Although there is no limitation in the number of inputted protein sequences in the TargetDBP server, we strongly suggested that you should input less than 10 protein sequences once, since our computation resource is limited.

4 Conclusion

Accurate identification of DBPs is one of the most important tasks in the annotation of protein functions. In order to enhance the performance of DBP prediction, in this study, we have designed and implemented a new DBP predictor, named TargetDBP. Experimental results have demonstrated that the proposed TargetDBP significantly outperforms other existing DBP predictors. The superior performances of TargetDBP are due to several reasons, including an appropriate benchmark dataset, more discriminative feature design, and careful construction of the prediction model. For easy to use, the TargetDBP has already been implemented as a web server that is now available at http://csbio.njust.edu.cn/bioinf/targetdbp/.

To further improve the DBP prediction performance, our future work includes the following three directions: (1) extracting more discriminative features to represent the information buried in protein sequence; (2) employing the deep learning algorithm [61, 62] to obtain the available information extracted from the original feature representation; (3) using the semi-supervised or unsupervised machine-learning algorithms [63, 64] to learn the DBP prediction model on these proteins with unknown structures; (4) developing a more accurate method by combining TargetDBP and other state-of-the-art DBP prediction methods. Although the TargetDBP still has room for optimization (for instance by integrating more programs when available), we believe that it would be one of the most accurate tools for DBP prediction.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 61772273, 61373062, 61773346, and 61573317), the Fundamental Research Funds for the Central Universities (No. 30918011104). The authors would like to thank Dr. Swakkhar Shatabda and Dr. Rianon Zaman for providing the standalone program of HMMBinder. D.J. Yu and G.J. Zhang are the corresponding authors for this paper.

REFERENCES

[1] K. A. Jones, J. T. Kadonaga, P. J. Rosenfeld, T. J. Kelly, and R. Tjian, "A cellular DNA-binding protein that activates eukaryotic transcription and DNA replication," Cell, vol. 48, no. 1, pp. 79-89, 1987.

[2] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, "Protein function in the post-genomic era," Nature, vol. 405, no. 6788, pp. 823, 2000.

[3] K. Qu, K. Han, S. Wu, G. Wang, and L. Wei, "Identification of DNA-Binding Proteins Using Mixed Feature Representation Methods," Molecules, vol. 22, no. 10, pp. 1602, 2017.

[4] W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, and H. Zhang, "Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes," PloS one, vol. 9, no. 1, pp. e86703, 2014.

[5] M. Gao, and J. Skolnick, "DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions," Nucleic Acids Research, vol. 36, no. 12, pp. 3978-3992, 2008.

[6] G. Nimrod, M. Schushan, A. Szilágyi, C. Leslie, and N. Ben-Tal, "iDBPs: a web server for the identification of DNA binding proteins," Bioinformatics, vol. 26, no. 5, pp. 692-693, 2010.

[7] W.-Z. Lin, J.-A. Fang, X. Xiao, and K.-C. Chou, "iDNA-Prot: identification of DNA binding proteins using random forest with grey model," PloS one, vol. 6, no. 9, pp. e24756, 2011.

[8] B. Liu, J. H. Xu, S. X. Fan, R. F. Xu, J. Y. Zhou, and X. L. Wang, "PseDNA-Pro: DNA-Binding Protein Identification by Combining Chou's PseAAC and Physicochemical Distance Transformation," Molecular Informatics, vol. 34, no. 1, pp. 8-17, Jan, 2015.

[9] B. Liu, S. Wang, and X. Wang, "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation," Scientific Reports, vol. 5, pp. 15479, 2015.

[10] B. Liu, J. Xu, X. Lan, R. Xu, J. Zhou, X. Wang, and K.-C. Chou, "iDNA-Prot|dis: 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, pp. e106691, 2014.

[11] L. Wei, J. Tang, and Q. Zou, "Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information," Information Sciences, vol. 384, pp. 135-144, 2017.

[12] J. Zhang, and B. Liu, "PSFM-DBT: Identifying DNA-binding proteins by combing position specific frequency matrix and distance-bigram transformation," International Journal of Molecular Sciences, vol. 18, no. 9, pp. 1856, 2017.

[13] R. Zaman, S. Y. Chowdhury, M. A. Rashid, A. Sharma, A. Dehzangi, and S. Shatabda, "HMMBinder: DNA-Binding Protein Prediction Using HMM Profile Based Features," BioMed Research International, vol. 2017, 2017.

[14] S. Y. Chowdhury, S. Shatabda, and A. Dehzangi, "iDNAProt-ES: Identification of DNA-binding proteins using evolutionary and structural features," Scientific Reports, vol. 7, no. 1, pp. 14938, 2017.

[15] M. S. Rahman, S. Shatabda, S. Saha, M. Kaykobad, and M. S. Rahman, "DPP-PseAAC: A DNA-binding protein prediction model using Chou's general PseAAC," Journal Of Theoretical Biology, vol. 452, pp. 22-34, Sep 7, 2018.

[16] R. F. Xu, J. Y. Zhou, B. Liu, Y. L. He, Q. Zou, X. L. Wang, and K. C. Chou, "Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach," Journal Of Biomolecular Structure & Dynamics, vol. 33, no. 8, pp. 1720-1730, Aug 3, 2015.

[17] X. W. Zhao, X. T. Li, Z. Q. Ma, and M. H. Yin, "Identify DNA-Binding Proteins with Optimal Chou's Amino Acid Composition," Protein And Peptide Letters, vol. 19, no. 4, pp. 398-405, Apr, 2012.

[18] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features," Amino Acids, vol. 34, no. 1, pp. 103-109, Jan, 2008.

[19] D. Julong, "Introduction to grey system theory," The Journal of grey system, vol. 1, no. 1, pp. 1-24, 1989.

[20] K. C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," Journal Of Theoretical Biology, vol. 273, no. 1, pp. 236-247, Mar 21, 2011.

[21] K. C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," Proteins-Structure Function And Genetics, vol. 43, no. 3, pp. 246-255, May 15, 2001.

[22] B. Liu, D. Zhang, R. Xu, J. Xu, X. Wang, Q. Chen, Q. Dong, and K.-C. Chou, "Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection," Bioinformatics, vol. 30, no. 4, pp. 472-479, 2013.

[23] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment," Nature Methods, vol. 9, no. 2, pp. 173, 2012.

[24] B. Liu, F. Yang, D. S. Huang, and K. C. Chou, "iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC," Bioinformatics, vol. 34, no. 1, pp. 33-40, Jan 1, 2018.

[25] X. Cheng, S. G. Zhao, X. Xiao, and K. C. Chou, "iATC-mISF: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals," Bioinformatics, vol. 33, no. 3, pp. 341-346, Feb 1, 2017.

[26] X. Cheng, X. Xiao, and K. C. Chou, "pLoc_bal-mGneg: Predict subcellular localization of Gram-negative bacterial proteins by quasi-balancing training dataset and general PseAAC," Journal Of Theoretical Biology, vol. 458, pp. 92-102, Dec 7, 2018.

[27] P. W. Rose, A. Prlić, C. Bi, W. F. Bluhm, C. H. Christie, S. Dutta, R. K. Green, D. S. Goodsell, J. D. Westbrook, and J. Woo, "The RCSB Protein Data Bank: views of structural biology for basic and applied research and education," Nucleic Acids Research, vol. 43, no. D1, pp. D345-D356, 2015.

[28] R. Storn, and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, no. 4, pp. 341-359, 1997.

[29] W. Li, and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," Bioinformatics, vol. 22, no. 13, pp. 1658-1659, 2006.

[30] Author ed.^eds., "Statistical Learning Theory ", New York: Wiley-Interscience, 1998, p.^pp. Pages.

[31] A. Liaw, and M. Wiener, "Classification and Regression by random Forest," R news, vol. 2, no. 3, pp. 18-22, 2002.

[32] K. C. Chou, "Impacts of Bioinformatics to Medicinal Chemistry," Medicinal Chemistry, vol. 11, no. 3, pp. 218-234, 2015.

[33] J. Hu, K. Han, Y. Li, J.-Y. Yang, H.-B. Shen, and D.-J. Yu, "TargetCrys: protein crystallization prediction by fusing multi-view features with two-layered SVM," Amino acids, vol. 48, no. 11, pp. 2533-2547, 2016.

[34] K. Chen, L. Kurgan, and M. Rahbari, "Prediction of protein crystallization using collocation of amino acid pairs," Biochemical And Biophysical Research Communications, vol. 355, no. 3, pp. 764-769, Apr 13, 2007.

[35] L. Kurgan, A. A. Razib, S. Aghakhani, S. Dick, M. Mizianty, and S. Jahandideh, "CRYSTALP2: sequence-based protein crystallization propensity prediction," Bmc Structural Biology, vol. 9, Jul 31, 2009.

[36] K. C. Chou, "Pseudo Amino Acid Composition and its Applications in Bioinformatics, Proteomics and System Biology," Current Proteomics,

vol. 6, no. 4, pp. 262-274, Dec, 2009.

[37] K.-C. Chou, and H.-B. Shen, "MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM," Biochemical and biophysical research communications, vol. 360, no. 2, pp. 339-345, 2007.

[38] B. Liu, F. L. Liu, X. L. Wang, J. J. Chen, L. Y. Fang, and K. C. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," Nucleic Acids Research, vol. 43, no. W1, pp. W65-W71, Jul 1, 2015.

[39] B. Liu, "BioSeq-Analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches," Briefings in bioinformatics, 2017.

[40] D. J. Yu, J. Hu, J. Yang, H. B. Shen, J. H. Tang, and J. Y. Yang, "Designing Template-Free Predictor for Targeting Protein-Ligand Binding Sites with Classifier Ensemble and Spatial Clustering," IEEE-ACM Transactions on Computational Biology and Bioinformatics, vol. 10, no. 4, pp. 994-1008, Jul-Aug, 2013.

[41] J. Yang, A. Roy, and Y. Zhang, "Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment," Bioinformatics, vol. 29, no. 20, pp. 2588-2595, 2013.

[42] B. He, S. Mortuza, Y. Wang, H.-B. Shen, and Y. Zhang, "NeBcon: protein contact map prediction using neural network training coupled with naïve Bayes classifiers," Bioinformatics, vol. 33, no. 15, pp. 2296-2306, 2017.

[43] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, pp. 1883, 2009.

[44] D.-J. Yu, J. Hu, X.-W. Wu, H.-B. Shen, J. Chen, Z.-M. Tang, J. Yang, and J.-Y. Yang, "Learning protein multi-view features in complex space," Amino acids, vol. 44, no. 5, pp. 1365-1379, 2013.

[45] A. A. Schaffer, L. Aravind, T. L. Madden, S. Shavirin, J. L. Spouge, Y. I. Wolf, E. V. Koonin, and S. F. Altschul, "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements," Nucleic Acids Research, vol. 29, no. 14, pp. 2994-3005, Jul 15, 2001.

[46] B. Lee, and F. M. Richards, "The interpretation of protein structures: estimation of static accessibility," Journal of Molecular Biology, vol. 55, no. 3, pp. 379–400, IN3–IN4, 1971.

[47] N. Bhardwaj, and H. Lu, "Residue-level prediction of DNA-binding sites and its application on DNA-binding protein predictions," FEBS letters, vol. 581, no. 5, pp. 1058-1066, 2007.

[48] K. Joo, S. J. Lee, and J. Lee, "Sann: solvent accessibility prediction of proteins by nearest neighbor method," Proteins-structure Function & Bioinformatics, vol. 80, no. 7, pp. 1791–1797, 2012.

[49] J. Hu, Y. Li, M. Zhang, X. Yang, H.-B. Shen, and D.-J. Yu, "Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs," IEEE/ACM transactions on computational biology and bioinformatics, vol. 14, no. 6, pp. 1389-1398, 2017.

[50] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," IEEE Intelligent Systems and their applications, vol. 13, no. 4, pp. 18-28, 1998.

[51] X.-G. Zhou, and G.-J. Zhang, "Abstract convex underestimation assisted multistage differential evolution," IEEE transactions on cybernetics, vol. 47, no. 9, pp. 2730-2741, 2017.

[52] F. Neri, and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," Artificial Intelligence Review, vol. 33, no. 1-2, pp. 61-106, 2010.

[53] X. G. Zhou, and G. J. Zhang, "Differential Evolution With Underestimation-Based Multimutation Strategy," IEEE transactions on cybernetics, vol. PP, no. 99, pp. 1-12, 2018.

[54] G.-J. Zhang, X.-G. Zhou, X.-F. Yu, X.-H. Hao, and L. Yu, "Enhancing protein conformational space sampling using distance profile-guided differential evolution," IEEE/ACM transactions on computational biology and bioinformatics, vol. 14, no. 6, pp. 1288-1301, 2017.

[55] D. Qiao, and G. K. Pang, "A modified differential evolution with heuristic algorithm for nonconvex optimization on sensor network localization," IEEE Transactions on Vehicular Technology, vol. 65, no. 3, pp. 1676-1689, 2016.

[56] K. Yan, and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," Sensors and Actuators B: Chemical, vol. 212, pp. 353-363, 2015.

[57] Y. Wang, Y. Ding, F. Guo, L. Wei, and J. Tang, "Improved detection of DNA-binding proteins via compression technology on PSSM information," PloS one, vol. 12, no. 9, pp. e0185587, 2017.

[58] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine learning, vol. 46, no. 1-3, pp. 389-422, 2002.

[59] C.-C. Chang, and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, pp. 27, 2011.

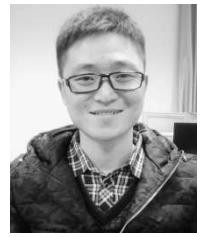
[60] W. Chen, P. M. Feng, H. Lin, and K. C. Chou, "iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition," Nucleic Acids Research, vol. 41, no. 6, Apr, 2013.

[61] M. Spencer, J. Eickholt, and J. Cheng, "A deep learning network approach to ab initio protein secondary structure prediction," IEEE/ACM transactions on computational biology and bioinformatics, vol. 12, no. 1, pp. 103-112, 2015.

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[63] B. B. Jiang, H. H. Chen, B. Yuan, and X. Yao, "Scalable Graph-Based Semi-Supervised Learning through Sparse Bayesian Model," Ieee Transactions on Knowledge And Data Engineering, vol. 29, no. 12, pp. 2758-2771, Dec, 2017.

[64] C. C. Chen, H. H. Juan, M. Y. Tsai, and H. H. S. Lu, "Unsupervised Learning and Pattern Recognition of Biological Data Structures with Density Functional Theory and Machine Learning," Scientific Reports, vol. 8, Jan 11, 2018.



Jun Hu received his B.S. degree in computer science from Anhui Normal University in 2011. From 2011 to 2018, he acted as a Ph.D Student of School of Computer Science and Engineering in Nanjing University of Science and Technology and a member of Pattern Recognition and Bioinformatics Group, led by professor Dong-Jun Yu. From 2016 to 2017, he acted as a visiting student at the University of Michigan (Ann Arbor) in U.S.A. He is currently a teacher in the College of Information Engineering at Zhejiang University of Technology. His current interests include pattern recognition, data mining and bioinformatics.

Xiao-Gen Zhou received the Ph.D degree in control science and engineering from College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2018. His research interests include intelligent information processing, optimization theory and algorithm design, and bioinformatics.

Yi-Heng Zhu received his B.S. degree in computer science from Nanjing Institute of Technology, China in 2015. Currently, he is a Ph.D Student of School of Computer Science and Engineering in NanJing University of Science and Technology and a member of Pattern Recognition and Bioinformatics Group, led by professor Dong-Jun Yu. His research interests include bioinformatics, data mining, and pattern recognition.

Dong-Jun Yu received the PhD degree from Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2003. In 2008, he acted as an academic visitor at Department of Computer of the University of York in the UK. He also visited the Department of Computational Medicine of the University of Michigan (Ann Arbor) in 2016. He is currently a full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition, machine learning and bioinformatics. He is the author of more than 50 scientific papers in pattern recognition and bioinformatics. He is a senior member of China Computer Federation (CCF) and a senior member of China Association of Artificial Intelligence (CAAI).

Gui-Jun Zhang received the Ph.D. degree in control theory and control engineering from Shanghai Jiaotong University, Shanghai, China, in 2004. He is currently a Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His current research interests include intelligent information processing, optimization theory and algorithm design, and bioinformatics.