JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

Cite This: J. Chem. Inf. Model. 2019, 59, 3057−3071

pubs.acs.org/jcim

DNAPred: Accurate Identification of DNA-Binding Sites from Protein Sequence by Ensembled Hyperplane-Distance-Based Support Vector Machines

Yi-Heng Zhu,[†] Jun Hu,[‡] Xiao-Ning Song,[§] and Dong-Jun Yu*[,†]

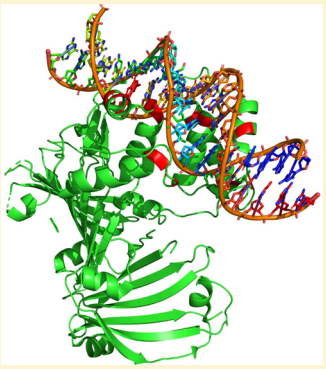
[†]School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing 210094, P. R. China

[‡]College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, P. R. China

[§]School of Internet of Things, Jiangnan University, 1800 Lihu Road, Wuxi 214122, P. R. China

Supporting Information

ABSTRACT: Accurate identification of protein−DNA binding sites is significant for both understanding protein function and drug design. Machine-learning-based methods have been extensively used for the prediction of protein−DNA binding sites. However, the data imbalance problem, in which the number of nonbinding residues (negative-class samples) is far larger than that of binding residues (positive-class samples), seriously restricts the performance improvements of machine-learning-based predictors. In this work, we designed a two-stage imbalanced learning algorithm, called ensembled hyperplane-distance-based support vector machines (E-HDSVM), to improve the prediction performance of protein−DNA binding sites. The first stage of E-HDSVM designs a new iterative sampling algorithm, called hyperplane-distance-based under-sampling (HD-US), to extract multiple subsets from the original imbalanced data set, each of which is used to train a support vector machine (SVM). Unlike traditional sampling algorithms, HD-US selects samples by calculating the distances between the samples and the separating hyperplane of the SVM. The second stage of E-HDSVM proposes an enhanced AdaBoost (EAdaBoost) algorithm to ensemble multiple trained SVMs. As an enhanced version of the original AdaBoost algorithm, EAdaBoost overcomes the overfitting problem. Stringent cross-validation and independent tests on benchmark data sets demonstrated the superiority of E-HDSVM over several popular imbalanced learning algorithms. Based on the proposed E-HDSVM algorithm, we further implemented a sequence-based protein−DNA binding site predictor, called DNAPred, which is freely available at http://csbio.njust.edu.cn/bioinf/dnapred/ for academic use. The computational experimental results showed that our predictor achieved an average overall accuracy of 91.7% and a Mathew's correlation coefficient of 0.395 on five benchmark data sets and outperformed several state-of-the-art sequence-based protein−DNA binding site predictors.



INTRODUCTION

In living cells, protein−DNA interactions participate in many essential biological processes, including DNA replication, repair, and modifications.[1−3] The accurate identification of protein−DNA binding residues not only contributes to the characterization of protein function but also has important practical significance for drug design.[4] Traditionally, researchers distinguish the DNA-binding sites through biochemical methods, such as fast ChIP[5] and X-ray crystallography.[6] However, these methods are time consuming and laborious and cannot keep pace with related research progress in the postgenome era, when a large amount of unannotated protein−DNA complexes are rapidly sequenced and deposited. In view of this, many intelligent computational approaches, such as template-based methods and machine-learning-based methods, have been developed for the effective prediction of protein−DNA binding residues during the past decades.

Template-based methods lead the trend in the field of protein−DNA interaction prediction at the early stage.[7−9] These methods identify the DNA-binding sites from a query protein by using the sequence and/or structure information of templates, which are selected by using the mature alignment or comparison algorithms. For example, Morozov et al.[10] identified DNA-binding affinities by applying two models: one was based on a physical energy function and the other based on the knowledge of the consensus sequence and the number of interactions between DNA bases and amino acids. Gao et al.[11] developed DBD-Hunter, which combined structural comparison with statistical potential between residues and DNA base pairs, to identify the protein−DNA interactions. There exist other elegant predictors, including PreDs,[12] DBD-Threader,[13] DR_bind,[14] and PreDNA.[15] Template-based methods can achieve a satisfactory performance for predicting DNA-binding residues in the situation where high-quality protein sequence and/or structure

Received: October 25, 2018
Published: April 3, 2019

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

templates are available. Nevertheless, many proteins have no available high-quality templates in the real world, which seriously restricts the prediction performances of the template-based methods.

In recent years, machine-learning-based methods have attracted more and more attention.[16−18] For example, Ofran et al.[19] presented DISIS-DNA, which incorporated sequence features and machine-learning algorithms, including support vector machine (SVM) and neural network (NN), to predict DNA-binding sites in proteins. Hwang et al.[20] implemented a web server, DP-Bind, which utilized three machine-learning models, i.e., kernel logistic regression (KLR), SVM, and penalized logistic regression (PLR), to identify the protein−DNA interactions. Wu et al.[21] implemented a random forest (RF)-based model with a novel hybrid feature, which was composed of sequence evolutionary profiles, predicted secondary structure, and orthogonal binary vector, for distinguishing protein−DNA binding residues. Yan et al.[22] designed a two-layered predictor (DRNApred), which utilized a hidden Markov model (HMM) and logistic regression (LR), for the identification of protein−DNA interactions.

Although great progress has been made by machine-learning-based methods for the prediction of protein−DNA interactions, challenges remain. An inevitable critical problem is the data imbalance phenomenon, where the number of negative-class samples (nonbinding sites) is significantly larger than that of positive-class samples (binding sites). Numerous reports[23−25] have shown that the prediction models, which are trained on imbalanced data through traditional machine-learning algorithms (such as SVM, RF, NN, etc.), tend to be biased to the negative classes (e.g., nonbinding residues being assigned to the negative class). Taking SVM as an example, which is one of the mostly used machine-learning algorithms in this field, it usually shows robust performance on balanced data sets but fails to obtain satisfactory results on imbalanced data sets. The underlying reason can be explained as follows. SVM is characterized by "support-vectors-dependency"; i.e., the performance of the SVM is only determined by the separating hyperplane between the positive and negative support vectors. In the special scenario where the SVM is trained on imbalanced data sets, the corresponding hyperplane may be pushed toward the positive class; as a result, the SVM tends to predict positive samples as negative ones.

Many imbalanced-learning methods[26−28] have emerged in recent years to enable effective learning from imbalanced data. Among various methods, random under-sampling is a basic technique and has been widely used for the prediction of protein−ligand binding sites.[29,30] However, random under-sampling cannot always achieve excellent performance. The underlying reason can be explained as follows. Random under-sampling removes numerous negative samples from the original imbalanced data set to form a balanced data set, thereby easily leading to information loss of important samples. To overcome the defects of random under-sampling, several improved under-sampling methods have been proposed,[31,32] such as cluster-based under-sampling[33] and evolutionary under-sampling.[34] Recently, Tang el al.[35,36] innovatively proposed a granular SVMs-repetitive under-sampling (GSVM-RU) algorithm, which iteratively combines the under-sampling procedure and the prediction model training procedure, to effectively deal with the class imbalance problem. Intrinsically, GSVM-RU is based on the granule computing theory[37,38] and the "support-vectors-dependency" of SVM, which is further described in the section "Procedures and Potential Defects of GSVM-RU". Previous studies[35,36] have found that GSVM-RU can elegantly deal with the data imbalance problem and achieve better performance than random under-sampling in most cases. However, we notice there is still room for further enhancing the performance of GSVM-RU by addressing its two potential defects, i.e., "ideal-hyperplane missing" and "information loss or redundancy" (refer to the section "Procedures and Potential Defects of GSVM-RU" for details).

To overcome the defects of GSVM-RU, in this work, we proposed an ensembled hyperplane-distance-based support vector machines (E-HDSVM), which is an improved version of GSVM-RU. There are two stages in E-HDSVM: hyperplane-distance-based support vector machines (HDSVMs) generation and HDSVMs ensemble. In the first stage, a hyperplane-distance-based under-sampling (HD-US) algorithm is utilized to generate multiple training subsets, each of which is subsequently used for training an individual SVM, called HDSVM; in the ensemble stage, multiple trained HDSVMs are ensembled as the final prediction model by applying an enhanced AdaBoost (EAdaBoost) algorithm.

We demonstrated the efficacy of the proposed E-HDSVM for imbalanced learning by stringently comparing it with the random under-sampling and GSVM-RU. The computational experimental results on five protein−DNA binding site data sets have shown that our predictor outperforms several other state-of-the-art sequence-based protein−DNA binding site predictors. Based on the proposed E-HDSVM, we further implemented a sequence-based protein−DNA binding site predictor, called DNAPred, which is freely available at http://csbio.njust.edu.cn/bioinf/dnapred/ for academic use.

MATERIALS AND METHODS

Benchmark Data Sets. In this study, five DNA−protein binding site data sets, including PDNA-543, PDNA-41, PDNA-335, PDNA-52, and PDNA-316, were used to evaluate the proposed methods.

PDNA-543 and PDNA-41 were constructed in our previous work (Hu et al.[39]). First, we collected 7186 DNA-binding protein chains, which were annotated in the Protein Data Bank (PDB)[40] before October 10, 2015, to form an original data set. After applying the CD-HIT software[41] to remove the redundant sequences, we obtained a nonredundant data set in which no two sequences had more than 30% identity. Finally, the nonredundant data set was divided into two subsets: the training data set (PDNA-543) and the independent test data set (PDNA-41). PDNA-543 included 543 protein sequences, which were released into the PDB before October 10, 2014; PDNA-41 contained 41 protein sequences, which were released into the PDB after October 10, 2014.

PDNA-335 and PDNA-52 were also employed in our previous work (Yu et al.[42]). PDNA-335 consisted of 335 protein sequences, released into PDB before 10 March 2010, from BioLip.[43] The maximal pairwise sequence identity of proteins in PDNA-335 was reduced to less than 40% with the PISCES software.[44] PDNA-52 was composed of 52 protein sequences, released into PDB after 10 March 2010, from BioLip. Again, the maximal pairwise sequence identity of proteins in PDNA-52 was culled to 40% by using PISCES. In addition, no sequences in PDNA-335 had more than 40% pairwise identity to the sequences in PDNA-52.

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

PDNA-316 was constructed by Si et al.[17] and consisted of 316 DNA-binding protein chains. The details of PDNA-316 can be found in Si et al.[17] The detailed statistical summary of PDNA-543, PDNA-41, PDNA-335, PDNA-52, and PDNA-316 is presented in Table 1.

Table 1. Statistical Summary of PDNA-543, PDNA-41, PDNA-335, PDNA-52, and PDNA-316

Data Set	No. of Sequences	Num_P[a]	Num_N[b]	Ratio[c]
PDNA-543	543	9549	134,995	14
PDNA-41	41	734	14,021	19
PDNA-335	335	6461	71,320	11
PDNA-52	52	973	16,225	17
PDNA-316	316	5609	67,109	12

[a]Num_P is the number of positive samples. [b]Num_N is the number of negative samples. [c]Ratio = Num_N/Num_P, which measures the imbalance degree of the data set.

Feature Representation. In this work, four features, i.e., position-specific scoring matrix (PSSM), predicted secondary structure (PSS), predicted relative solvent accessibility (PRSA), and amino acid frequency difference between binding and nonbinding (AAFD-BN), are serially combined to form the feature representation of each residue in a protein sequence.

Position-Specific Scoring Matrix. The position-specific scoring matrix (PSSM) encodes the evolutionary conservation of protein sequences.[45] For a protein sequence with L residues, we used the PSI-BLAST software[46] to search against the SWISS-PROT database[47] via three iterations with 0.001 as the E-value cutoff to generate the corresponding original PSSM, with L rows and 20 columns. Then, a standard logistic function was utilized to normalize the original PSSM:

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (1)$$

where x is an element in the original PSSM.

Predicted Secondary Structure. For a protein sequence with L residues, we used the PSIPRED software[48] to extract its predicted secondary structure (PSS) matrix with L rows and three columns; the three values of the ith row in the matrix are the respective probabilities that the ith residue belongs to three secondary structure classes: coil (C), helix (H), and strand (E).

Predicted Relative Solvent Accessibility. We extracted predicted relative solvent accessibility (PRSA) of each residue as follows. For a protein sequence with L residues, the SANN program[49] was applied to generate an $L \times 3$ PRSA matrix; the three values of the ith row in the PRSA matrix indicate the respective probabilities that the ith residue belongs to three solvent accessibility classes: buried (B), intermediate (I), and exposed (E).

Amino Acid Frequency Difference between Binding and Nonbinding. Previous researches[50,51] have shown that different ligands tend to interact with different types of residues. Inspired by this, we considered that the frequencies of the amino acids among binding or nonbinding residues may help to improve the accuracy in ligand-binding site prediction, so we designed a new sequence-based feature, called amino acid frequency difference between binding and nonbinding (AAFD-BN). Concretely, in a protein data set, the frequencies of the 20 native amino acids (numbered from 1 to 20) among DNA-binding and nonbinding residues can be separately represented as two different vectors with 20-dimensionality, denoted as F_1 and F_2. Then, a frequency difference vector, represented as F, was obtained, where $F = F_1 - F_2 = \{f_1, f_2, \cdots, f_{20}\}$, and f_i was the frequency difference of the ith type of amino acid between DNA-binding and nonbinding residues. Hence, for a protein sequence with L residues, an $L \times 1$ AAFD-BN matrix was extracted; the value of the ith row in matrix is f_{C_i}, where $C_i \in \{1,2,3,..., 20\}$ is the amino acid type of the ith residue.

After obtaining the above four feature matrices, we adopted the sliding window technique[52] to represent the feature vector of each residue. Our previous work[39] has shown that the best window size is 9. Thus, the dimensionalities of the PSSM, PSS, PRSA, and AAFD-BN feature vectors of a residue are 180, 27, 27, and 9, respectively. The final feature of a residue is a 243-dimensionality vector generated by serially combining the above four vectors.

Procedures and Potential Defects of GSVM-RU. *Procedures of GSVM-RU.* Intrinsically, GSVM-RU is an iterative algorithm where each iteration has two stages, (A) granule extraction and (B) granule fusion, for dealing with class imbalanced learning problem. It should be noted that a granule here refers to a subset of the original training data set. Granule extraction aims to extract a granule by a support-vectors-based under-sampling method (SV-US), while the second stage focuses on integrating the extracted multiple granules to form a fused data set. Taking the ith iteration as an example, the details of the two stages of GSVM-RU are as follows.

(A) Granule Extraction. A negative granule, denoted as NG_i, is extracted by the SV-US as follows. First, an original SVM, denoted as $ORSVM_i$, is learned on the training data set (TD_i); then all negative support vectors (SVs) of $ORSVM_i$ are extracted as NG_i. Finally, the NG_i is removed from TD_i, and the remaining set, denoted as TD_{i+1}, is used as the training data set in the next iteration; i.e., $TD_{i+1} \leftarrow TD_i - NG_i$. Note that TD_i is the original training data set (OTD) in the first iteration; i.e., $TD_1 \leftarrow OTD$.

(B) Granule Fusion. Let PG be the positive granule which consists of all positive samples in OTD. After NG_i is extracted, it and PG are appropriately integrated to generate a fused data set, denoted as FD_i, at the ith iteration by using one of the two data-level fusion strategies (i.e., "Discard" and "Combine"). More specifically, the "Discard" strategy generates FD_i by combining the positive granule (PG) with NG_i (i.e., $FD_i \leftarrow PG \cup NG_i$), while the "Combine" strategy generates FD_i by combining PG with all extracted NGs (i.e., $FD_i \leftarrow PG \cup NG_1 \cup NG_2 \cup \ldots \cup NG_i$). Finally, a granular SVM, called $GSVM_i$, is trained on the obtained FD_i.

This iteration will be terminated when the performance of $GSVM_{i+1}$ is inferior to that of $GSVM_i$. $GSVM_i$ is then selected as the final prediction model. Accordingly, GSVM-RU has two versions according to the data-level fusion strategies used: One is a "Discard"-based version, denoted as GSVM-RD; the other is a "Combine"-based version, denoted as GSVM-RC.

Potential Defects of GSVM-RU. Although GSVM-RU has been demonstrated to be useful for imbalanced learning,[35,36] it does have two potential defects deserved to be further investigated. First, SV-US may potentially cause the so-called ideal-hyperplane-missing (IHM) phenomenon, which is adverse to the subsequent granule fusion. Concretely, in the first several iterations, the sample number of removed NGs is

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

too large (refer to the section "Performance Comparisons between HD-US and SV-US"), which easily leads to the emergence of IHM, resulting in the performance deterioration of the corresponding original SVMs (ORSVMs). A detailed description of the IHM phenomenon and its negative effect is further discussed in the section "Performance Comparisons between HD-US and SV-US". Due to the inferior performances of ORSVMs, the GSVMs also show deteriorated performances in the "Discard"-based granule fusion stage (Note that when the "Discard" method is adopted, ORSVM and GSVM have nearly the same performances in each iteration due to the "support-vectors-dependency" of SVM).

Second, the data-level fusion methods may cause information loss or redundancy. Concretely, for "Discard"-based granule fusion, in the ith iteration, GSVM-RU only considers NG_i and discards the important information buried in other NGs (i.e., NG_1, NG_2,⋯, NG_{i-1}), which easily leads to information loss. In the "Combine"-based granule fusion, all NGs, which may contain noise and overlapping information, are blindly incorporated. As a result, information redundancy is inevitable.

Ensembled Hyperplane-Distance-Based Support Vector Machines. To overcome the defects of GSVM-RU, we propose an enhanced version of GSVM-RU, denoted as E-HDSVM. Compared with GSVM-RU, E-HDSVM has two advantages: First, a novel hyperplane-distance-based under-sampling (HD-US) method, rather than SV-US, is designed to eliminate the IHM phenomenon; moreover, to relieve the negative impact of information loss and/or redundancy, a decision-level fusion strategy is adopted to replace the original data-level fusion strategies. More specifically, we try to fuse multiple SVMs trained on multiple granules rather than directly fusing those multiple granules. In this section, we describe the architecture of E-HDSVM, which includes HDSVMs generation and a HDSVMs ensemble procedure.

HDSVMs Generation. In the first stage, multiple subsets are orderly generated from the original training data set (OTD) by the proposed HD-US, and the corresponding SVM, called HDSVM, is constructed on each subset. HD-US is an iterative under-sampling algorithm that can generate a new subset in each iteration. Taking the ith iteration as an example, the process of HD-US is performed as follows. A SVM, denoted as $HDSVM_i$, is first constructed on the training data set S_i. Then a few negative samples, which have the shortest distances to the hyperplane of trained $HDSVM_i$, are removed from the S_i, and the remaining samples form a new subset of OTD, which is denoted as S_{i+1} and used as the training data set in the $(i+1)$ iteration. It should be noted that S_i is OTD in the first iteration; i.e., $S_1 \leftarrow OTD$.

To implement the first stage, the following two problems should be further investigated: (1) How many times do we need to repeat the iterations? (2) How many negative samples should be removed in each iteration?

The above-mentioned problems could be solved as follows. First, in this work, we only consider a special situation where the numbers of removed negative samples in each iteration are equal. To facilitate the description, the number of iterations and the number of removed negative samples in each iteration are represented as I and K, respectively. Obviously, the number of generated HDSVMs is also equal to I. Increasing the number of HDSVMs may help to improve the performance of ensembled HDSVM. Thus, to obtain as many HDSVMs as possible, we execute the iteration procedure until the newly generated subset does not contain negative samples. Under the above conditions, I is inversely proportional to K:

$$I = M_N/K \tag{2}$$

where M_N is the total number of negative samples in OTD. At the extreme, if we only remove one negative sample in each iteration ($K = 1$), we will obtain M_N HDSVMs. Unfortunately, due to the tremendous time consumption, this strategy is impractical. Thus, selecting the value of K is a tradeoff between the time consumption and the performance improvement. Moreover, for the data sets with different scales, it is difficult to preset a fixed K value but easy to choose a constant I. Therefore, unless otherwise stated, we set $I = 10$ in all experiments of E-HDSVM (the reason is carefully explained in the section "Choosing the value of I in E-HDSVM"). The procedures of the first stage (i.e., HDSVMs generation) in E-HDSVM are described in Algorithm 1 as follows.

Algorithm 1. HDSVMs Generation:

Input: OTD − original training data set; I − number of iterations

Output: $HDSVMSet$ − set of trained HDSVMs

Initialization: $i \leftarrow 1$; $S_i \leftarrow OTD$; $HDSVMSet \leftarrow \phi$; $K = M_N/I$

Step 1. Train a SVM, called $HDSVM_i$, on S_i, and add it to $HDSVMSet$:

$$HDSVM_i \leftarrow Train(S_i) \tag{3}$$

$$HDSVMSet \leftarrow HDSVMSet \cup \{HDSVM_i\} \tag{4}$$

Step 2. Let DS_i be an empty set, where DS_i is the distance set in the ith iteration. Let H_i be the separating hyperplane of $HDSVM_i$. For each negative sample x_j in S_i, calculate its distance to H_i, denoted as $d_{i,j}$, and then add $d_{i,j}$ to DS_i:

$$d_{i,j} \leftarrow \text{Caculate\_}Distance(H_i, x_j) \tag{5}$$

$$DS_i \leftarrow DS_i \cup \{d_{i,j}\} \tag{6}$$

Step 3. Based on the distance set DS_i, the K negative samples, which have the shortest distances to H_i, are removed from S_i, and the remaining samples form a new subset S_{i+1}, which is used as the training data set in the next iteration:

$$S_{i+1} \leftarrow \text{Remove}(S_i, DS_i, K) \tag{7}$$

Step 4. $i \leftarrow i + 1$; if $i > I$, Return $HDSVMSet = \{HDSVM_1, HDSVM_2, \cdots, HDSVM_I\}$; otherwise, go to Step 1.

HDSVMs Ensemble. In the second stage, all elements in $HDSVMSet$ are ensembled to obtain the final prediction model by an enhanced AdaBoost (EAdaBoost) algorithm. As an enhanced version of the original AdaBoost,[53] the proposed EAdaBoost overcomes the overfitting problem. Before introducing EAdaBoost, we review the details of the original AdaBoost as follows. The original AdaBoost first calculates the weight of each base classifier and then combines all base classifiers with their weights to generate an ensembled classifier. However, the original AdaBoost easily leads to the overfitting problem, especially when dealing with prediction tasks related to protein residues.[42] Specifically, in the original AdaBoost, samples in the entire training data set are used as evaluation samples to calculate the weights of base classifiers. In other words, the evaluation samples and the training samples originate from the same data set. As a result, the ensembled classifier has an excellent performance on the training data set and then would be with poor generalization performances on other test data sets.

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

Table 2. *AUC* Performances of $HDSVM_i$ and $ORSVM_i$ with Respect to Different Values of i on PDNA-543, PDNA-335, and PDNA-316 over 10-Fold Cross-Validation

i	1	2	3	4	5	6	7	8	9	10	11	12	Avg[g]
HPDNA-543[a]	0.816	0.829	0.842	0.849	**0.850**	0.847	0.830	0.803	0.761	0.701	–	–	0.813
OPDNA-543[b]	0.816	0.844	**0.846**	0.840	0.824	0.803	0.780	0.752	0.726	0.698	0.669	0.646	0.770
HPDNA-335[c]	0.826	0.838	0.847	**0.851**	0.850	0.847	0.840	0.811	0.765	0.709	–	–	0.818
OPDNA-335[d]	0.826	**0.848**	0.846	0.834	0.810	0.777	0.745	0.714	0.688	–	–	–	0.788
HPDNA-316[e]	0.847	0.859	0.869	**0.872**	0.871	0.867	0.855	0.826	0.780	0.725	–	–	0.837
OPDNA-316[f]	0.847	**0.871**	0.869	0.857	0.839	0.811	0.780	0.756	0.731	–	–	–	0.818

[a]HPDNA-543 is the *AUC* value of $HDSVM_i$ on PDNA-543. [b]OPDNA-543 is the *AUC* value of $ORSVM_i$ on PDNA-543. [c]HPDNA-335 is the *AUC* value of $HDSVM_i$ on PDNA-335. [d]OPDNA-335 is the *AUC* value of $ORSVM_i$ on PDNA-335. [e]HPDNA-316 is the *AUC* value of $HDSVM_i$ on PDNA-316. [f]OPDNA-316 is the *AUC* value of $ORSVM_i$ on PDNA-316. [g]Avg is the averaged *AUC* value of all HDSVMs/ORSVMs on a given data set; '–' indicates that the corresponding value does not exist.

In light of this, we modify the original AdaBoost as follows. An independent evaluation data set (*IED*), which does not share samples with the training data set, is used to perform the boosting procedure, and we rename the modified AdaBoost as enhanced AdaBoost (EAdaBoost). In addition, we construct *IED* in this work as follows. Given a training data set (*TRD*) for E-HDSVM, we randomly select 20% of samples from *TRD* to form *IED*, and the rest of the samples are used as a training data set (i.e., *OTD*) in the HDSVMs generation stage. Based on EAdaBoost, we perform HDSVMs ensemble procedures as described in Algorithm 2.

Algorithm 2. HDSVMs Ensemble:

Input: *HDSVMSet* − set of HDSVMs (*HDSVMSet*) that are trained with Algorithm 1. $IED = \{(x_j^{eval}, y_j^{eval})\}_{j=1}^{M}$ − independent evaluation data set, where x_j^{eval} and y_j^{eval} are, respectively, the feature vector and the label of the jth evaluation sample; $y_j^{eval} \in \{+1, -1\}$, where + 1 and −1 represent the positive and negative classes, respectively; M is the number of evaluation samples.

Output: $AdaHDSVM(x)$ − ensembled HDSVMs

Initialization: $i \leftarrow 1$; $w_j^i = 1/M$, $j = 1, 2, ..., M$, where w_j^i is the weight of the jth evaluation sample in the ith iteration

Step 1. Calculate the error of $HDSVM_i$, called ε_i, using eq 8 as follows:

$$\varepsilon_i = \sum_{j=1}^{M} w_j^i \times F(y_j^{eval} \times sign(HDSVM_i(x_j^{eval}) - T_{pre})) \tag{8}$$

where $HDSVM_i(x_j^{eval})$ is the predicted probability of belonging to the positive class for x_j^{eval} by $HDSVM_i$; T_{pre} is a prescribed threshold; $sign(t)$ is a sign function that equals 1 when $t > 0$ and −1 otherwise; $F(t)$ is a scalar function that equals 1 when $t < 0$ and 0 otherwise.

Step 2. Compute the weight of $HDSVM_i$, denoted as β_i, based on ε_i:

$$\beta_i = \begin{cases} 0, & \varepsilon_i > 0.5 \text{ or } \varepsilon_i = 0 \\ \frac{1}{2}\log\frac{1-\varepsilon_i}{\varepsilon_i}, & \text{otherwise} \end{cases} \tag{9}$$

Step 3. Update the weight of each evaluation sample by eq 10:

$$w_j^{i+1} = \begin{cases} 1/M, & \beta_i = 0 \\ w_j^i \times \exp(-y_j^{eval} \times sign(HDSVM_i(x_j^{eval}) - T_{pre}) \times \beta_i)/Z, & \beta_i \neq 0 \end{cases}, \quad j = 1, 2, ..., M \tag{10}$$

where Z is a normalization factor and $Z = \sum_{j=1}^{M} w_j^i \times \exp(-y_j^{eval} \times sign(HDSVM_i(x_j^{eval}) - T_{pre}) \times \beta_i)$

Step 4. $i \leftarrow i + 1$; if $i \leq I$, go to Step 1; otherwise, return the ensembled classifier as follows: Return $AdaHDSVM(x) = \sum_{i=1}^{I} \beta_i \times HDSVM_i(x)$.

In this study, we use scikit-learn software,[54] which can be freely downloaded at http://scikit-learn.org/, to train the SVM and calculate the distance between the sample and the separating hyperplane of the SVM. Two parameters of the SVM, including penalty parameter C and RBF kernel width parameter γ, are optimized by performing a grid search strategy over 10-fold cross-validation.

Evaluation Indices. To evaluate the performances of the proposed methods, four evaluation indices,[55−64] i.e., Sensitivity (*Sen*), Specificity (*Spe*), Accuracy (*Acc*), and Mathew's correlation coefficient (*MCC*), are utilized as follows:

$$Sen = TP/(TP + FN) \tag{11}$$

$$Spe = TN/(TN + FP) \tag{12}$$

$$Acc = (TP + TN)/(TP + FP + TN + FN) \tag{13}$$

$$MCC = (TP \times TN - FP \times FN) / \sqrt{(TP + FP) \times (TN + FN) \times (TP + FN) \times (TN + FP)} \tag{14}$$

where *TP*, *TN*, *FP*, and *FN* represent true positives, true negatives, false positives, and false negatives, respectively.

Because the above four indices are threshold dependent, it is critical to select an appropriate threshold for fair comparisons between various predictors. In this work, we select the threshold T that maximizes the value of *MCC* over 10-fold cross-validation. Moreover, the area under the receiver operating characteristic curve (*AUC*),[59−63,65] which is threshold independent and reflects the overall prediction perform-

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

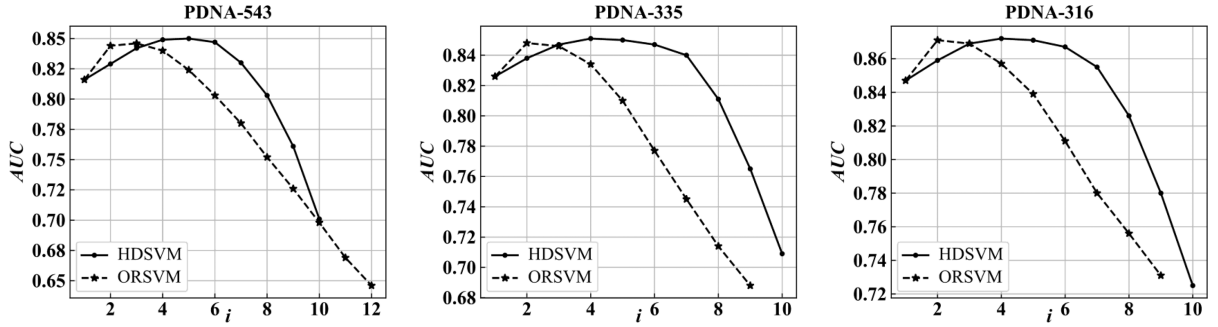


Figure 1. Variation curves of *AUC* values of $HDSVM_i$ and $ORSVM_i$ with respect to the values of i on three protein−DNA data sets.

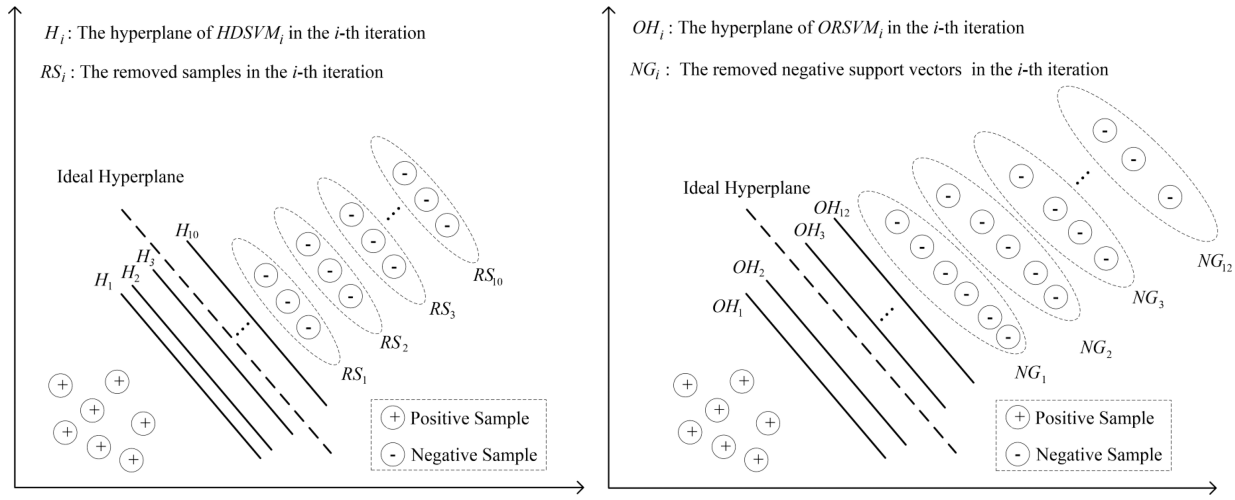


Figure 2. Relative position variations of the separating hyperplanes for HDSVMs and ORSVMs: (A) HDSVMs with HD-US on PDNA-543. (B) ORSVMs with SV-US on PDNA-543.

ance of a predictor, is utilized as another important evaluation index.

RESULTS AND DISCUSSIONS

Performance Comparisons between HD-US and SV-US. In this section, the performances of HD-US and SV-US are evaluated as follows. For HD-US, we evaluate the *AUC* performance of each of the 10 HDSVMs ($I = 10$). For SV-US, we repeatedly perform it and evaluate the corresponding *AUC* performance of each ORSVM until the new training data set becomes a balanced data set. Table 2 illustrates the *AUC* values of $HDSVM_i$ and $ORSVM_i$ versus the values of i on three protein−DNA data sets over 10-fold cross-validation, while Figure 1 plots the *AUC* variation curves of $HDSVM_i$ and $ORSVM_i$ versus the value of i for each data set.

From Table 2 and Figure 1, we observe an interesting phenomenon: the performances of $HDSVM_i$ and $ORSVM_i$ first improve and then decrease with the increase in the value of i on all three protein−DNA data sets. Taking PDNA-543 as an example, this phenomenon can be explained as follows. Initially, (i.e., $i = 1$), $HDSVM_i$ and $ORSVM_i$ are both trained on the original imbalanced data set. The separating hyperplanes of $HDSVM_i$ and $ORSVM_i$, denoted as H_i and OH_i, respectively, are pushed toward the positive samples, which leads to the poor performances. After removing a few negative samples that have the shortest distances to H_i, H_i moves away from positives and closer to negatives. Similarly, OH_i moves from positives to negatives if the *NGs* (i.e., negative support vectors) are removed. As a result, the performances of $HDSVM_i$ and $ORSVM_i$ both gradually improve (i.e., $1 < i < 5$ for $HDSVM_i$ and $1 < i < 3$ for $ORSVM_i$). When H_i and OH_i arrive at or close to an ideal position, $HDSVM_i$ and $ORSVM_i$ separately achieve the optimal performances (i.e., $i = 5$ for $HDSVM_i$ and $i = 3$ for $ORSVM_i$). Nevertheless, by continuously removing negatives, H_i and OH_i both move more and more closer to negatives, and the predictions of $HDSVM_i$ and $ORSVM_i$ are skewed to positive class. Hence, the corresponding performances will be deteriorated (i.e., $5 < i \leq 10$ for $HDSVM_i$ and $3 < i \leq 12$ for $ORSVM_i$). Figure 2 intuitively shows the relative position variations of the separating hyperplanes for HDSVMs with HD-US and ORSVMs with SV-US on PDNA-543.

It has not escaped our notice that the overall performances of HDSVMs are better than those of ORSVMs for each data set, which indicates HD-US outperforms SV-US. Specifically, the averaged *AUC* values of all HDSVMs are approximately 5.6%, 3.8%, and 2.3% higher than the corresponding values of ORSVMs on PDNA-543, PDNA-335, and PDNA-316, respectively. In addition, we notice that HD-US can generate more SVMs with higher performances than SV-US, which means that HD-US is more suitable for the subsequent SVMs ensemble. For example, on PDNA-335, HD-US produces four HDSVMs whose *AUC* values exceed 0.845, which is the baseline *AUC* value of the SVM trained on a balanced data set by random under-sampling as shown in the section "Performance Comparisons between E-HDSVM, GSVM-RU, and SVM-RU" (i.e., $HDSVM_3$, $HDSVM_4$, $HDSVM_5$, and $HDSVM_6$), while SV-US only produces two corresponding high-quality

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

Table 3. Values of Pe_i of $ORSVM_i$ on Three Protein−DNA Binding Site Data Sets over 10-Fold Cross-Validation

i	1	2	3	4	5	6	7	8	9	10	11	12
Pe_i on PDNA-543	26.1%	15.2%	10.5%	8.1%	6.5%	5.4%	4.6%	3.8%	3.2%	2.6%	2.1%	1.6%
Pe_i on PDNA-335	29.2%	16.4%	11.4%	8.6%	6.7%	5.4%	4.2%	3.2%	2.3%	–[a]	–	–
Pe_i on PDNA-316	26.1%	15.7%	11.1%	8.7%	7.1%	5.8%	4.6%	3.6%	2.8%	–	–	–

[a]'–' indicates that the corresponding value does not exist.

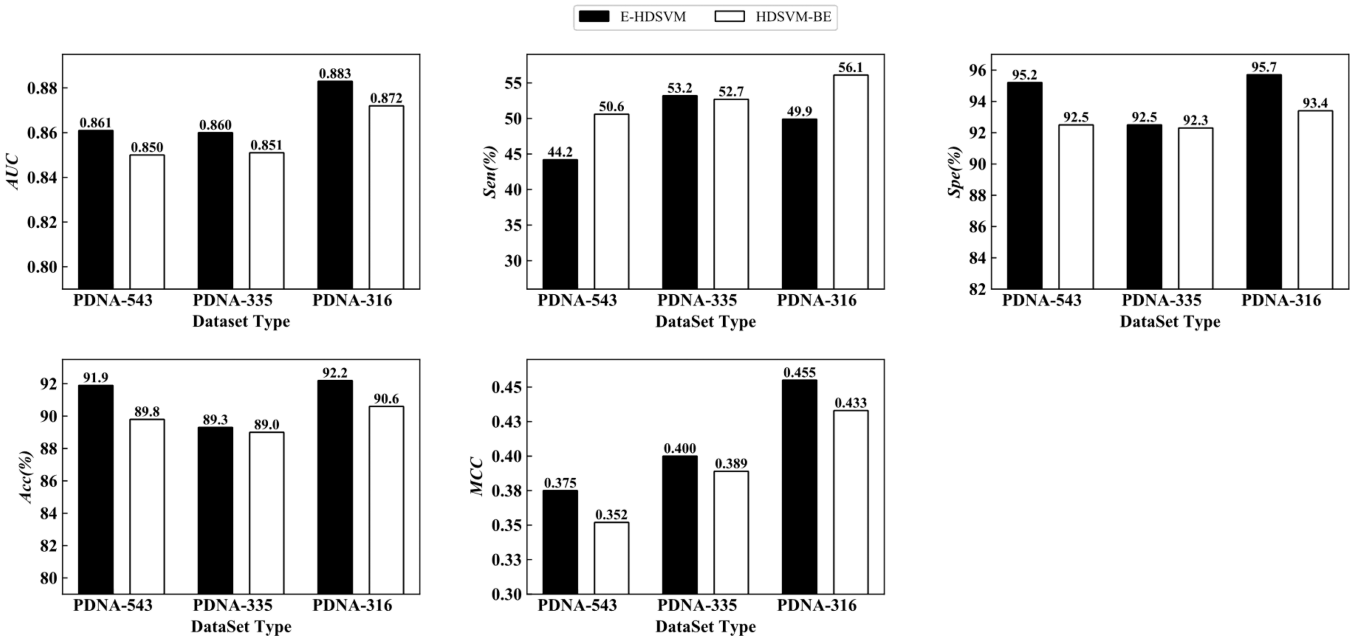


Figure 3. Performance comparisons between E-HDSVM and HDSVM-BE on three protein−DNA data sets over 10-fold cross-validation.

ORSVMs (i.e., $ORSVM_2$ and $ORSVM_3$) with *AUC* values greater than 0.845.

The reason that SV-US shows an inferior performance can be explained as follows. In SV-US, the separating hyperplane of ORSVM is gradually moved from positive samples to negatives and close to an ideal position by continuously removing *NGs*. However, if the sample number in the removed *NG* is too large in each iteration, the hyperplane will move with large span and easily miss the ideal position (in this work, we called this phenomenon ideal-hyperplane-missing, i.e., IHM). As a result, the corresponding ORSVM will achieve a suboptimal performance. To further investigate the above reason, in SV-US, we calculate the percentage, called Pe_i, between the sample number of NG_i and that of the original training data set in the ith iteration. Table 3 shows the value of Pe_i versus the value of i for each protein−DNA binding site data set.

Based on the experimental results listed in Table 3, we can explain the inferior performance of SV-US as follows. For each data set, more than 26% of samples are removed in the first iteration, and nearly a half of samples are removed in the first three iterations, which means that the hyperplane of ORSVM moves from positive samples to negatives with large spans, easily leading to an IHM. As a result, in the first three iterations, the performance of ORSVM can be improved but cannot achieve the optimal performance; after the third iteration, since the hyperplane misses the ideal position and moves toward negatives, the performance of ORSVM rapidly deteriorates. Different from SV-US, HD-US only removes 10% of negative samples in each iteration, which indicates that hyperplane of HDSVM is moved with a smaller span. Therefore, the hyperplane of HDSVM more easily arrives at or close to the ideal hyperplane position than ORSVM, which can explain why the overall performances of HDSVMs are superior to those of ORSVMs.

Decision-Level Fusion Helps to Improve the Performance of a Single Classifier. To demonstrate that the proposed decision-level fusion scheme (multiple HDSVMs are ensembled by EAdaBoost, denoted as E-HDSVM) is useful, we compare the performances between E-HDSVM and HDSVM-BE. Here, HDSVM-BE denotes the HDSVM that has the highest *AUC* value among all HDSVMs. For example, HDSVM-BE is $HDSVM_5$, $HDSVM_4$, and $HDSVM_4$ on PDNA-543, PDNA-335, and PDNA-316, respectively (refer to Table 2). Figure 3 shows the performance comparisons of the above two models on PDNA-543, PDNA-335, and PDNA-316 over 10-fold cross-validation.

From Figure 3, it is easy to see that E-HDSVM consistently outperforms HDSVM-BE on all three considered data sets. Concretely, the *Spe*, *Acc*, *MCC*, and *AUC* values of E-HDSVM increase by approximately 1.9%, 1.5%, 4.8%, and 1.2% on average over the respective values evaluated for HDSVM-BE on the three data sets. As for *Sen*, although the values of HDSVM-BE on PDNA-543 and PDNA-316 are higher than those of E-HDSVM, the corresponding values of *Spe* of HDSVM-BE are relatively lower. The underlying reason is that too many negative samples are predicted as positives by HDSVM-BE. Together with the fact that the number of negatives is far larger than that of positives, this makes the *MCC* performance of HDSVM-BE inferior to that of E-HDSVM. Based on the experimental results in Figure 3, we can conclude that our decision-level fusion strategy helps to

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

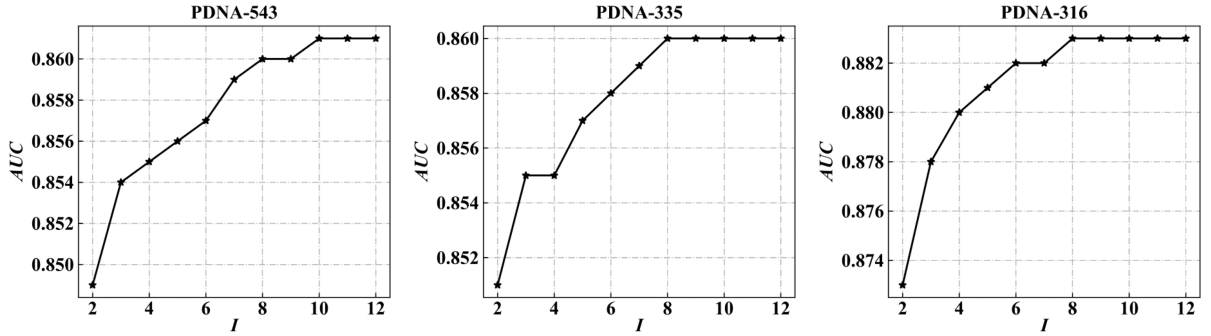


Figure 4. Variation curves of *AUC* values in the range of $I \in [2,12]$ on PDNA-543, PDNA-335, and PDNA-316 over 10-fold cross-validation.

enhance the prediction performance of a single classifier, even on the imbalanced data set.

Choosing the Value of *I* in E-HDSVM. In this section, we explain the reason why $I = 10$ is a better choice for E-HDSVM. We observe the performance variations of the E-HDSVM through gradually increasing the value of I from 2 to 12 with a step size of 1. For each value of I, we measure the *AUC* values of E-HDSVM on PDNA-543, PDNA-335, and PDNA-316 over 10-fold cross-validation. Figure 4 plots the variation curves of *AUC* performance versus the value of I on three individual data sets.

From Figure 4, the following phenomena can be observed. On PDNA-335 and PDNA-316, the overall trend of the *AUC* value continuously improves with increasing I when $I \leq 8$, and it keeps constant after $I > 8$. On PDNA-543, the value of *AUC* gradually increases with increasing I when $I \leq 10$ and keeps stable after $I > 10$. Therefore, the better values of I are 8 and 10 on PDNA-335 (PDNA-316) and PDNA-543, respectively, from the view of the single data set. However, considering the generality of the proposed E-HDSVM on different data sets, we choose the larger value ($I = 10$) for all the experiments in this work.

Analysis of the Contributions of Different Types of Features. In this section, the contributions of different types of features were carefully analyzed. Specifically, we separately used four serially combined features, i.e., PSSM (P), PSSM +PSS (PP), PSSM+PSS+PRSA (PPP), and PSSM+PSS+PRSA +AAFD-BN (PPPA), as the inputs of E-HDSVM models and then evaluated the prediction performances of the corresponding models. Table 4 summarizes the prediction performances of the four combined features on PDNA-543, PDNA-335, and PDNA-316 over 10-fold cross-validation. In addition, we further compared the performances of these four combined features on the two independent test data sets, i.e., PDNA-41 and PDNA-52, as presented in Table S2 in the Supporting Information.

From Table 4, the following two observations can be made: (1) The PSSM feature is very useful for the prediction of protein−DNA binding sites. On all the three considered data sets, E-HDSVM with a single PSSM feature achieved a relatively satisfactory performance regarding *MCC* and *AUC*. For example, the *MCC* values of E-HDSVM with the PSSM feature are 0.337, 0.355, and 0.418 on PDNA-543, PDNA-335, and PDNA-316, respectively, while the *AUC* values are 0.836, 0.833, and 0.860, respectively, on the three data sets. Both the *MCC* and *AUC* values are not far below those of the best performer, i.e. E-HDSVM with the PPPA feature, which demonstrates the efficacy of the PSSM feature for protein−DNA binding site prediction.

Table 4. Prediction Performances of the Four Combination Features on the Three Protein−DNA Binding Site Data Sets over 10-Fold Cross-Validation

Data Set	Feature	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
PDNA-543	P	43.9	94.0	90.7	0.337	0.836
	PP	**45.9**	93.9	90.8	0.351	0.845
	PPP	43.1	**95.4**	**92.0**	0.373	0.856
	PPPA	44.2	95.2	91.9	**0.375**	**0.861**
PDNA-335	P	51.9	90.8	87.6	0.355	0.833
	PP	51.3	92.0	88.7	0.374	0.844
	PPP	50.4	**93.3**	**89.7**	0.396	0.858
	PPPA	**53.2**	92.5	89.3	**0.400**	**0.860**
PDNA-316	P	38.4	**97.3**	**92.7**	0.418	0.860
	PP	41.3	97.0	**92.7**	0.432	0.868
	PPP	48.0	96.0	92.3	0.448	0.878
	PPPA	**49.9**	95.7	92.2	**0.455**	**0.883**

(2) PSS, PRSA, and AAFD-BN are also beneficial for the prediction of protein−DNA binding sites. As illustrated in Table 4, the performance of E-HDSVM is consistently improved by gradually adding PSS, PRSA, and AAFD-BN to the PSSM feature for all three considered data sets. Taking the results on PDNA-543 as an example, the *AUC* value was improved from 0.836 to 0.845, 0.856, and 0.861 after gradually adding PSS, PRSA, and AAFD-BN, respectively, to the PSSM feature. Similarly, the *MCC* value was also improved from 0.337 to 0.351, 0.373, and 0.375. On average, the E-HDSVM model with the PPPA feature improved by approximately 11.0% and 3.0% for *MCC* and *AUC*, respectively, compared with the E-HDSVM model with only the PSSM feature on the three considered data sets.

Performance Comparisons between E-HDSVM, GSVM-RU, and SVM-RU. To further examine the efficacy of the proposed E-HDSVM, we compare it with GSVM-RU, including GSVM-RC and GSVM-RD, and a baseline model SVM-RU that trains the SVM on a balanced data set by using the random under-sampling technique. Table 5 summarizes the performances of E-HDSVM, GSVM-RD, GSVM-RC, and SVM-RU on PDNA-543, PDNA-335, and PDNA-316 over 10-fold cross-validation.

It is straightforward to find from Table 5 that the performance of E-HDSVM is obviously better than the other three considered methods. Compared with GSVM-RD and GSVM-RC, E-HDSVM achieves the best values of *MCC* and *AUC* on all three protein−DNA data sets. For example, the *MCC* and *AUC* of E-HDSVM are 9.3% and 1.5%, respectively,

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

Table 5. Performance Comparisons between E-HDSVM, GSVM-RD, GSVM-RC, and SVM-RU on Three Protein−DNA Binding Site Data Sets over 10-Fold Cross-Validation

Data Set	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
PDNA-543	E-HDSVM	44.2	95.2	91.9	**0.375**	**0.861**
	GSVM-RD	53.1	91.3	88.8	0.343	0.848
	GSVM-RC	36.2	**96.8**	**92.8**	0.363	0.820
	SVM-RU	**54.9**	90.2	87.9	0.335	0.844
PDNA-335	E-HDSVM	53.2	92.5	89.3	**0.400**	**0.860**
	GSVM-RD	**55.6**	90.9	87.9	0.381	0.851
	GSVM-RC	41.5	**95.6**	**91.1**	0.391	0.830
	SVM-RU	51.8	91.6	88.3	0.369	0.845
PDNA-316	E-HDSVM	49.9	95.7	92.2	**0.455**	**0.883**
	GSVM-RD	**53.7**	94.2	91.1	0.436	0.876
	GSVM-RC	43.9	**96.9**	**92.8**	0.448	0.849
	SVM-RU	50.7	94.0	90.7	0.408	0.863

higher than those of GSVM-RD on PDNA-543. As another example, the enhancements of *MCC* and *AUC* on PDNA-316 reach 1.6% and 4.0%, respectively, in the comparisons with GSVM-RC. Although GSVM-RC has higher values of *Spe* and *Acc* than E-HDSVM on each data set, its corresponding *Sen* is significantly lower. The lower *Sen* indicates that GSVM-RC tends to predict positive samples as negatives, which is opposite to the purpose of learning from the imbalanced data set. Compared to SVM-RU, the performance of E-HDSVM shows an absolute advantage. Specifically, E-HDSVM separately shares approximately 2.8%, 2.5%, 10.6%, and 2.0% improvements of *Spe*, *Acc*, *MCC*, and *AUC* on average on the three data sets. In addition, on PDNA-335, all five evaluation indices of E-HDSVM are higher than the corresponding values measured for SVM-RU.

To further evaluate the generalization performance of E-HDSVM, we compare it with GSVM-RD, GSVM-RC, and SVM-RU on the independent test data sets. Specifically, for each type of the above four methods, we use it to train a model on the training data set with the parameters selected in cross-validation and test the trained model on the corresponding independent test data set. In this study, we use PDNA-543 and PDNA-335 as the training data sets, and the corresponding independent test data sets are PDNA-41 and PDNA-52, respectively. Table 6 shows the performances of E-HDSVM and other three methods on the two independent test data sets in detail.

Table 6. Performance Comparisons between E-HDSVM, GSVM-RD, GSVM-RC, and SVM-RU on PDNA-41 and PDNA-52

Data Set	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
PDNA-41	E-HDSVM	44.0	95.2	92.7	**0.340**	**0.851**
	GSVM-RD	52.3	90.9	88.9	0.297	0.836
	GSVM-RC	33.5	**97.5**	**94.3**	**0.340**	0.805
	SVM-RU	**54.0**	90.3	88.5	0.297	0.839
PDNA-52	E-HDSVM	**51.8**	94.9	92.5	0.405	**0.876**
	GSVM-RD	51.0	94.0	91.5	0.371	0.868
	GSVM-RC	43.8	**96.8**	**93.8**	**0.412**	0.854
	SVM-RU	45.8	93.8	91.1	0.331	0.855

From Table 6, we can conclude that the generalization capability of E-HDSVM outperforms that of the other three considered methods. In terms of *AUC*, E-HDSVM achieves the highest values among the four methods for both PDNA-41 and PDNA-52 data sets. The values of *AUC* for E-HDSVM are increased 1.4% and 0.9% over those of the second best method on PDNA-41 and PDNA-52, respectively. Moreover, the improvements of *AUC* between the worst method and E-HDSVM reach 5.7% and 2.6% for PDNA-41 and PDNA-52, respectively. From the view of *MCC*, the performance of E-HDSVM is also satisfactory. Compared with GSVM-RD and SVM-RU, E-HDSVM achieves 11.8% and 18.4% average improvements on the two validation data sets, although it has a slightly lower *MCC* value than GSVM-RC on PDNA-52.

Performance Comparisons between E-HDSVM and the CNN Model. Recently, deep learning models, such as convolutional neural network (CNN)[66] and long short-term memory (LSTM),[67] have been widely used in bioinformatics studies. In view of this, we further compared the proposed E-HDSVM model with one of the typical deep learning models, i.e., CNN, in this section. On the one hand, CNN is one of the most commonly used deep learning models in bioinformatics studies; on the other hand, to the best of our knowledge, there is no available deep-learning-based web server for protein−DNA binding sites prediction that can be directly used for comparison. In view of this, we deigned two types of CNN models, denoted as CNN-A and CNN-B, which are trained by two different training strategies, i.e., TS-A and TS-B, respectively, for protein−DNA binding sites prediction. The details of two CNN models are illustrated in Text S1 in the Supporting Information. The performance comparisons between the proposed E-HDSVM, CNN-A, and CNN-B on PDNA-543, PDNA-335, PDNA-316, PDNA-41, and PDNA-52 are summarized in Table 7.

As shown in Table 7, E-HDSVM achieves a better performance than CNN-A and CNN-B with respect to the values of *MCC* and *AUC*, which demonstrates the strong performance of E-HDSVM again. For example, compared with CNN-A, E-HDSVM obtains an average of 13.3% *MCC* improvement on the five data sets, while the *AUC* value of E-HDSVM is about an average of 1.3% higher than that of CNN-B at the same time. On the other hand, we can observe that the performance of CNN-B is better than that of CNN-A, which indicates that the proposed training strategy TS-B can partially relieve the negative effect caused by class imbalance (see details in Text S1). Compared with CNN-A, CNN-B achieves an average of 6.2% improvement of *MCC* on the five considered data sets. Although the *AUC* values of CNN-B are slightly lower than that of CNN-A on PDNA-316 and PDNA-335, CNN-B still achieves approximately 0.6%, 0.6%, and 1.3% enhancements on PDNA-543, PDNA-41, and PDNA-52, respectively.

Considering that the deep learning model (e.g., CNN used in this study) prefers more training data, we further constructed a large-scale training data set, called PDNA-1151, as follows. First, we combined PDNA-543, PDNA-335, and PDNA-316 to form a data set PDNA-1194 consisting of 1194 sequences. Then, the CD-HIT-2D software[41] was used to remove the sequences in PDNA-1194 that had more than 40% sequence identity with any sequences in PDNA-41 or PDNA-52. Finally, the remaining 1151 sequences constituted the new data set PDNA-1151, among which there were 20,936 binding sites and 263,977 nonbinding sites. In this study, we separately

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

Table 7. Performance Comparisons between E-HDSVM, CNN-A, and CNN-B on the Five Protein−DNA Binding Site Data Sets

Data Set	Model	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
PDNA-543[a]	E-HDSVM	44.2	**95.2**	**91.9**	**0.375**	**0.861**
	CNN-A	46.2	93.2	90.1	0.335	0.848
	CNN-B	**51.8**	92.7	90.0	0.364	0.853
PDNA-335[a]	E-HDSVM	53.2	92.5	89.3	**0.400**	**0.860**
	CNN-A	**59.3**	88.1	85.7	0.358	0.844
	CNN-B	44.3	**94.3**	**90.1**	0.373	0.842
PDNA-316[a]	E-HDSVM	49.9	95.7	**92.2**	**0.455**	**0.883**
	CNN-A	**51.5**	94.2	90.9	0.418	0.871
	CNN-B	47.2	**95.9**	**92.2**	0.439	0.870
PDNA-41[b]	E-HDSVM	44.0	95.2	92.7	**0.340**	**0.851**
	CNN-A	36.0	**96.1**	**93.1**	0.307	0.845
	CNN-B	**52.6**	91.4	89.5	0.309	0.850
PDNA-52[c]	E-HDSVM	51.8	**94.9**	**92.5**	**0.405**	**0.876**
	CNN-A	**55.5**	90.7	88.7	0.329	0.849
	CNN-B	47.7	94.8	92.1	0.370	0.860

[a]We compare the performances of three models on PDNA-543, PDNA-335, and PDNA-316 by 10-fold cross-validation. [b]We evaluate the performances of three models on independent test data set PDNA-41. The corresponding training data set is PDNA-543. [c]We evaluate the performances of three models on independent test data set PDNA-52. The corresponding training data set is PDNA-335.

trained E-HDSVM, CNN-A, and CNN-B on the new data set PDNA-1151 and tested the trained models with PDNA-41 and PDNA-52, respectively. Table 8 illustrates the generalization performance comparisons between E-HDSVM, CNN-A, and CNN-B on PDNA-41 and PDNA-52.

Table 8. Generalization Performance Comparisons between E-HDSVM, CNN-A, and CNN-B on PDNA-41 and PDNA-52 with the New Constructed PDNA-1151 as Training Data Set

Data Set	Model	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
PDNA-41	E-HDSVM	**48.8**	94.3	92.1	**0.349**	**0.858**
	CNN-A	44.4	95.2	92.7	0.344	0.850
	CNN-B	41.4	**96.0**	**93.3**	0.348	0.854
PDNA-52	E-HDSVM	**61.0**	95.2	93.2	**0.478**	**0.905**
	CNN-A	37.1	**97.6**	**94.2**	0.394	0.878
	CNN-B	53.5	95.6	93.2	0.439	0.895

From Tables 7 and 8, the following two observations can be made:

(1) A large-scale data set helps to improve the performances of CNN models. It is found that the performances of CNN models trained on a large-scale data set (i.e., PDNA-1151) are better than those of CNN models trained on a relative small data set (i.e., PDNA-543 or PDNA-335). Taking the independent test data set PDNA-41 as an example, the CNN-A and CNN-B trained with PDNA-1151 achieve a 12.1% and 12.6%, respectively, improvement of *MCC* values in comparisons with the CNN-A and CNN-B trained with PDNA-543. As for the independent test data set PDNA-52, the *AUC* values of CNN-A and CNN-B trained with PDNA-1151 are also 3.4% and 4.1%, respectively, higher than those of the corresponding models trained with PDNA-335.

(2) E-HDSVM again performs better than CNN-A and CNN-B even when all of the three models are trained on the large-scale data set PDNA-1151. Compared to CNN-A and CNN-B, E-HDSVM achieves averaged improvements of 2.0% and 0.8%, respectively, of *AUC* values on the two independent test data sets. We speculate that the following aspects may account for the inferior performances of the CNN models. First, since CNN is not specifically designed to deal with class imbalance problem,[66] it cannot perform well when there is a serious class imbalance in the data set (e.g., DNA-binding data set in this study). As described in Text S1, the batch data set in each iteration used for training CNN-A is also a severe imbalanced data set, which may cause the CNN model to learn skewed knowledge leading to a deteriorated performance, while in CNN-B, we construct a balanced batch data set by random under-sampling from the original imbalanced training data set, which partially relieves the negative impact of class imbalance and thus improves the performance of the CNN model (i.e., the performance CNN-B is better than that of CNN-A, refer to Tables7 and 8). Nevertheless, the random under-sampling changes the original data distribution and may lose information, leading to a nonoptimal performance of the CNN model. By contrast, E-HDSVM is specially designed to solve the class imbalance problem. The above-mentioned issues may explain why E-HDSVM achieves a better performance than the CNN models on the considered DNA-binding data sets.

Comparisons with Existing Predictors. Based on the proposed E-HDSVM, we further implement a new sequence-based predictor, called DNAPred, for the prediction of protein−DNA binding residues. To demonstrate the strong performance of DNAPred, we compare it with other popular sequence-based protein−DNA binding site predictors on PDNA-543, PDNA-335, PDNA-316, PDNA-41, and PDNA-52, respectively.

Table 9 summarizes the prediction performances of DNAPred and TargetDNA,[39] which is one of the most recently released predictors, on PDNA-543. For the purpose of fairness, we take 10-fold cross-validation in the comparison because TargetDNA is also evaluated on PDNA-543 under 10-fold cross-validation. In addition, we compare DNAPred with TargetDNA under two different thresholds, as does in TargetDNA. One is the threshold that makes $Sen \approx Spe$, and the other is the threshold that makes $FPR = 1 - Spe \approx 5\%$, where *FPR* denotes False Positive Rate.

Table 9. Performance Comparisons between DNAPred and TargetDNA on PDNA-543 over 10-Fold Cross-Validation

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
TargetDNA ($Sen \approx Spe$)[a,c]	77.0	77.1	77.0	0.304	0.845
DNAPred ($Sen \approx Spe$)[a]	**77.1**	**78.5**	**78.4**	**0.318**	**0.861**
TargetDNA ($FPR \approx 5\%$)[b,c]	40.6	95.0	91.4	0.339	0.845
DNAPred ($FPR \approx 5\%$)[b]	**44.9**	**95.0**	**91.7**	**0.373**	**0.861**

[a]The threshold T which makes $Sen \approx Spe$ in cross-validation is chosen. [b]The threshold T which makes $FPR = 1 - Spe \approx 5\%$ in cross-validation is chosen. [c]Results excerpted from TargetDNA.[39]

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

From Table 9, it is easy to find that the proposed DNAPred consistently outperforms TargetDNA for all five indices under both of the considered thresholds. Taking *MCC* and *AUC*, which are two overall performance evaluation indices, as examples, DNAPred improves the value of *MCC* by 4.6% and 10.0%, respectively, under the thresholds that make *Sen* ≈ *Spe* and *FPR* ≈ 5%; as to *AUC*, an improvement of 1.9% is also observed under the two thresholds.

Further, we compare our predictor with TargetS[42] and EC-RUS[68] on PDNA-335 under five-fold cross-validation, as shown in Table 10. Again, the proposed DNAPred achieves the best overall performances regarding *MCC* and *AUC*. Compared with the second-best performer, i.e., EC-RUS, improvements of 3.2% and 0.5% are achieved by DNAPred for *MCC* and *AUC*, respectively.

Table 10. Comparisons with EC-RUS and TargetS on PDNA-335 over Five-Fold Cross-Validation

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
TargetS[a,c]	41.7	94.5	89.9	0.362	0.824
EC-RUS[b]	48.7	**95.1**	**92.6**	0.378	0.852
DNAPred[c]	**54.3**	91.7	88.6	**0.390**	**0.856**

[a]Results excerpted from TargetS.[42] [b]Results excerpted from EC-RUS.[68] [c]The threshold *T* which maximizes the value of *MCC* in cross-validation is chosen.

The performance comparisons on PDNA-316 over 10-fold cross-validation of DNAPred and other common protein−DNA binding site predictors, including DBS-PRED,[69] BindN,[70] DNABindR,[71] DISIS,[19] DP-Bind,[20] BindN-rf,[16] MetaDBSite,[17] and TargetDNA,[39] are listed in Table 11.

Table 11. Performance Comparisons between DNAPred and the State-of-the-Art Predictors on PDNA-316 over 10-Fold Cross-Validation

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
DBS-PRED[a]	53.0	76.0	75.0	0.170
BindN[a]	54.0	80.0	78.0	0.210
DNABindR[a]	66.0	74.0	73.0	0.230
DISIS[a]	19.0	**98.0**	**92.0**	0.250
DP-Bind[a]	69.0	79.0	78.0	0.290
BindN-rf[a]	67.0	83.0	82.0	0.320
MetaDBSite[a]	77.0	77.0	77.0	0.320
TargetDNA (*Sen* ≈ *Spe*)[a,b]	78.0	78.0	78.0	0.339
TargetDNA (*FPR* ≈ 5%)[a,c]	43.0	95.0	91.0	0.375
DNAPred (*Sen* ≈ *Spe*)[b]	**80.0**	79.9	79.9	0.370
DNAPred (*FPR* ≈ 5%)[c]	52.1	95.1	91.8	**0.452**

[a]Results excerpted from TargetDNA.[39] [b]The threshold *T* which makes *Sen* ≈ *Spe* in cross-validation is chosen. [c]The threshold *T* which makes *FPR* ≈ 5% in cross-validation is chosen.

The results illustrated in Table 11 intuitively demonstrate that DNAPred enjoys better performance than the other eight predictors in terms of *MCC*. Compared with TargetDNA, the second best predictor among all predictors, under *Sen* ≈ *Spe*, the values of *Sen*, *Spe*, *Acc*, and *MCC* of DNAPred are enhanced by 2.6%, 2.4%, 2.4%, and 9.1%, respectively; under *FPR* ≈ 5%, the *MCC* of DNAPred reaches 0.452, which is 20.5% higher than the corresponding value of TargetDNA. Moreover, all indices of DNAPred under *Sen* ≈ *Spe* are better than the values evaluated for DBS-PRED, DNABindR, DP-Bind, and MetaDBSite. Taking DNABindR as an example, the proposed predictor obtains 21.2%, 8.0%, 9.5%, and 60.9% increases in *Sen*, *Spe*, *Acc*, and *MCC*, respectively. In addition, we can notice that DISIS has the highest *Spe* value (98.0%) but the lowest *Sen* (19.0%). The reason for the lower *Sen* is that DISIS predicts too many false negatives.

To further highlight the generalization capability of our predictor, we compare it with other state-of-the-art protein−DNA binding site predictors by independent validation (the process of independent validation refers to the section "Performance Comparisons between E-HDSVM, GSVM-RU, and SVM-RU") on PDNA-41 and PDNA-52. Table 12 displays the performance comparisons between BindN,[70] ProteDNA,[72] BindN+,[73] MetaDBSite,[17] DP-Bind,[20] DNABind,[74] TargetDNA,[39] and DNAPred on PDNA-41. Under *FPR* ≈ 5%, DNAPred obtains the highest *MCC* value of 0.337, which is increased by 12.3% and 89.3% over TargetDNA and BindN+, respectively. More significantly, all four indices of DNAPred are higher than the values measured for MetaDBSite, and the improvements of *Sen*, *Spe*, *Acc*, and *MCC* are 30.7%, 1.6%, 2.2%, and 52.5%, respectively. Under *Sen* ≈ *Spe*, although DNAPred has a 3.3% decrease in *MCC* compared to TargetDNA, it remains competitive in the comparisons with BindN, ProteDNA, BindN+, MetaDBSite, and DP-Bind. For example, the *MCC* of our method is 22.1% better than the corresponding value of BindN+ (*Spe* ≈ 85%). As another example, compared with BindN, the *MCC* value of DNAPred is increased by 81.8%.

Table 12. Comparisons with Other Popular Predictors on PDNA-41 under Independent Validation

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*
BindN[a]	45.6	80.9	79.2	0.143
ProteDNA[a]	4.8	**99.8**	**95.1**	0.160
BindN+ (*FPR* ≈ 5%)[a,b]	24.1	95.1	91.6	0.178
BindN+ (*Spe* ≈ 85%)[a,c]	50.8	85.4	83.7	0.213
MetaDBSite[a]	34.2	93.4	90.4	0.221
DP-Bind[a]	61.7	82.4	81.4	0.241
DNABind[a]	70.2	80.3	79.8	0.264
TargetDNA (*Sen* ≈ *Spe*)[a,d]	60.2	85.8	84.5	0.269
TargetDNA (*FPR* ≈ 5%)[a,b]	45.5	93.3	90.9	0.300
DNAPred (*Sen* ≈ *Spe*)[d]	**76.1**	76.7	76.1	0.260
DNAPred (*FPR* ≈ 5%)[b]	44.7	94.9	92.4	**0.337**

[a]Results excerpted from TargetDNA.[39] [b]The threshold *T* which makes *FPR* ≈ 5% in cross-validation is chosen. [c]The threshold *T* which makes *Spe* ≈ 85% in cross-validation is chosen. [d]The threshold *T* which makes *Sen* ≈ *Spe* in cross-validation is chosen.

Table 13 illustrates the results by comparing DNAPred with DNABR,[18] alignment-based,[42] MetaDBSite,[17] and TargetS[42] on PDNA-52. Compared to TargetS, DNAPred enjoys 25.4%, 7.4%, and 4.8% increases in terms of *Sen*, *MCC*, and *AUC*, respectively. Moreover, the evaluation indices of our predictor are remarkably better those of DNABR, alignment-based, and MetaDBSite. Taking alignment-based as an example, its *Sen*, *Spe*, *Acc*, and *MCC* are, respectively, 48.6%, 0.6%, 2.2%, and 53.1% lower than the values measured for DNAPred. Additionally, we can see that MetaDBSite produces the highest *Sen* value of 58.0%. However, the corresponding *Spe* is lowest among the five predictors. The reason is that too many negative samples are predicted as positives in MetaDBSite. Along with the scenario that the number of

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

Table 13. Performance Comparisons between the Proposed Predictor DNAPred and Other Existing Predictors on PDNA-52

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
DNABR[a]	40.7	87.3	84.6	0.185	-
alignment-based[a]	26.6	94.3	90.5	0.190	-
MetaDBSite[a]	**58.0**	76.4	75.2	0.192	-
TargetS[a,b]	41.3	**96.5**	**93.3**	0.377	0.836
DNAPred[b]	51.8	94.9	92.5	**0.405**	**0.876**

[a]Results excerpted from TaegetS;[42] '–' indicates that the corresponding value does not exist. [b]The threshold *T* which maximizes the value of *MCC* in cross-validation is chosen.

negatives is far more than that of positives, the performance of MetaDBSite with respect to *MCC* is lower.

Case Studies. To further demonstrate the effectiveness of the proposed predictor, we select two DNA-binding proteins, named 4WWC-B (PDB ID: 4WWC, chain: B) and 5BMZ-D (PDB ID: 5BMZ, chain: D), from the independent test data set PDNA-41 for case studies. For each selected protein, we use the three predictors (i.e., DNAPred, TargetDNA,[39] and DNABind[74]) that show the best *MCC* performances in Table 12 to predict its DNA-binding residues, where TargetDNA and DNABind are available at http://csbio.njust.edu.cn/bioinf/TargetDNA/ and http://mleg.cse.sc.edu/DNABind/, respectively. Figures 5 and 6 illustrate the prediction results of 4WWC-B and 5BMZ-D, respectively, by the three predictors.

From Figure 5, it is easy to see that the proposed DNAPred outperforms both TargetDNA and DNABind on 4WWC-B. DNAPred correctly identifies 15 out of the 17 observed binding residues (i.e., B13, S37, E38, R39, S48, R49, M50, T51, R53, R67, K69, G70, R71, G72, and T73, highlighted in blue), and only 1 nonbinding residue (i.e., I47, highlighted in red) is mistakenly predicted as a binding residue. By contrast, TargetDNA correctly identifies 14 out of the 17 binding residues (i.e., B13, E38, R39, S48, R49, M50, T51, R53, R67, K69, G70, R71, G72, and T73) but with 4 false positives (i.e., K6, I47, Y205, and G221). DNABind predicts 15 true positives, which are identical to those of DNAPred, but it mistakenly predicts 38 nonbinding residues as binding residues. As for 5BMZ-D, DNAPred also achieves the best performance, with 14 true positives, 6 false positives, and 3 false negatives (TargetDNA: 13 true positives, 8 false positives, and 4 false negatives. DNABind: 14 true positives, 10 false positives, and 3 false negatives), as shown in Figure 6.

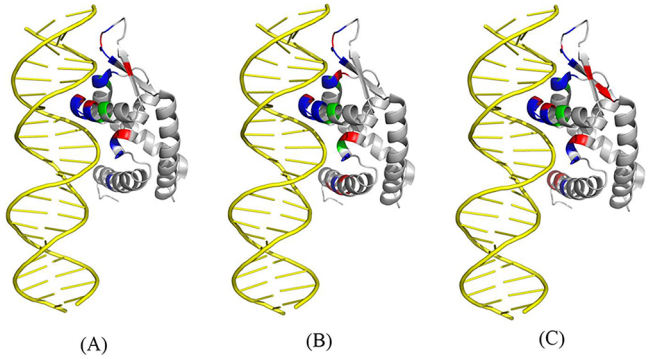


Figure 6. Visualization of prediction results for 5BMZ-D: (A) DNAPred, (B) TargetDNA, and (C) DNABind. The color scheme is used as follows: DNA in yellow, true positives in blue, false positives in red, false negatives in green. The pictures are made with PyMOL.[75]

In addition, we found that DNABind had the worst performance among the three predictors because it predicted too many false positives. The underlying reason for this poor performance can be explained as follows. DNABind was trained on a small DS123 data set[74] that only consisted of 2912 DNA-binding sites and 16,016 nonbinding sites, while TargetDNA and DNAPred were both trained on a relatively larger data set, PDNA-543, which contained 9549 DNA-binding sites and 134,995 nonbinding sites. A larger data set may contain more discriminative information, which is beneficial for training a machine-learning based predictor. We believe this is one of the important reasons that account for the poor performance of DNABind.

Large-Scale Application. To show the applicability of DNAPred, a large-scale prediction is performed in this work as follows. First, we downloaded 2572 protein sequences deposited in the Swiss-Prot database[47] between January 1, 2018, and December 31, 2018. Then, we used the CD-HIT-2D software[41] to reduce the sequence identity of the downloaded sequences. Sequences that had more than 40% sequence identity with the sequences in the training data set of DNAPred (i.e., PDNA-543) were removed; the remaining 2441 sequences constituted a nonredundant large-scale data set, denoted as PDNA-Large, which contained 1,297,707 amino acid residues. Finally, DNAPred was used to predict the DNA-binding sites of the sequences in PDNA-Large, and the corresponding prediction results can be downloaded at http://csbio.njust.edu.cn/bioinf/dnapred. To facilitate a better understanding of protein−DNA interactions, we will periodically

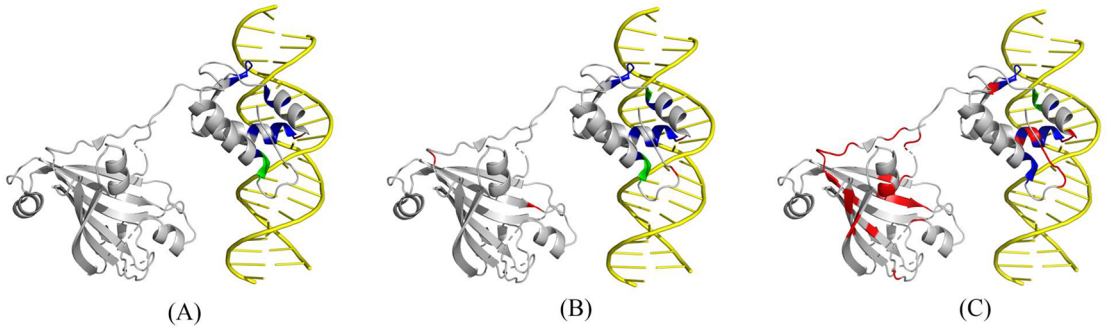


Figure 5. Visualization of the prediction results for 4WWC-B: (A) DNAPred, (B) TargetDNA, and (C) DNABind. The color scheme is used as follows: DNA in yellow, true positives in blue, false positives in red, false negatives in green. The pictures are made with PyMOL.[75]

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

release the prediction results of DNAPred for newly deposited protein sequences in the Swiss-Prot database.

CONCLUSIONS

In this study, a new machine-learning algorithm (E-HDSVM), which incorporates multiple HDSVMs by EAdaBoost, is designed to effectively learn from imbalanced data. Using the proposed E-HDSVM, a robust sequence-based predictor is developed for the prediction of protein−DNA binding residues, called DNAPred. By comparison with several state-of-the-art sequence-based predictors on five protein−DNA binding site data sets, the efficacy of the proposed predictor has been demonstrated. The superior performance of DNAPred is mainly attributed to the strong capability of E-HDSVM for effectively dealing with the data imbalance problem.

Although DNAPred achieves some improvements, there is still room to further enhance its performance due to the following three points. First, the input feature is generated by serially combining PSSM, PSS, PRSA, and AAFD-BN feature vectors, which may result in information redundancy. In our future work, we will take into account other strategies, such as parallel feature fusion,[76] feature reduction,[77] and feature selection,[78] to effectively utilize multiple types of features. Moreover, EAdaBoost was used in this study, but it may not be the best ensemble algorithm for the task. As is known, EAdaBoost shares the advantages of simple principle and easy implementation and overcomes the overfitting problem. However, there is no evidence that EAdaBoost is the best algorithm for decision-level fusion. Therefore, we will consider employing other advanced ensemble learning algorithms in the future. Finally, it might be a promising way to further improve the performance of DNAPred by incorporating more informative heterogeneous features that are complementary to the encoded features that are currently being utilized. A good choice is to use features that can be calculated by using feature-generating software tools or online web servers from the DNA, RNA, or protein sequences.

It should be noted that the proposed DNAPred is specifically designed to predict DNA-binding residues from protein sequences. In view of the diversity of ligands and the importance of protein−ligand interactions, we will investigate the applicability of DNAPred to other types of ligand-binding site prediction problems, e.g., RNA-binding sites[79] and ATP-binding sites,[80] in future work.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.8b00749.

Information as mentioned in the text. (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: njyudj@njust.edu.cn

ORCID

Dong-Jun Yu: 0000-0002-6786-8053

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Nos. 61772273, 61373062, and 61876072), the Fundamental Research Funds for the Central Universities (No. 30918011104), and the Natural Science Foundation of Anhui Province of China (No. KJ2018A0572).

ABBREVIATIONS

DNA, deoxyribonucleic acid; SVM, support vector machine; AdaBoost, adaptive boosting; MCC, Mathew's correlation coefficient; AUC, area under the receiver operating characteristic curve; NN, neural network; LR, logistic regression; RF, random forest; PLR, penalized logistic regression; KLR, kernel logistic regression; HMM, hidden Markov model; PDB, protein data bank; GSVM-RU, granular support vector machine-repetitive under-sampling; E-HDSVM, ensembled hyperplane-distance-based support vector machines; PSSM, position specific scoring matrix; PSS, predicted secondary structure; PRSA, predicted relative solvent accessibility; AAFD-BN, amino acid frequency difference between binding and nonbinding; SV-US, support-vector-based under-sampling; HD-US, hyperplane-distance-based under-sampling; EAdaBoost, enhanced adaptive boosting

REFERENCES

(1) Aeling, K. A.; Steffen, N. R.; Johnson, M.; Hatfield, G. W.; Lathrop, R. H.; Senear, D. F. DNA Deformation Energy as an Indirect Recognition Mechanism in Protein-DNA Interactions. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2007**, *4*, 117−125.

(2) Si, J.; Zhao, R.; Wu, R. An Overview of the Prediction of Protein DNA-Binding Sites. *Int. J. Mol. Sci.* **2015**, *16*, 5194−5215.

(3) Wong, K. C.; Li, Y.; Peng, C.; Wong, H. S. A Comparison Study for DNA Motif Modeling on Protein Binding Microarray. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2016**, *13*, 261−271.

(4) Schmidtke, P.; Barril, X. Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites. *J. Med. Chem.* **2010**, *53*, 5858−5867.

(5) Mandel-Gutfreund, G. Y.; Margalit, H. Quantitative Parameters for Amino Acid-Base Interaction: Implications for Prediction of Protein-DNA Binding Sites. *Nucleic Acids Res.* **1998**, *26*, 2306−2312.

(6) Orengo, C. A.; Michie, A.; Jones, S.; Jones, D. T.; Swindells, M.; Thornton, J. M. CATH-A Hierarchic Classification of Protein Domain Structures. *Structure* **1997**, *5*, 1093−1109.

(7) Luscombe, N. M.; Laskowski, R. A.; Thornton, J. M. Amino Acid-Base Interactions: A Three-Dimensional Analysis of Protein-DNA Interactions at an Atomic Level. *Nucleic Acids Res.* **2001**, *29*, 2860−2874.

(8) Jones, S.; Shanahan, H. P.; Berman, H. M.; Thornton, J. M. Using Electrostatic Potentials to Predict DNA-Binding Sites on DNA-Binding Proteins. *Nucleic Acids Res.* **2003**, *31*, 7189−7198.

(9) Tsuchiya, Y.; Kinoshita, K.; Nakamura, H. Structure-Based Prediction of DNA-Binding Sites on Proteins Using the Empirical Preference of Electrostatic Potential and the Shape of Molecular Surfaces. *Proteins: Struct., Funct., Genet.* **2004**, *55*, 885−894.

(10) Morozov, A. V.; Havranek, J. J.; Baker, D.; Siggia, E. D. Protein-DNA Binding Specificity Predictions with Structural Models. *Nucleic Acids Res.* **2005**, *33*, 5781−5798.

(11) Gao, M.; Skolnick, J. DBD-Hunter: A Knowledge-Based Method for the Prediction of DNA-Protein Interactions. *Nucleic Acids Res.* **2008**, *36*, 3978−3992.

(12) Tsuchiya, Y.; Kinoshita, K.; Nakamura, H. PreDs: A Server for Predicting dsDNA-Binding Site on Protein Molecular Surfaces. *Bioinformatics* **2005**, *21*, 1721−1723.

(13) Gao, M.; Skolnick, J. A Threading-Based Method for the Prediction of DNA-Binding Proteins with Application to the Human Genome. *PLoS Comput. Biol.* **2009**, *5*, No. e1000567.

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

(14) Chen, Y. C.; Wright, J. D.; Lim, C. DR_bind: A Web Server for Predicting DNA-Binding Residues from the Protein Structure Based on Electrostatics, Evolution and Geometry. *Nucleic Acids Res.* **2012**, *40*, W249−W256.
(15) Li, T.; Li, Q. Z.; Liu, S.; Fan, G. L.; Zuo, Y. C.; Peng, Y. PreDNA: Accurate Prediction of DNA-Binding Sites in Proteins by Integrating Sequence and Geometric Structure Information. *Bioinformatics* **2013**, *29*, 678−685.
(16) Wang, L.; Yang, M. Q.; Yang, J. Y. Prediction of DNA-Binding Residues from Protein Sequence Information Using Random Forests. *BMC Genomics* **2009**, *10* (Suppl 1), S1.
(17) Si, J.; Zhang, Z.; Lin, B.; Schroeder, M.; Huang, B. MetaDBSite: A Meta Approach to Improve Protein DNA-Binding Sites Prediction. *BMC Syst. Biol.* **2011**, *5*, S7.
(18) Ma, X.; Guo, J.; Liu, H. D.; Xie, J. M.; Sun, X. Sequence-Based Prediction of DNA-Binding Residues in Proteins with Conservation and Correlation Information. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2012**, *9*, 1766−1775.
(19) Ofran, Y.; Mysore, V.; Rost, B. Prediction of DNA-Binding Residues from Sequence. *Bioinformatics* **2007**, *23*, i347−i353.
(20) Hwang, S.; Gou, Z.; Kuznetsov, I. B. DP-Bind: A Web Server for Sequence-Based Prediction of DNA-Binding Residues in DNA-Binding Proteins. *Bioinformatics* **2007**, *23*, 634−636.
(21) Wu, J.; Liu, H.; Duan, X.; Ding, Y.; Wu, H.; Bai, Y.; Sun, X. Prediction of DNA-Binding Residues in Proteins from Amino Acid Sequences Using A Random Forest Model with a Hybrid Feature. *Bioinformatics* **2009**, *25*, 30−35.
(22) Yan, J.; Kurgan, L. DRNApred, Fast Sequence-Based Method that Accurately Predicts and Discriminates DNA- and RNA-Binding Residues. *Nucleic Acids Res.* **2017**, *45*, e84.
(23) Chawla, N. V.; Japkowicz, N.; Kotcz, A. Editorial. Special Issue on Learning from Imbalanced Data Sets. *Acm Sigkdd Explorations Newsletter.* **2004**, *6*, 1−6.
(24) Kotsiantis, S. B.; Kanellopoulos, D.; Pintelas, P. E. Handling Imbalanced Datasets: A Review. *GESTS International Transactions on Computer Science and Engineering.* **2006**, *30*, 25−36.
(25) He, H.; Garcia, E. A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263−1284.
(26) Liu, Y.; Yu, X.; Huang, J. X.; An, A. Combining Integrated Sampling with SVM Ensembles for Learning from Imbalanced Datasets. *Inf. Process. Manage.* **2011**, *47*, 617−631.
(27) Ertekin, S.; Huang, J.; Bottou, L.; Giles, L. Learning on the Border: Active Learning in Imbalanced Data Classification. In the *Sixteenth ACM Conference on Conference on Information and Knowledge Management*, 2007, pp 127−136.
(28) Ertekin, S.; Huang, J.; Giles, C. L. Active Learning for Class Imbalance Problem. In the *30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp 823−824.
(29) Yu, D. J.; Hu, J.; Tang, Z. M.; Shen, H. B.; Yang, J.; Yang, J. Y. Improving Protein-ATP Binding Residues Prediction by Boosting SVMs with Random Under-Sampling. *Neurocomputing.* **2013**, *104*, 180−190.
(30) Chen, P.; Huang, J. Z.; Gao, X. LigandRFs: Random Forest Ensemble to Identify Ligand-Binding Residues from Sequence Information Alone. *BMC Bioinf.* **2014**, *15*, S4.
(31) Yu, H.; Ni, J.; Zhao, J. ACOSampling: An Ant Colony Optimization-Based Undersampling Method for Classifying Imbalanced DNA Microarray Data. *Neurocomputing.* **2013**, *101*, 309−318.
(32) Galar, M.; Barrenechea, E.; Herrera, F.; Fernandez, A. EUSBoost: Enhancing Ensembles for Highly Imbalanced Data-Sets by Evolutionary Undersampling. *Pattern Recognit.* **2013**, *46*, 3460−3471.
(33) Yen, S. J.; Lee, Y. S. Cluster-Based Under-Sampling Approaches for Imbalanced Data Distributions. *Expert Syst. Appl.* **2009**, *36*, 5718−5727.
(34) García, S.; Herrera, F. Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy. *Evol. Comput.* **2009**, *17*, 275−306.
(35) Tang, Y.; Zhang, Y. Q.; Chawla, N. V.; Krasser, S. SVMs Modeling for Highly Imbalanced Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics).* **2009**, *39*, 281−288.
(36) Tang, Y.; Zhang, Y. Q. Granular SVM with Repetitive Undersampling for Highly Imbalanced Protein Homology Prediction. In *IEEE International Conference on Granular Computing*, 2006, pp 457−460.
(37) Pedrycz, W. Granular Computing: An Introduction. In *Ifsa World Congress and Nafips International Conference*, 2001. Joint, 2002, pp 1349−1354.
(38) Pedrycz, W.; Skowron, A.; Kreinovich, V., *Handbook of Granular Computing*. Wiley-Interscience: 2008, pp 719−740.
(39) Hu, J.; Li, Y.; Zhang, M.; Yang, X.; Shen, H. B.; Yu, D. J. Predicting Protein-DNA Binding Residues by Weightedly Combining Sequence-Based Features and Boosting Multiple SVMs. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2017**, *14*, 1389−1398.
(40) Rose, P. W.; Prlić, A.; Bi, C.; Bluhm, W. F.; Christie, C. H.; Dutta, S.; Green, R. K.; Goodsell, D. S.; Westbrook, J. D.; Woo, J.; et al. The RCSB Protein Data Bank: Views of Structural Biology for Basic and Applied Research and Education. *Nucleic Acids Res.* **2015**, *43*, D345−D356.
(41) Li, W.; Godzik, A. Cd-Hit: A Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. *Bioinformatics* **2006**, *22*, 1658−1659.
(42) Yu, D. J.; Hu, J.; Yang, J.; Shen, H. B.; Tang, J.; Yang, J. Y. Designing Template-Free Predictor for Targeting Protein-Ligand Binding Sites with Classifier Ensemble and Spatial Clustering. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2013**, *10*, 994−1008.
(43) Yang, J.; Roy, A.; Zhang, Y. BioLiP: A Semi-Manually Curated Database for Biologically Relevant Ligand-Protein Interactions. *Nucleic Acids Res.* **2012**, *41*, D1096−D1103.
(44) Wang, G.; Dunbrack, R. L., Jr PISCES: A Protein Sequence Culling Server. *Bioinformatics* **2003**, *19*, 1589−1591.
(45) Chen, Z.; Zhao, P.; Li, F.; Leier, A.; Marquez-Lago, T. T.; Wang, Y.; Webb, G. I.; Smith, A. I.; Daly, R. J.; Chou, K.-C.; Song, J. iFeature: A Python Package and Web Server for Features Extraction and Selection from Protein and Peptide Sequences. *Bioinformatics* **2018**, *34*, 2499−2502.
(46) Schäffer, A. A.; Aravind, L.; Madden, T. L.; Shavirin, S.; Spouge, J. L.; Wolf, Y. I.; Koonin, E. V.; Altschul, S. F. Improving the Accuracy of PSI-BLAST Protein Database Searches with Composition-Based Statistics and Other Refinements. *Nucleic Acids Res.* **2001**, *29*, 2994−3005.
(47) Bairoch, A.; Apweiler, R. The SWISS-PROT Protein Sequence Database and Its Supplement TrEMBL in 2000. *Nucleic Acids Res.* **2000**, *28*, 45−48.
(48) Jones, D. T. Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices. *J. Mol. Biol.* **1999**, *292*, 195−202.
(49) Joo, K.; Lee, S. J.; Lee, J. SANN: Solvent Accessibility Prediction of Proteins by Nearest Neighbor Method. *Proteins: Struct., Funct., Genet.* **2012**, *80*, 1791−1797.
(50) Henrich, S.; Saloahen, O. M.; Huang, B.; Rippmann, F. F.; Cruciani, G.; Wade, R. C. Computational Approaches to Identifying and Characterizing Protein Binding Sites for Ligand Design. *J. Mol. Recognit.* **2009**, *23*, 209−219.
(51) Gromiha, M. M. Development of RNA Stiffness Parameters and Analysis on Protein-RNA Binding Specificity: Comparison with DNA. *Curr. Bioinf.* **2012**, *7*, 173−179.
(52) Mishra, N. K.; Chauhan, J. S.; Raghava Gajendra, P. S. Identification of ATP Binding Residues of a Protein from Its Primary Sequence. *BMC Bioinf.* **2009**, *10*, 434−440.
(53) Freund, Y.; Schapire, R. E. Experiments with A New Boosting Algorithm. In *International Conference on Machine Learning*, 1996, Vol. *13*, pp 148−156.
(54) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071

(55) Song, J.; Tan, H.; Shen, H.; Mahmood, K.; Boyd, S. E.; Webb, G. I.; Akutsu, T.; Whisstock, J. C. Cascleave: Towards More Accurate Prediction of Caspase Substrate Cleavage Sites. *Bioinformatics* **2010**, *26*, 752−760.
(56) Wang, M.; Zhao, X. M.; Tan, H.; Akutsu, T.; Whisstock, J. C.; Song, J. Cascleave 2.0, A New Approach for Predicting Caspase and Granzyme Cleavage Targets. *Bioinformatics* **2014**, *30*, 71−80.
(57) Li, F.; Li, C.; Wang, M.; Webb, G. I.; Zhang, Y.; Whisstock, J. C.; Song, J. GlycoMine: A Machine Learning-Based Approach for Predicting N-, C- and O-Linked Glycosylation in the Human Proteome. *Bioinformatics* **2015**, *31*, 1411−1419.
(58) Li, F.; Li, C.; Revote, J.; Zhang, Y.; Webb, G. I.; Li, J.; Song, J.; Lithgow, T. GlycoMine(Struct): A New Bioinformatics Tool for Highly Accurate Mapping of the Human N-Linked and O-Linked Glycoproteomes by Incorporating Structural Features. *Sci. Rep.* **2016**, *6*, 34595.
(59) Chen, Z.; Liu, X.; Li, F.; Li, C.; Marquez-Lago, T.; Leier, A.; Akutsu, T.; Webb, G. I.; Xu, D.; Smith, A. I.; Li, L.; Chou, K.-C.; Song, J. Large-Scale Comparative Assessment of Computational Predictors for Lysine Post-Translational Modification Sites. *Briefings Bioinf.* **2018**, bby089−bby089.
(60) Li, F.; Li, C.; Marquez-Lago, T. T.; Leier, A.; Akutsu, T.; Purcell, A. W.; Smith, A. I.; Lithgow, T.; Daly, R. J.; Song, J.; Chou, K.-C. Quokka: A Comprehensive Tool for Rapid and Accurate Prediction of Kinase Family-Specific Phosphorylation Sites in the Human Proteome. *Bioinformatics* **2018**, *34*, 4223.
(61) Li, F.; Wang, Y.; Li, C.; Marquez-Lago, T. T.; Leier, A.; Rawlings, N. D.; Haffari, G.; Revote, J.; Akutsu, T.; Chou, K.-C.; Purcell, A. W.; Pike, R. N.; Webb, G. I.; Ian Smith, A.; Lithgow, T.; Daly, R. J.; Whisstock, J. C.; Song, J. Twenty Years of Bioinformatics Research for Protease-Specific Substrate and Cleavage Site Prediction: A Comprehensive Revisit and Benchmarking of Existing Methods. *Briefings Bioinf.* **2018**, bby077−bby077.
(62) Song, J.; Li, F.; Leier, A.; Marquez-Lago, T. T.; Akutsu, T.; Haffari, G.; Chou, K. C.; Webb, G. I.; Pike, R. N.; Hancock, J. PROSPERous: High-Throughput Prediction of Substrate Cleavage Sites for 90 Proteases with Improved Accuracy. *Bioinformatics* **2018**, *34*, 684−687.
(63) Song, J.; Wang, Y.; Li, F.; Akutsu, T.; Rawlings, N. D.; Webb, G. I.; Chou, K.-C. iProt-Sub: A Comprehensive Package for Accurately Mapping and Predicting Protease-Specific Substrates and Cleavage Sites. *Briefings Bioinf.* **2018**, bby028−bby028.
(64) Hu, J.; Zhou, X.; Zhu, Y.; Yu, D.; Zhang, G. TargetDBP: Accurate DNA-Binding Protein Prediction via Sequence-based Multi-View Feature Learning. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2019**, 1−1.
(65) Song, J.; Li, F.; Takemoto, K.; Haffari, G.; Akutsu, T.; Chou, K. C.; Webb, G. I. PREvaIL, An Integrative Approach for Inferring Catalytic Residues Using Sequence, Structural, and Network Features in a Machine-Learning Framework. *J. Theor. Biol.* **2018**, *443*, 125−137.
(66) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012, pp 1097−1105.
(67) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation.* **1997**, *9*, 1735−1780.
(68) Ding, Y.; Tang, J.; Guo, F. Identification of Protein-Ligand Binding Sites by Sequence Information and Ensemble Classifier. *J. Chem. Inf. Model.* **2017**, *57*, 3149−3161.
(69) Ahmad, S.; Gromiha, M. M.; Sarai, A. Analysis and Prediction of DNA-Binding Proteins and Their Binding Residues Based on Composition, Sequence and Structural Information. *Bioinformatics* **2004**, *20*, 477−486.
(70) Wang, L.; Brown, S. J. BindN: A Web-Based Tool for Efficient Prediction of DNA and RNA Binding Sites in Amino Acid Sequences. *Nucleic Acids Res.* **2006**, *34*, W243−W248.
(71) Yan, C.; Terribilini, M.; Wu, F.; Jernigan, R. L.; Dobbs, D.; Honavar, V. Predicting DNA-Binding Sites of Proteins from Amino Acid Sequence. *BMC Bioinf.* **2006**, *7*, 262−267.
(72) Chu, W. Y.; Huang, Y. F.; Huang, C. C.; Cheng, Y. S.; Huang, C. K.; Oyang, Y. J. ProteDNA: A Sequence-Based Predictor of Sequence-Specific DNA-Binding Residues in Transcription Factors. *Nucleic Acids Res.* **2009**, *37*, W396−W401.
(73) Wang, L.; Huang, C.; Yang, M. Q.; Yang, J. Y. BindN+ for Accurate Prediction of DNA and RNA-Binding Residues from Protein Sequence Features. *BMC Syst. Biol.* **2010**, *4*, S3.
(74) Liu, R.; Hu, J. DNABind: A Hybrid Algorithm for Structure-Based Prediction of DNA-Binding Residues by Combining Machine Learning- and Template-Based Approaches. *Proteins: Struct., Funct., Genet.* **2013**, *81*, 1885−1899.
(75) Delano, W. L. The PyMOL User's Manual. *DPSM for Modeling Engineering Problems* **2002**, *4*, 148−149.
(76) Yang, J.; Yang, J. Y.; Zhang, D.; Lu, J. F. Feature Fusion: Parallel Strategy vs. Serial Strategy. *Pattern Recognit.* **2003**, *36*, 1369−1381.
(77) Phinyomark, A.; Phukpattaranont, P.; Limsakul, C. Feature Reduction and Selection for EMG Signal Classification. *Expert Syst. Appl.* **2012**, *39*, 7420−7431.
(78) Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157−1182.
(79) Ren, H.; Shen, Y. RNA-Binding Residues Prediction Using Structural Features. *BMC Bioinf.* **2015**, *16*, 249.
(80) Chen, K.; Mizianty, M. J.; Kurgan, L. ATPsite: Sequence-Based Prediction of ATP-Binding Residues. *Proteome Sci.* **2011**, *9* (Suppl 1), S4.

NOTE ADDED AFTER ASAP PUBLICATION

Due to a production error, this paper was published on the Web on April 16, 2019, with incorrect defining values for F, F_1, and F_2 on the third page of the document. The corrected version was reposted on April 17, 2019.

DOI: 10.1021/acs.jcim.8b00749
J. Chem. Inf. Model. 2019, 59, 3057−3071