Subject Section

StackDPPred: A Stacking based Prediction of DNA-binding Protein from Sequence

Avdesh Mishra[1,§], Pujan Pokhrel[1,§] and Md Tamjidul Hoque[1,*]

[1]Department of Computer Science, University of New Orleans, New Orleans, LA, 70148, USA.

[§]These authors contributed equally to this work as first authors.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Identification of DNA-binding proteins from only sequence information is one of the most challenging problems in the field of genome annotation. DNA-binding proteins play an important role in various biological processes such as DNA replication, repair, transcription and splicing. Existing experimental techniques for identifying DNA-binding proteins are time-consuming and expensive. Thus, prediction of DNA-binding proteins from sequences alone using computational methods can be useful to quickly annotate and guide the experimental process. Most of the methods developed for predicting DNA-binding proteins use the information from the evolutionary profile, called the position-specific scoring matrix (PSSM) profile, alone and the accuracies of such methods have been limited. Here, we propose a method, called StackDPPred, which utilizes features extracted from PSSM and residue specific contact-energy to help train a stacking based machine learning method for the effective prediction of DNA-binding proteins.

Results: Based on benchmark sequences of 1063 (518 DNA-binding and 545 non DNA-binding) proteins and using jackknife validation, StackDPPred achieved an ACC of 89.96%, MCC of 0.799 and AUC of 94.50%. This outcome outperforms several state-of-the-art approaches. Furthermore, when tested on recently designed two independent test datasets, StackDPPred outperforms existing approaches consistently. The proposed StackDPPred can be used for effective prediction of DNA-binding proteins from sequence alone.

Availability: Online server is at http://bmll.cs.uno.edu/add and code-data is at http://cs.uno.edu/~tamjid/Software/StackDPPred/code_data.zip.

Contact: thoque@uno.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

DNA-binding proteins carry out many crucial intercellular and intracellular functions such as DNA replication and repair, transcriptional regulation, the combination and separation of single-stranded DNA and other biological activities associated with DNA. In the eukaryotic cells, histone, which is a usual type of DNA-binding protein, often helps in packaging chromosomal DNA into a compact structure. Similarly, DNA-binding proteins, such as restriction enzymes, are DNA-cutting enzymes, which are found in bacteria that recognize and cut DNA only at a particular sequence of nucleotides to serve a host-defense role. DNA-binding proteins represent a broad category of proteins, which are found to be highly diverse in structure as well as in sequence. Structurally, they have been divided into 8 structural groups, which are further classified into 54 structural families (Lin et al. 2013; Luscombe et al. 2000). Additionally, it has been reported that 2 to 3% of a prokaryotic genomes and 6 to 7% of a eukaryotic genomes encode DNA-binding proteins (Luscombe et al. 2000; Walter et al. 2008). Moreover, the past decade has witnessed tremendous progress in genome sequencing (Harris et al. 2008; Wheeler et al. 2008). According to the genome online database, the complete sequenced genomes of almost 1000 cellular organisms have been released, and about 5000 active genome sequencing projects are on the way

(Liolios et al. 2006; Q. Zou et al. 2014). The unprecedented amount of genetic information has provided a plethora of protein sequences (Wu et al. 2006), posing a challenging problem of annotating them and elucidating their functions. Thus, the task of identifying DNA-binding proteins has become crucial. This motivated us to develop a predictor for effective identification and characterization of DNA-binding proteins from sequence.

In the early days, the identification of DNA-binding proteins was carried out by experimental techniques, including filter binding assays, genetic analysis, chromatin immunoprecipitation on microarrays and X-ray crystallography. However, it is both time-consuming and expensive to identify DNA-binding proteins purely based on biochemical experiments. Particularly, given the abundance of biological sequences generated in the post-genomic era. Hence, it is important to develop computational methods for fast and effective identification of DNA-binding proteins. Therefore, the fundamental objective of this work is to develop a computational tool for the rapid and effective identification and annotation of DNA-binding proteins from sequence.

The computational methods refer to a wide range of approaches that capture various information such as structural, sequence and other physicochemical properties. Many attempts have been made in identifying DNA-binding proteins and many effective computational prediction methods have been proposed in the literature for analyzing them. The computational approaches for the identification of DNA-binding proteins can be divided into two broad categories: *i)* template based; and *ii)* machine learning based. Template-based methods detects significant structural or sequence similarity between the query and a template known to bind DNA, to assess the DNA-binding preference of the target sequence (Gao and Skolnick 2008; Shanahan et al. 2004). Unlike template-based methods, machine learning methods learn the relevant predictive model to make predictions by finding a pattern in the input feature space for example, support vector machine (SVM) (Bhardwaj et al. 2005; Brown and Akutsu 2009; Huang et al. 2011; Kumar et al. 2007; Xiong et al. 2011), neural network (Ahmad and Sarai 2004; Andrabi et al. 2009; Stawiski et al. 2003; Wei et al. 2014), random forest (Nimrod et al. 2010), naïve Bayes classifier (Govindan and Nair 2011; Yan et al. 2006), nearest neighbors algorithm (Qian et al. 2006) and ensemble classifiers (Nanni and Lumini 2008; Xia et al. 2010; Q. Zou et al. 2013b).

The task of predicting DNA-binding proteins using machine learning, involves two important steps: *i*) extraction of relevant features; and *ii*) selection of an appropriate classification algorithm. Depending on the feature extraction mechanism, the existing predictive methods can be classified into two different categories: *i*) extraction of relevant features from the structure of the protein (Iqbal and Hoque 2015; Liu et al. 2008; Liu et al. 2012; Tjong and Zhou 2007); and *ii*) extraction of relevant features from the sequence of amino acids (Feng and Zhang 2000; Moroni et al. 2007; R. Sharma et al. 2018; Wang et al. 2005; Q. Zou et al. 2014). According to Xu *et al*. (Xu et al. 2015), the accuracy of structure-based prediction methods for DNA-binding proteins is usually higher but, those cannot be used in high throughput annotation because those require a high-resolution 3D structure of the protein sequence. Thus, to date, various computational methods have been proposed for identifying DNA-binding proteins directly from their amino acid sequences. These methods independently explore four different categories of protein sequence features and four different kinds of sequence encoding methods (Huang et al. 2011; Lou et al. 2014; Vapnik 1999; Xu et al. 2015; L. Zhang et al. 2014). Specifically, the four different categories of features include: (a) composition information; (b) structural and functional information; (c) physicochemical properties; and (d) evolutionary information. The four different kinds of encoding methods are: (a) overall composition-transition-distribution called OCTD (global method); (b) autocross-covariance (ACC) transformation (nonlocal method); (c) split amino acid (SAA) transformation (local method); and (d) PSSM distance transformation called PSSM-DT. These methods have been studied in detail in the relevant research work (Nanni et al. 2010; Xu et al. 2015; Z. Zhang et al. 2005; C. Zou et al. 2013a). However, most of the proposed methods are limited in their ability to explain how protein-DNA interactions occur. Thus, it is essential to identify new features, effective encoding techniques and advanced machine learning techniques that can help further improve our understanding of DNA-protein interactions with higher accuracy.

In this study, we explore different features, encoding techniques and machine learning approaches, to further improve the prediction accuracy and understand the binding mechanism of DNA-protein interaction. We propose a method, StackDPPred, which utilizes two different types of features: PSSM profile and residue wise contact energy profile. It uses four different types of feature transformation techniques: PSSM-DT, residue probing transformation (Dosztányi et al. 2005), evolutionary distance transformation (L. Zhang et al. 2014), residue wise contact energy matrix transformation (L. Zhang et al. 2014), and a machine learning ensembles, known as stacking (Wolpert 1992). The development of StackDPPred offered a significant improvement in prediction accuracies based on the benchmark and independent test data when compared to several state-of-the-art predictors. We believe that the superior performance of StackDPPred will motivate the researchers to use this method to identify DNA-binding proteins from sequence. In addition, the proposed stacking based machine learning technique and features discussed in this work could be applied to solve various other biologically important problems.

2 Methods

In this section, we describe the procedure for benchmark and independent test data preparation, feature extraction and encoding, performance evaluation metrics and machine learning framework development.

2.1 Dataset

We used the benchmark dataset (Xu et al. 2015) that contains 525 DNA-binding protein sequences and 550 non DNA-binding protein sequences. In our implementation, we only used 518 DNA-binding proteins and 545 non DNA-binding proteins as we were unable to obtain the PSSM profile of 7 DNA-binding proteins and 5 non DNA-binding proteins using PSI-BLAST (Altschul et al. 1997).

We used an independent test dataset PDB186 (Lou et al. 2014) to compare the performance of StackDPPred with several state-of-the-art approaches. This dataset consists of 93 DNA-binding proteins and 93 non DNA-binding proteins selected from the PDB to validate the quality of predictions.

In addition, to further test the robustness of our predictor, we collected an additional independent test dataset, called PDB676. This dataset consist of 338 DNA-binding proteins and 338 non DNA-binding proteins. The dataset was obtained according to the following procedure: *i)* extract all the DNA-binding protein sequences from PDB by searching the mmCIF keyword of "contains DNA binding protein"; *ii)* remove the sequences with length of < 60 amino acids and character of "X"; *iii)* utilize BLASTCLUST (Camacho et al. 2009) to cutoff those sequences that have ≥ 25% sequence similarity to any other in the same subset as well as in the benchmark dataset; *iv)* filter proteins already present in the PDB186 dataset; *v)* randomly extract non DNA-binding proteins from PDB using mmCIF keyword "does not contain DNA binding protein" and apply same procedure as for DNA-binding proteins to filter out the sequences with < 60 amino acids, character of “X” and having ≥ 25% sequence similarity to any other in the same subset.

2.2 Feature Extraction

To create an effective machine learning method to predict DNA-binding proteins from sequence alone, we use various features derived from the Position Specific Scoring Matrix (PSSM) and the Reside Wise Contact Energy Matrix (RCEM), described next.

2.2.1 Position Specific Scoring Matrix (PSSM) feature

PSSM captures the conservation pattern in multiple alignments and stores it as a matrix of scores for each position in the alignment. High scores represent more conserved positions and scores close to zero or negative, represent weakly conserved positions. Thus, PSSM captures the evolutionary information in proteins. Evolutionary information is one of the most important kinds of information for protein functionality annotation in biological analysis and is widely used in many studies (Biswas et al. 2010; Iqbal and Hoque 2015; Iqbal et al. 2015; Islam et al. 2016; Kumar et al. 2007; R. Verma et al. 2010). For this study, the PSSM profile of every protein sequence is obtained by executing three iteration of PSI-BLAST against NCBI’s non-redundant database (Altschul et al. 1997). The PSSM profile is a matrix of L*20 dimensions, which can be represented as follows:

$$PSSM = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,20} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ P_{L,1} & P_{L,2} & \ldots & P_{L,20} \end{bmatrix} \qquad (1)$$

where L is the length of the protein and $P_{i,j}$ represents the occurrence probability of amino acid i at position j of the protein sequence, the rows represent the position of amino acid in the sequence and the columns represent the 20 standard amino acid types. The large positive scores indicate conserved positions, which in turn implies critical functional residues that are required to perform various intermolecular interactions.

2.2.2 Reside Wise Contact Energy Matrix (RCEM) feature

Structural stability of proteins is the result of a large number of inter and intra-residual interactions. The energy contribution of these interactions can be approximated by the energy functions extracted from known structures (Hoque et al. 2016; Iqbal and Hoque 2015; Iqbal et al. 2015; Iqbal and Hoque 2016; Mishra et al. 2016; Mishra and Hoque 2017; Tarafder et al. 2018; Zhou and Skolnick 2011). However, the energy functions require that the 3D structure be known in order to compute the summation of such energies. Thus, they are not applicable for proteins whose structure is not known or for intrinsically disordered proteins (IDPs) (Babu et al. 2016). Moreover, it was found that IDPs are very common in DNA-binding proteins, particularly in disordered tails (Dosztányi et al. 2005; Vuzman et al. 2012). Thus, to inherently incorporate important information regarding the amino acid interactions and intrinsically disordered regions (IDRs), we employed the predicted residue wise contact energies. These contact energies are derived in (Dosztányi et al. 2005), using 674 protein’s primary sequences by the least square fitting with the contact energies derived from tertiary structure of 785 proteins. The RCEM is a 20 × 20 dimensional matrix whose rows and columns represent 20 standard amino acids. Supplementary Table 1S. shows RCEM table (Dosztányi et al. 2005) that we have applied in this work.

2.3 Feature Extraction from PSSM profile

In this section, we describe various feature extraction techniques we utilized to obtain a fixed dimensional feature vector from the PSSM profile to encode protein sequences.

2.3.1 PSSM-Distance Transformation (PSSM-DT) feature

It has been demonstrated in the literature that two amino acids either consecutive or separated by a distance along the sequence e.g. profile-bigram or k-separated bigrams are important for protein fold recognition, protein functionality annotation, protein subcellular localization, structural class prediction, drug-interaction and other related problems (Liu et al. 2014; Saini et al. 2014; Saini et al. 2015; A. Sharma et al. 2013). We use two forms of PSSM distance transformation techniques (Xu et al. 2015) to transform the PSSM information into fixed dimensional vectors. Given a PSSM (Equation (1)) of protein sequences, two distance transformation schemes: *i*) the PSSM distance transformation for pairs of same amino acids (PSSM-SDT); and *ii*) the PSSM distance transformation for pairs of different amino acids (PSSM-DDT) were used to extract fixed sized feature vectors.

PSSM-SDT features approximately measure the occurrence probabilities for the pairs of the same amino acids separated by a distance *d* along the sequence, which can be calculated as:

$$PSSM\text{-}SDT(j,d) = \sum_{i=1}^{L-d} P_{i,j} * P_{i+d,j}/(L-d) \qquad (2)$$

where, *j* is one type of the amino acid, *L* is the length of the sequence, $P_{i,j}$ is the PSSM score of amino acid *j* at position *i,* and $P_{i+d,,j}$ is the PSSM score of amino acid *j* at position *i+d*. Through this approach, 20**D* number of PSSM-SDT features were generated, where *D* is the maximum range of *d* (*d* = 1, 2,…, *D*).

Similarly, PSSM-DDT calculates the occurrence probabilities for pairs of different amino acids separated by a distance of d along the sequence, which can be calculated as:

$$PSSM\text{-}DDT(i_1,i_2,d) = \sum_{j=1}^{L-d} P_{j,i_1} * P_{j+d,\ i_2}/(L-d) \qquad (3)$$

where, i_1 and i_2 represent two different types of amino acids. The total number of features obtained by PSSM-DDT are *380*D*. Furthermore, *D* = 5, was found (Xu et al. 2015) to perform the best. As we are using the same benchmark dataset, we use the same value of *D* in our experiment.

2.3.2 Residue Probing Transformation (RPT) feature

RPT, proposed by Jeong *et al.* (Jeong et al. 2011), emphasizes domains with similar conservation rates by grouping domain families based on their conservation score in the PSSM profile. Here, each probe is a standard amino acid, and corresponds to a particular column in the PSSM profile. The rows in the PSSM profile are divided into 20 groups according to 20 different standard amino acids. Then, for each group, the sum of the PSSM values in every column are computed, leading to a feature vector of a 20×20 dimensional matrix, called RPT matrix, represented as in Equation (4):

$$RPT = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,20} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ S_{20,1} & S_{20,2} & \ldots & S_{20,20} \end{bmatrix} \qquad (4)$$

The RPT matrix (Equation (4)) is then transferred into a feature vector of 400 dimensions, as shown in Equation (5):

$$V = [f_{s_{1,1}}, f_{s_{1,2}}, \cdots, f_{s_{i,j}}, \cdots, f_{s_{20,20}}] \tag{5}$$

where, $f_{s_{i,j}}$ is calculated using Equation (6):

$$f_{s_{i,j}} = \frac{s_{i,j}}{L}\ (i, j = 1,2, \cdots, 20) \tag{6}$$

2.3.3. Evolutionary Distance Transformation (EDT) feature

The EDT extracts the information of the non-co-occurrence probability for two amino acids separated by a certain distance d in a protein from the PSSM profile (L. Zhang et al. 2014). Here, d is the distance between these two amino acids in a sequence $(d = 1, 2, \ldots, L_{min}\text{-}1)$, where L_{min} is the length of the shortest protein in the benchmark dataset. For example, $d = 1$ implies that the two amino acids are consecutive; $d = 2$ implies that there is one amino acid between the two, and so on. The EDT feature vector computed from the PSSM profile can be represented as (7):

$$P = [\partial_1, \partial_2, \cdots, \partial_\Omega] \tag{7}$$

where, Ω is an integer that represents the dimension of the vector whose value is 400. The non-co-occurrence probability of two amino acids separated by distance d (here, d ranges from 1 to D, where, D is the maximum value of d) can be computed using Equation (8):

$$f(A_x, A_y) = \sum_{d=1}^{D} \frac{1}{L-d} \sum_{i=1}^{L-d} (P_{i,x} - P_{i+d,y})^2 \tag{8}$$

where, $P_{i,x}$ and $P_{i+d,y}$ are the elements in the PSSM profile; A_x and A_y represent any of the 20 different amino acids in the protein. Finally, each element in feature vector P is obtained as given in Equation (9):

$$\begin{gathered} \{\partial_1 = f(A_1, A_1)\} \\ \{\partial_2 = f(A_2, A_2)\} \\ \cdots \\ \{\partial_{400} = f(A_{20}, A_{20})\} \end{gathered} \tag{9}$$

2.4 Feature extracted from RCEM

Depending on the type of amino acid, a 20-dimensional vector for each amino acid in a sequence is obtained from the rows of RCEM (see supplementary Table 1S). Thus, for a protein sequence of length L, an L × 20-dimensional matrix E_m is obtained. Next, the matrix E_m is transformed into a feature vector of 20 dimensions by calculating the column-wise sum. If the element of matrix E_m is represented by $e_{i,j}$, where i represents the amino acid index in a sequence (rows) and j represents 20 standard amino acid types (columns), the column-wise sum of matrix E_m can be obtained using Equation (10):

$$f(A_j) = \sum_{i=1}^{L} e_{i,j}\ (j = 1,2, \cdots, 20) \tag{10}$$

Then, the final feature vector, RCEM-Transformation $(RCEMT) = [E_1, E_2, \ldots, E_{20}]$ is obtained by dividing each element in RCEMT by the total sum of the elements in the same vector, which can be represented as shown in Equation (11):

$$RCEMT(E_i) = \frac{E_i}{L} \tag{11}$$

2.5 Performance Evaluation

The performance of StackDPPred is measured by the jackknife validation approach. In jackknife validation, every sample is tested by the predictor trained with all the other samples in the benchmark set. We employ various performance evaluation measures listed in the supplementary Table 2S to test the predictive capacity of our proposed method as well as to compare it with several state-of-the-art methods.

Moreover, we use AUC and ROC performance measures. AUC is the area under the receiver operating characteristics (ROC) curve and is commonly used to evaluate a predictor to see how well it separates two classes of information, i.e., in this case, DNA-binding versus non DNA-binding proteins.

2.6 Framework of StackDPPred

We applied a stacking technique (Wolpert 1992) to develop the StackDPPred predictor for DNA-binding proteins. Stacking is an ensemble technique used to combine information from multiple predictive models to generate a new model. It provides a scheme for minimizing the generalization error rate of one or more predictive models and has been successfully applied in several machine learning tasks (Frank et al. 2004; Hu et al. 2015; Nagi and Bhattacharyya 2013; A. Verma and Mehta 2017), http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/.

Stacking framework involves two-stages of learning. The classifiers of the first-stage and second-stage are called base-classifier and meta-classifier respectively. A pool of base-classifiers are employed in the first-stage. Then, using meta-classifier in the second-stage, the outputs of the base-classifiers are combined with the aim of reducing the generalization error. To enrich the meta-classifier with more information on the solution space, it is desirable to use classifiers that are different from each other based on their underlying operating principle as the base-classifiers.

In order to find the base-classifiers to use in the first-stage and the meta-classifier to use in the second-stage of stacking framework, we explored six different machine learning algorithms: (a) Support Vector Machine (SVM) (Vapnik 1999), (b) Logistic Regression (LogReg) (Hastie et al. 2009; Szilágyi and Skolnick 2006), (c) Extra Trees (ET) (Geurts et al. 2006), (d) Random Decision Forest (RDF) (Ho 1995), (e) K Nearest Neighbor (KNN) (Altman 1992) and (f) Bagging (BAG) (Breiman 1996). The algorithms and their configuration details are briefly discussed in the supplementary document.

All of the above mentioned classifiers are built and tuned using scikit-learn (Pedregosa et al. 2011). To select a pool of algorithms to be used as the base-classifiers for the stacked model (SM), we evaluate three different combinations of base-classifier. The three different combinations of base-classifiers formed and tested in this study are:

i) **SM1:** includes SVM, LogReg, KNN and RDF,
ii) **SM2:** includes SVM, LogReg, KNN and ET, and
iii) **SM3:** includes SVM, LogReg, KNN and BAG.

As RDF, ET and BAG are tree based methods, we individually combined them with the other three methods, SVM, LogReg and KNN, whose underlying principles are different from each other. For all of the above combinations SM1, SM2 and SM3, the meta-level classifier is SVM. The jackknife validation of the above three combinations show that the SM1 when combined with SVM provides the best accuracy. Thus, we employ four classifiers SVM, LogReg, KNN and RDF as base-classifiers in the StackDPPred framework. The output probabilities (binding probability p and non-binding probability (1-p)) generated by the four base-classifiers are combined with the original 2,820 features and used as input features to train a new SVM classifier (meta-classifier) which is then used to obtain a robust StackDPPred classifier. The framework of StackDPPred is shown in supplementary Figure 1S.

3 Results

In this section, we first present the results of the potential base-classifiers for the development of StackDPPred. Then, we report the performance of StackDPPred on the benchmark dataset and the independent test dataset.

Selection of Base-classifiers

To select the best combination of classifiers to be used as the base-classifiers, we first analyze the performance of six different machine learning methods, SVM, LogReg, KNN, RDF, BAG and ET, on the benchmark dataset. The individual predictive ability of each of the classifiers is obtained using the jackknife validation approach and is shown in Table 1.

Table 1 highlights that the optimized SVM with RBF-kernel gives an outstanding jackknife validation accuracy compared to other methods in this application. The SVM with RBF-kernel gives the best recall (defined as proportion of real positive cases that are correctly predicted positive), specificity (defined as proportion of real negative cases that are correctly predicted negative), fall out rate (defined as proportion of real negative cases that occur as predicted positive), miss rate (defined as proportion of real positive cases that occur as predicted negative), balanced accuracy (defined as mean of recall and specificity), accuracy (defined as proportion of both positive cases and negative cases correctly predicted), precision (defined as proportion of predicted positive cases that are correctly real positives), F1 score (defined as harmonic mean between precision and recall) and MCC (measures the degree of overlap between the predicted labels and true labels of all the samples in the benchmark dataset). Similarly, based on all of the performance measures provided in Table 1, LogReg provides the second highest performance. As the learning principle of SVM and LogReg are different from each other and they are the two highest performing methods, we have selected them as two of the base-classifier initially. Next, out of the remaining four classifiers KNN, RDF, ET and BAG, we have selected KNN, which operates by learning from the K number of training samples closest in distance to the target point as the third base-classifier. Finally, we have selected one out of the three remaining ensemble based classifiers, RDF, ET and BAG, as the fourth base-classifier and formulated 3 models, namely SM1, SM2, and SM3.

Table 1. Comparisons of various base learners on the benchmark dataset using jackknife cross-validation.

Metric/Method	LogReg	ET	KNN	SVM	BAG	RDF
Sensitivity	0.7851	0.6737	0.6602	**0.8030**	0.6988	0.6853
Specificity	0.7559	0.7522	0.7945	**0.8055**	0.7596	0.7761
Fall out rate	0.2440	0.2477	0.2055	**0.1945**	0.2403	0.2238
Miss Rate	0.2142	0.3262	0.3397	**0.1969**	0.3011	0.3146
Bal. Accuracy	0.7708	0.7130	0.7273	**0.8043**	0.7292	0.7307
Accuracy	0.7705	0.7385	0.7291	**0.8043**	0.7422	0.7440
Precision	0.7537	0.7210	0.7533	**0.7969**	0.7342	0.7442
F1 score	0.7693	0.6966	0.7037	**0.8000**	0.7161	0.7135
MCC	0.5415	0.4276	0.4594	**0.6084**	0.4595	0.4637

Table 2. Comparisons of stacked models with different set of base-classifiers through jackknife validation.

Method / Metric	Sensitivity	Specificity	Fall out rate	Miss Rate	Bal. Accuracy	Accuracy	Precision	F1 score	MCC
SM1	0.911	0.888	0.111	0.088	0.899	0.899	0.898	0.899	0.799
SM2	0.899	0.884	0.115	0.100	0.892	0.891	0.880	0.890	0.783
SM3	0.901	0.882	0.117	0.098	0.892	0.891	0.879	0.890	0.783

Table 2 shows that the SM1, SM2 and SM3 stacked model categories provide similar performance. However, SM1, which includes SVM, LogReg, KNN and RDF as base-classifiers and another SVM as a meta-classifier provides the best performance. Thus, we select SM1 as our final predictor.

Performance Comparison on Benchmark Dataset

In this section, we compare the performance of StackDPPred with several state-of-the-art methods on the benchmark dataset. The quantities for all the evaluation metrics for several state-of-the-art methods, iDNA-Prot (Huang et al. 2011), DNAbinder (Kumar et al. 2007), DNA-Prot (Kandaswamy et al. 2011) and PSSM-DT (Xu et al. 2015) are obtained from Xu et al. (Xu et al. 2015).

Table 3. Comparisons of StackDPPred with other state-of-art methods on benchmark dataset through jackknife validation.

Method/Metric	ACC	Sensitivity	Specificity	MCC	AUC
PSSM-DT (imp. %)	0.7996 (12.50%)	0.8191 (11.24%)	0.7800 (13.9%)	0.6220 (28.5%)	0.8650 (11.56%)
iDNA-Prot (imp. %)	0.7540 (19.21%)	0.8381 (8.72%)	0.6473 (37.2%)	0.5000 (59.8%)	0.7610 (24.1%)
DNAbinder (imp. %)	0.7358 (22.26%)	0.6647 (37.08%)	0.8036 (10.5%)	0.4700 (70%)	0.8150 (15.95%)
DNA-Prot (imp. %)	0.7255 (24.0%)	0.8267 (10.22%)	0.5976 (48.6%)	0.4400 (81.5%)	0.7890 (19.8%)
StackDPPred (avg. imp. %)	**0.8996** **(19.5%)**	**0.9112** **(36.4%)**	**0.8880** **(27.6%)**	**0.7990** **(60.0%)**	**0.9449** **(17.8%)**

Here, 'imp.' stands for improvement. The 'imp. %' represents improvement in percentage achieved by StackDPPred over the respective method and 'avg. imp. %' represents average improvement in percentage achieved by StackDPPred over the listed methods.

From Table 3, we observed that StackDPPred performs higher than the listed state-of-the-art methods. Specifically, StackDPPred provides 19.5%, 36.4%, 27.6%, 60.0% and 17.8% improvement on average over PSSM-DT, iDNA-Prot, DNAbinder and DNA-Prot methods based on ACC, sensitivity, specificity, MCC and AUC respectively.

Furthermore, Figure 1 presents the ROC curves generated by StackDPPred and PSSM-DT while the predictions are evaluated through jackknife validation on the benchmark dataset. We only compare StackDPPred with PSSM-DT as it is the highest performing method among those listed in Table 3. The ROC curves show the TPR (sensitivity)/FPR (1-specificity) pairs at different classification thresholds. The curves highlight the strength of StackDPPred in achieving a high TPR of ≥ 85% (rate of correct prediction of DNA-binding proteins) at a very low FPR of ≤ 20%. Whereas, PSSM-DT provides TPR of ≥ 80% at a cost of higher FPR of ≥ 20%. Moreover, the AUC score given by StackDPPred is

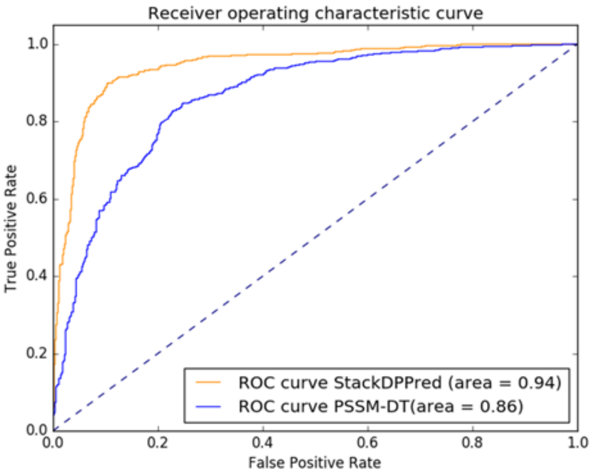


Fig. 1. Comparisons of ROC curves and AUC scores given by StackDPPred and PSSM-DT on benchmark dataset.

9.30% higher than that of PSSM-DT when evaluated on benchmark dataset. These results show that StackDPPred is a promising predictor.

Performance Comparison using Independent Test Datasets

In this section, the performance of StackDPPred is further compared with several state-of-the-art methods on two independent test datasets, PDB186 and PDB676. The PDB186 dataset was recently constructed by Lou *et al* (Lou et al. 2014) to validate the quality of DNA-binding predictions. It consists of 93 DNA-binding proteins and an equal number of non DNA-binding proteins. Some proteins on the benchmark dataset which share more than 25% sequence similarity to proteins in PDB186 dataset were removed using the program BLASTCLUST (Camacho et al. 2009) to remove homologous bias. Then, StackDPPred was first trained on the benchmark dataset and tested on PDB186 test set. Table 4 lists the predictive results of StackDPPred and several state-of-the-art methods including PSSM-DT (Xu et al. 2015), iDNA-Prot (Huang et al. 2011), DNA-Prot (Kandaswamy et al. 2011), DNAbinder (Kumar et al. 2007), DNA-BIND (Szilágyi and Skolnick 2006), DNA-Threader (Gao and Skolnick 2009) and DBPPred (Lou et al. 2014) on the PDB186 test set.

Table 4 indicates that based on the PDB186 test set, the StackDPPred outperforms PSSM-DT, iDNA-Prot, DNA-Prot, DNAbinder, DNA-BIND and DNAPred methods on average by 26.7%, 41.5%, 22.1%, 121.3% and 11.7% based on ACC, sensitivity, specificity, MCC and AUC respectively.

Moreover, Figure 2 presents the ROC curves generated by StackDPPred and PSSM-DT while the predictions are evaluated on PDB186 test set. As PSSM-DT is the highest performing among the existing methods, we again directly compare the performance of StackDPPred with PSSM-DT. The curves in Figure 2 highlight the strength of StackDPPred in achieving a high TPR of ≥ 85% (rate of correct prediction of DNA-binding proteins) at a very low FPR of around 20%. Whereas, PSSM-DT provides TPR of ≥ 80% at a cost of higher FPR ≥ 20%. In addition, the AUC score given by StackDPPred is 1.58% higher than that of PSSM-DT when evaluated on PDB186 test set.

Additionally, to further evaluate the robustness of StackDPPred, we performed a test on PDB676 test set, collected in this study. As PSSM-DT is the highest performing among the existing methods, we directly compare the performance of StackDPPred with PSSM-DT. Table 5 lists the predictive results of StackDPPred and PSSM-DT (Xu et al. 2015) on the PDB676 test set. Table 5 indicates that based on PDB676, the StackDPPred outperforms PSSM-DT method by 5.84%, 3.63%, 8.06%, 15.2% and 1.2% based on ACC, sensitivity, specificity, MCC and AUC respectively.

Table 4. Comparisons of StackDPPred with other state-of-the-art methods on independent test dataset, PDB186.

Methods	ACC	Sensitivity	Specificity	MCC	AUC
PSSM-DT (imp. %)	0.8000 (8.20%)	0.8709 (6.18%)	0.7283 (10.73%)	0.6470 (13.8%)	0.8740 (1.58%)
iDNA-Prot (imp. %)	0.6720 (28.8%)	0.6770 (36.6%)	0.6670 (20.9%)	0.3440 (114.1%)	0.8330 (6.58%)
DNA-Prot (imp. %)	0.6180 (40.06%)	0.6990 (32.3%)	0.5380 (49.9%)	0.2400 (206.8%)	0.7960 (11.5%)
DNAbinder (imp. %)	0.6080 (42.4%)	0.5700 (95.7%)	0.6450 (25.0%)	0.2160 (240.9%)	0.6070 (46.3%)
DNA-BIND (imp. %)	0.6770 (27.8%)	0.6670 (62.2%)	0.6880 (17.21%)	0.3550 (115.1%)	0.6940 (27.9%)
DBPPred (imp. %)	0.7690 (12.6%)	0.7960 (16.2%)	0.7420 (8.7%)	0.5380 (36.9)	0.7910 (12.2%)
StackDPPred (avg. imp. %)	**0.8655** **(26.7%)**	**0.9247** **(41.5%)**	**0.8064** **(22.1%)**	**0.7363** **(121.3%)**	**0.8878** **(11.7%)**

Here, 'imp.' stands for improvement. The 'imp. %' represents improvement in percentage achieved by StackDPPred over the respective method and 'avg. imp. %' represents average improvement in percentage achieved by StackDPPred over the listed methods.

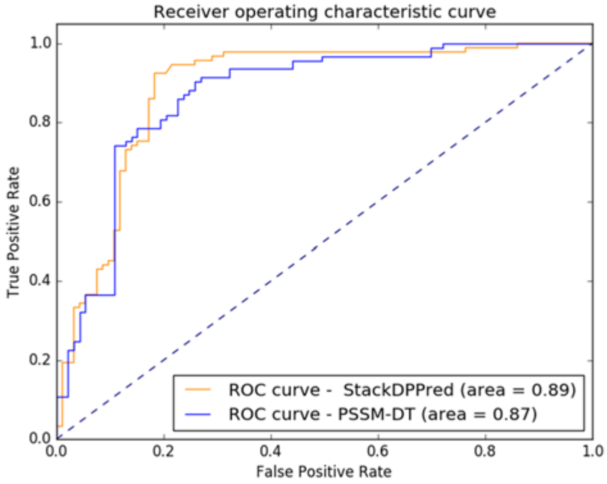


Fig. 2. Comparison of ROC curve and AUC scores given by StackDPPred and PSSM-DT on an independent test dataset, PDB186.

Table 5. Comparisons of StackDPPred with PSSM-DT on independent dataset, PDB676.

Methods	ACC	Sensitivity	Specificity	MCC	AUC
PSSM-DT (imp. %)	0.8121 (5.84%)	0.8166 (3.63%)	0.8077 (8.06%)	0.6243 (15.2%)	0.8872 (1.2%)
StackDPPred	**0.8595**	**0.8462**	**0.8728**	**0.7192**	**0.8978**

Here, 'imp.' stands for improvement. The 'imp. %' represents improvement in percentage achieved by StackDPPred over the respective method.

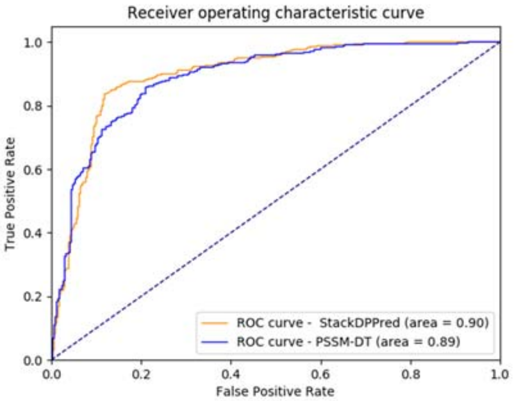


Fig. 3. Comparison of ROC curve and AUC scores given by StackDPPred and PSSM-DT on an independent test dataset, PDB676.

Moreover, Figure 3 presents the ROC curves generated by StackDPPred and PSSM-DT while the predictions are evaluated on PDB676 test set. Figure 3 also shows that the AUC score given by StackDPPred is 1.2% higher than that of PSSM-DT when evaluated on PDB676 test set.

This indicates that the proposed method, StackDPPred outperforms the existing methods and is a very promising predictor. This comprehensive investigation of the stacking based machine learning method and features in predicting DNA binding functions of proteins might yield a competitive tool for future proteomics studies.

4 Conclusions

In this work, we have designed and developed a stacking based machine learning technique, called StackDPPred, for the prediction of DNA-binding proteins given the protein sequence. With an aim to improve the prediction accuracy of DNA-binding proteins, we have investigated and utilized different feature extractions and encoding techniques along with the advanced machine learning technique called stacking. We have extracted the useful evolutionary information feature from the PSSM and residue-wise contact energy from the RCEM. Next, we use them to train the ensemble of predictors at the first-stage (base-layer). Then, we combine the output of the predictors at the base-layer using another SVM at the second-stage (meta-layer). As a result, the meta-learner SVM of the StackDPPred achieves an ACC of 89.96%, MCC of 0.7990 and AUC of 0.9449 on a benchmark dataset, whereas the base-learner SVM achieves an ACC of 80.43% and MCC of 0.60849. This achievement, allows us to conclude that stacking technique helps reduce the generalization error and thus can improve accuracy significantly. As for the commonly used independent test dataset PDB186, StackDPPred attains an ACC of 86.56%, MCC of 0.7363 and AUC of 0.8878. Additionally, for the new independent test dataset, PDB676 introduced in this study, StackDPPred attains an ACC of 85.95%, MCC of 0.7192 and AUC of 0.8978. We have found that the proposed method, StackDPPred, outperforms existing methods based on both benchmark validation and independent test datasets. These promising results indicate that StackDPPred can be effectively used for large-scale annotation of proteins based on DNA-binding affinity, given only the sequence information.

Acknowledgements

The authors thank Glenn McLellan for the proofreading of this manuscript.

Funding

The authors gratefully acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016-19)-RD-B-07.

Conflict of Interest: none declared.

References

Ahmad, Shandar and Sarai, Akinori (2004), 'Moment-based prediction of DNA-binding proteins', *Journal of Molecular Biology,* 341 (1), 65-71.

Altman, N. S. (1992), 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', *The American Statistician,* 46, 175-85.

Altschul, Stephen F., et al. (1997), 'Gapped BLAST and PSI-BLAST: a new generation of protein database search programs', *nucleic Acids Research,* 25 (17).

Andrabi, Munazah, et al. (2009), 'Prediction of mono- and di-nucleotide-specific DNA-binding sites in proteins using neural networks', *BMC Structural Biology,* 9 (30).

Babu, M Madan, et al. (2016), 'Intrinsically disordered proteins: regulation and disease', *Current Opinion on Structural Biology,* 21 (3).

Bhardwaj, Nitin, et al. (2005), 'Kernel-based machine learning protocol for predicting DNA-binding proteins', *Nucleic Acids Res.,* 33 (20), 6486-93.

Biswas, Ashis Kumer, Noman, Nasimul, and Sikder, Abdur Rahman (2010), 'Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information.', *BMC Bioinformatics,* 11 (273).

Breiman, Leo (1996), 'Bagging predictors', *Machine Learning,* 24 (2), 123-40.

Brown, JB and Akutsu, Tatsuya (2009), 'Identification of novel DNA repair proteins via primary sequence, secondary structure, and homology', *BMC Bioinformatics,* 10 (25).

Camacho, Christiam, et al. (2009), 'BLAST+: architecture and applications.', *BMC Bioinformatics,* 10 (421).

Dosztányi, Zsuzsanna, et al. (2005), 'The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins', *Journal of Molecular Biology,* 347 (4), 827-39.

Feng, Zhi-Ping and Zhang, Chu-Ting (2000), 'Prediction of membrane protein types based on the hydrophobic index of amino acids.', *Journal of Protein Chemistry,* 19 (4), 269-75.

Frank, Eibe, et al. (2004), 'Data mining in bioinformatics using Weka', *Bioinformatics,* 20, 2479-81.

Gao, Mu and Skolnick, Jeffrey (2008), 'DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions', *Nucleic Acids Res.,* 36 (12), 3978-92.

--- (2009), 'A threading-based method for the prediction of DNAbinding proteins with application to the human genome', *Plos One,* 5.

Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis (2006), 'Extremely randomized trees', *Machine Learning,* 63 (1), 3-42.

Govindan, Geetha and Nair, Achuthsankar S. (2011), 'New feature vector for apoptosis protein subcellular localization prediction', *Advances in Computing and Communications,* 170, 294-301.

Harris, Timothy D., et al. (2008), 'Single-molecule DNA sequencing of a viral genome.', *Science,* 320 (5872), 106-09.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome (2009), *The Elements of Statistical Learning* (2 edn., Springer Series in Statics: Springer-Verlag New York).

Ho, Tin Kam (1995), 'Random decision forests', *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* (Montreal, Que., Canada: IEEE), 278-82.

Hoque, Md Tamjidul, et al. (2016), 'sDFIRE: Sequence-specific statistical energy function for protein structure prediction by decoy selections.', *Journal of Computational Chemistry,* 37 (12), 1119-24.

Hu, Qiwen, et al. (2015), 'A Stacking-Based Approach to Identify Translated Upstream Open Reading Frames in Arabidopsis Thaliana', *International Symposium on Bioinformatics Research and Applications* (Bioinformatics Research and Applications), 138-49.

Huang, Hui-Lin, et al. (2011), 'Predicting and analyzing DNA-binding domains using a systematic approach to identifying a set of informative physicochemical and biochemical properties.', *BMC Bioinformatics,* 12 ((Suppl 1):S47).

Iqbal, Sumaiya and Hoque, Md Tamjidul (2015), 'DisPredict: A Predictor of Disordered Protein Using Optimized RBF Kernel', *PLOS ONE,* 10 (10), e0141551.

--- (2016), 'Estimation of Free Energy Contribution of Protein Residues as Feature for Structure Prediction from Sequence', *PLOS ONE,* 11 (9), e0161452.

Iqbal, Sumaiya, Mishra, Avdesh, and Hoque, Tamjidul (2015), 'Improved Prediction of Accessible Surface Area Results in Efficient Energy Function Application', *Journal of Theoretical Biology,* 380, 380-91.

Islam, Nasrul, et al. (2016), 'A Balanced Secondary Structure Predictor', *Journal of Theoretical Biology,* 389, 60-71.

Jeong, Jong cheol, Lin, Xiaotong, and Chen, Xue-wen (2011), 'On Position-Specific Scoring Matrix for Protein Function Prediction', *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 308 (15).

Kandaswamy, Krishna Kumar, Pugalenthi, Ganesan, and Suganthan, Ponnuthurai N. (2011), 'DNA-Prot: Identification of DNA Binding Proteins from Protein Sequence Information using Random Forest', *Journal of Biomolecular Structure and Dynamics,* 26 (6).

Kumar, Manish, Gromiha, Michael M, and Raghava, Gajendra PS (2007), 'Identification of DNA-binding proteins using support vector machines and evolutionary profiles.', *BMC Bioinformatics,* 8 (463).

Lin, Chen, et al. (2013), 'Hierarchical Classification of Protein Folds Using a Novel Ensemble Classifier', *Plos One*.

Liolios, Konstantinos, et al. (2006), 'The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide.', *Nucleic Acids Res.,* 34, D332-D34.

Liu, Bin, et al. (2008), 'A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis.', *BMC Bioinformatics,* 9 (510).

Liu, Bin, et al. (2012), 'Using amino acid physicochemical distance transformation for fast protein remote homology detection', *PLOS ONE,* 7 (9:e46633).

Liu, Bin, et al. (2014), 'Using distances between Top-n-gram and residue pairs for protein remote homology detection.', *BMC Bioinformatics,* 15(Supple 2):S3.

Lou, Wangchao, et al. (2014), 'Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and gaussian naïve bayes.', *PLOS ONE,* 9 (1), e86703.

Luscombe, Nicholas M, et al. (2000), 'An overview of the structures of protein-DNA complexes.', *Genome Biology*.

Mishra, Avdesh and Hoque, Md Tamjidul (2017), 'Three-Dimensional Ideal Gas Reference Sstate Based Energy Function', *Current Bioinformatics,* 12 (2), 171-80.

Mishra, Avdesh, Iqbal, Sumaiya, and Hoque, Md Tamjidul (2016), 'Discriminate protein decoys from native by using a scoring function based on ubiquitous Phi and Psi angles computed for all atom.', *Journal of theoretical biology,* 398, 112-21.

Moroni, Elisabetta, Caselle, Michele, and Fogolari, Federico (2007), 'Identification of DNA-binding protein target sequences by physical effective energy functions: free energy analysis of lambda repressor-DNA complexes.', *BMC Structural Biology,* 7 (61).

Nagi, Sajid and Bhattacharyya, Dhruba Kr. (2013), 'Classification of microarray cancer data using ensemble approach', *Network Modeling Analysis in Health Informatics and Bioinformatics,* 2 (3), 159-73.

Nanni, Loris and Lumini, Alessandra (2008), 'Combing ontologies and dipeptide composition for predicting DNA-binding proteins', *Amino Acids,* 34 (4), 635-41.

Nanni, Loris, Brahnam, Sheryl, and Lumini, Alessandra (2010), 'High performance set of PseAAC and sequence based descriptors for protein classification.', *Journal of Theoretical Biology,* 266 (1), 1-10.

Nimrod, Guy, et al. (2010), 'iDBPs: a web server for the identification of DNA binding proteins.', *Bioinformatics,* 26 (5), 692-93.

Pedregosa, Fabian, et al. (2011), 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research,* 12, 2825-30.

Qian, Ziliang, Cai, Yu-Dong, and Li, Yixue (2006), 'A novel computational method to predict transcription factor DNA binding preference.', *Biochemical and biophysical research communications,* 348 (3), 1034-37.

Saini, Harsh, et al. (2014), 'Protein structural class prediction via k-separated bigrams using position specific scoring matrix.', *Journal of Advanced Computational Intelligence and Intelligent Informatics,* 18, 474-79.

Saini, Harsh, et al. (2015), 'Probabilistic expression of spatially varied amino acid dimers into general form of Chou׳s pseudo amino acid composition for protein fold recognition.', *Journal of Theoretical Biology,* 380, 291-98.

Shanahan, Hugh P., et al. (2004), 'Identifying DNAbinding proteins using structural motifs and the electrostatic potential.', *Nucleic Acids Res.,* 32 (16), 4732-41.

Sharma, Alok, et al. (2013), 'A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition.', *Journal of Theoretical Biology,* 320, 41-46.

Sharma, Ronesh, et al. (2018), 'OPAL: prediction of MoRF regions in intrinsically disordered protein sequences.', *Bioinformatics,* 34 (11), 1850-58.

Stawiski, Eric W., Gregoret, Lydia M., and Mandel-Gutfreund, Yael (2003), 'Annotating nucleic acid-binding function based on protein structure', *Journal of Molecular Biology,* 326 (4), 1065-79.

Szilágyi, András and Skolnick, Jeffrey (2006), 'Efficient prediction of nucleic acid binding function from low-resolution protein structures.', *Journal of Molecular Biology,* 358 (3), 922-33.

Tarafder, Sumit, et al. (2018), 'RBSURFpred: Modeling protein accessible surface area in real and binary space using regularized and optimized regression.', *Journal of Theoretical Biology,* 441, 44-57.

Tjong, Harianto and Zhou, Huan-Xiang (2007), 'DISPLAR: an accurate method for predicting DNA-binding sites on protein surfaces.', *Nucleic Acids Res.,* 35 (5), 1465-77.

Vapnik, Vladimir N. (1999), 'An overview of statistical learning theory.', *IEEE Transactions on Neural Networks,* 10 (5), 988-99.

Verma, Aayushi and Mehta, Shikha (2017), 'A comparative study of ensemble learning methods for classification in bioinformatics', *7th International Conference on Cloud Computing, Data Science & Engineering - Confluence* (Noida, India: IEEE).

Verma, Ruchi, Varshney, Grish C., and Raghava, G. P. S. (2010), 'Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile.', *Amino Acids,* 39 (1), 101-10.

Vuzman, Dana, Hoffman, Yonit, and Levy, Yaakov (2012), 'Modulating protein–DNA interactions by post-translational modifications at disordered regions.', *Biocomputing*, 188-99.

Walter, Mathias C., et al. (2008), 'PEDANT covers all complete RefSeq genomes', *Neucleic Acids Res,* (37).

Wang, Bing, et al. (2005), 'Predicting protein interaction sites from residue spatial sequence profile and evolution rate.', *FEBS Letters,* 580 (2), 380-84.

Wei, Leyi, et al. (2014), 'Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set.', *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 11 (1), 192-201.

Wheeler, David A., et al. (2008), 'The complete genome of an individual by massively parallel DNA sequencing.', *Nature,* 452, 872-76.

Wolpert, David H. (1992), 'Stacked generalization', *Neural Networks,* 5 (2), 241-59.

Wu, Cathy H, et al. (2006), 'The Universal Protein Resource (UniProt): an expanding universe of protein information', *Nucleic Acids Research,* (34).

Xia, Jun-Feng, Zhao, Xing-Ming, and Huang, De-Shuang (2010), 'Predicting protein–protein interactions from protein sequences using meta predictor', *Amino Acids,* 39 (5), 1595-99.

Xiong, Yi, Liu, Juan, and Wei, Dong-Qing (2011), 'An accurate feature-based method for identifying DNA-binding residues on protein surfaces.', *Proteins,* 79 (2), 509-17.

Xu, Ruifeng, et al. (2015), 'Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation', *BMC Systems Biology,* 9.

Yan, Changhui, et al. (2006), 'Predicting DNA-binding sites of proteins from amino acid sequence', *BMC Bioinformatics,* 7 (1), 262.

Zhang, Lichao, Zhao, Xiqiang, and Kong, Liang (2014), 'Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou׳s pseudo amino acid composition.', *Journal of Theoretical Biology,* 355, 105-10.

Zhang, Ziding, Kochhar, Sunil, and Grigorov, Martin G. (2005), 'Descriptor-based protein remote homology identification.', *Protein Sci.,* 14 (2), 431-44.

Zhou, Hongyi and Skolnick, Jeffrey (2011), 'GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction.', *Biophys. J* . 101 (October 2011), 2043-52.

Zou, Chuanxin, Gong, Jiayu, and Li, Honglin (2013a), 'An improved sequence based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis.', *BMC Bioinformatics, 14* (90).

Zou, Quan, et al. (2013b), 'BinMemPredict: a web server and software for predicting membrane protein types', *Current Proteomics,* 10 (1), 2-9.

Zou, Quan, et al. (2014), 'Survey of MapReduce frame operation in bioinformatics.', *Brefings in Bioinformatics,* 15 (4), 637-47.

StackDPPred: A Stacking based Prediction of DNA-binding Protein from Sequence

(Supplementary Document)

Avdesh Mishra[1], Pujan Pokhrel[1] and Md Tamjidul Hoque[1,*]

[1]Department of Computer Science, University of New Orleans, 2000 Lakeshore Dr., New Orleans, LA, 70148, USA
*To whom correspondence should be addressed. Phone: +1(504)2802406, E-mail: thoque@uno.edu

Supplementary Table 1S: RCEM table used in the proposed experiment.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	-1.65	-2.83	1.16	1.8	-3.73	-0.41	1.9	-3.69	0.49	-3.01	-2.08	0.66	1.54	1.2	0.98	-0.08	0.46	-2.31	0.32	-4.62
C	-2.83	-39.58	-0.82	-0.53	-3.07	-2.96	-4.98	0.34	-1.38	-2.15	1.43	-4.18	-2.13	-2.91	-0.41	-2.33	-1.84	-0.16	4.26	-4 .46
D	1.16	-0.82	0.84	1.97	-0.92	0.88	-1.07	0.68	-1.93	0.23	0.61	0.32	3.31	2.67	-2.02	0.91	-0.65	0.94	-0.71	0. 90
E	1.8	-0.53	1.97	1.45	0.94	1.31	0.61	1.3	-2.51	1.14	2.53	0.2	1.44	0.1	-3.13	0.81	1.54	0.12	-1.07	1. 29
F	-3.73	-3.07	-0.92	0.94	-11.25	0.35	-3.57	-5.88	-0.82	-8.59	-5.34	0.73	0.32	0.77	-0.4	-2.22	0.11	-7.05	-7.09	-8 .80
G	-0.41	-2.96	0.88	1.31	0.35	-0.2	1.09	-0.65	-0.16	-0.55	-0.52	-0.32	2.25	1.11	0.84	0.71	0.59	-0.38	1.69	-1 .90
H	1.9	-4.98	-1.07	0.61	-3.57	1.09	1.97	-0.71	2.89	-0.86	-0.75	1.84	0.35	2.64	2.05	0.82	-0.01	0.27	-7.58	-3 .20
I	-3.69	0.34	0.68	1.3	-5.88	-0.65	-0.71	-6.74	-0.01	-9.01	-3.62	-0.07	0.12	-0.18	0.19	-0.15	0.63	-6.54	-3.78	-5 .26
K	0.49	-1.38	-1.93	-2.51	-0.82	-0.16	2.89	-0.01	1.24	0.49	1.61	1.12	0.51	0.43	2.34	0.19	-1.11	0.19	0.02	-1 .19
L	-3.01	-2.15	0.23	1.14	-8.59	-0.55	-0.86	-9.01	0.49	-6.37	-2.88	0.97	1.81	-0.58	-0.6	-0.41	0.72	-5.43	-8.31	-4 .90
M	-2.08	1.43	0.61	2.53	-5.34	-0.52	-0.75	-3.62	1.61	-2.88	-6.49	0.21	0.75	1.9	2.09	1.39	0.63	-2.59	-6.88	-9 .73
N	0.66	-4.18	0.32	0.2	0.73	-0.32	1.84	-0.07	1.12	0.97	0.21	0.61	1.15	1.28	1.08	0.29	0.46	0.93	-0.74	0. 93
P	1.54	-2.13	3.31	1.44	0.32	2.25	0.35	0.12	0.51	1.81	0.75	1.15	-0.42	2.97	1.06	1.12	1.65	0.38	-2.06	-2 .09
Q	1.2	-2.91	2.67	0.1	0.77	1.11	2.64	-0.18	0.43	-0.58	1.9	1.28	2.97	-1.54	0.91	0.85	-0.07	-1.91	-0.76	0. 01
R	0.98	-0.41	-2.02	-3.13	-0.4	0.84	2.05	0.19	2.34	-0.6	2.09	1.08	1.06	0.91	0.21	0.95	0.98	0.08	-5.89	0. 36
S	-0.08	-2.33	0.91	0.81	-2.22	0.71	0.82	-0.15	0.19	-0.41	1.39	0.29	1.12	0.85	0.95	-0.48	-0.06	0.13	-3.03	-0 .82
T	0.46	-1.84	-0.65	1.54	0.11	0.59	-0.01	0.63	-1.11	0.72	0.63	0.46	1.65	-0.07	0.98	-0.06	-0.96	1.14	-0.65	-0 .37
V	-2.31	-0.16	0.94	0.12	-7.05	-0.38	0.27	-6.54	0.19	-5.43	-2.59	0.93	0.38	-1.91	0.08	0.13	1.14	-4.82	-2.13	-3 .59
W	0.32	4.26	-0.71	-1.07	-7.09	1.69	-7.58	-3.78	0.02	-8.31	-6.88	-0.74	-2.06	-0.76	-5.89	-3.03	-0.65	-2.13	-1.73	-1 2.39
Y	-4.62	-4.46	0.9	1.29	-8.8	-1.9	-3.2	-5.26	-1.19	-4.9	-9.73	0.93	-2.09	0.01	0.36	-0.82	-0.37	-3.59	-12.39	-2 .68

Supplementary Table 2S: Name and definition of the evaluation metric.

Name of Metric	Definition
True Positive (TP)	Correctly predicted DNA-binding proteins
True Negative (TN)	Correctly predicted non DNA-binding proteins
False Positive (FP)	Incorrectly predicted DNA-binding proteins
False Negative (FN)	Incorrectly predicted non DNA-binding proteins
Recall/Sensitivity/True Positive Rate (TPR)	$\frac{TP}{TP + FN}$
Specificity/True Negative Rate (TNR)	$\frac{TN}{TN + FP}$
Fall-out Rate/False Positive Rate (FPR)	$\frac{FP}{FP + TN}$
Miss Rate/False Negative Rate (FNR)	$\frac{FN}{FN + TP}$
Accuracy (ACC)	$\frac{TP + TN}{FP + FP + TN + FN}$
Balanced Accuracy (Bal_ACC)	$\frac{1}{2}(\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$
Precision	$\frac{TP}{TP + FP}$
F1 score (Harmonic mean of precision and recall)	$\frac{2TP}{2TP + FP + FN}$
Mathews Correlation Coefficient (MCC)	$\frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FN) * (TP + FP) * (TN + FP) * (TN + FN)}}$

Explored Base and Meta Classifiers: In order to find the base-classifiers to use in the first-stage and the meta-classifier to use in the second-stage of stacking framework, we explored six different machine learning algorithms and they are:

i) *Support Vector Machine (SVM):* We have used SVM (Vapnik 1999) with the radial basis function (RBF) kernel as one of the base-classifiers as well as a meta-classifier. SVM classifies by maximizing the separating hyperplane between two classes and penalizes the instances on the wrong side of the decision boundary using a cost parameter, *C*. The RBF kernel parameter, γ and the cost parameter, *C* are optimized to achieve the best jackknife validation accuracy using a grid search (Bergstra and Bengio 2012). The best values of the parameters of the base-classifier SVM are found to be $C = 2^{3.50}$ and $\gamma = 2^{-11.25}$. Likewise, the best values of the parameters of the SVM, used as meta-classifier, are $C = 2^{11}$ and $\gamma = 2^{-16.25}$.

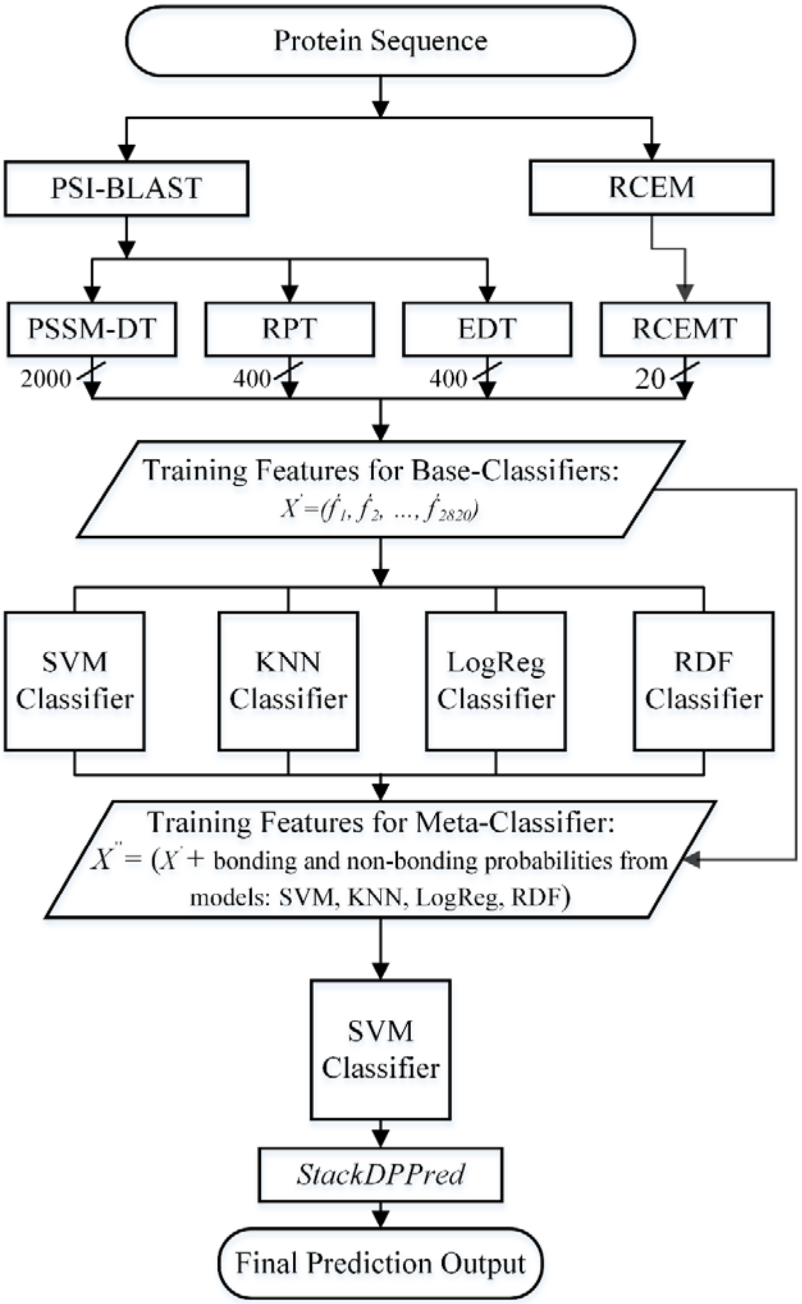
ii) *Logistic Regression (LogReg):* We have used LogReg (Hastie et al. 2009; Szilágyi and Skolnick 2006) with *L2* regularization as one of the base-classifiers. LogReg measures the relationship between the dependent variable, which is categorical (in our case: a protein being DNA-binding or not), and one or more independent variables by generating an estimation probability using logistic regression. The parameter, *C* which controls the regularization strength is optimized to achieve the best jackknife validation accuracy using grid search (Bergstra and Bengio 2012). In our implementation, we find $C = 0.1$ results in the best accuracy.

iii) *Extra Trees (ET) Classifier:* We have explored extremely randomized tree or ET (Geurts et al. 2006) which is one of the ensemble methods as a base-learner here. ET fits a number of randomized decision trees from the original learning sample and uses averaging to improve the predictive accuracy and control over-fitting. We have constructed the ET model with 1,000 trees and the quality of a split is measured by the Gini impurity index.

iv) *Random Decision Forest (RDF) Classifier:* We have used RDF (Ho 1995) as one of the methods for base-classifiers. It operates by constructing a multitude of decision trees on various sub-samples of the dataset and outputs the mean prediction of the decision trees to improve the predictive accuracy and control over-fitting. In our implementation of the RDF ensemble learner, we have used bootstrap samples to construct 1,000 trees in the forest.

v) *K Nearest Neighbor (KNN) Classifier:* We have used KNN (Altman 1992) classifier as one of the methods for base-classifiers. The KNN operates by learning from the *K* number of training samples closest in distance to the target point in the feature space. The classification decision is produced based on the majority votes coming from the neighbors. In this work, the value of *K* is set to 9 and all the neighbors are weighted uniformly.

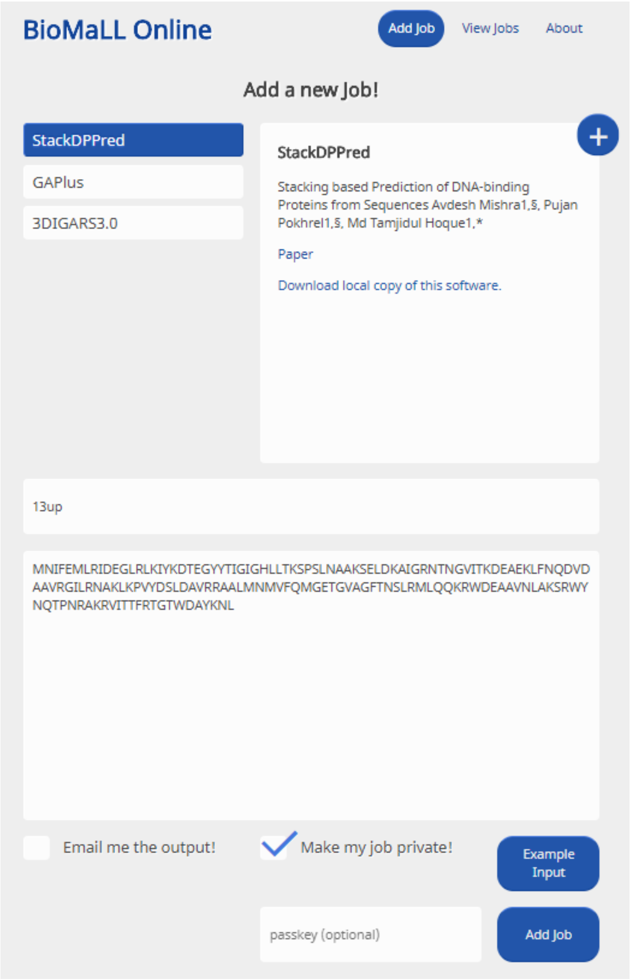
vi) *Bagging (BAG) Classifier:* We explored bootstrap aggregation or BAG (Breiman 1996) as one of the methods for base-classifiers in this study. The BAG method forms a class of algorithms which builds several instances of a classifier/estimator on random subsets of the original training set and then aggregates their individual predictions to form a final prediction. The BAG method is useful for reducing variance in the prediction. In this study, the bagging classifier is fit on multiple subsets of data with the repetitions using 1,000 decision trees, and the outputs are combined by weighted averaging.


Protein Sequence
PSI-BLAST
RCEM
PSSM-DT
RPT
EDT
RCEMT
2000
400
400
20
Training Features for Base-Classifiers:
$X^{'}=(f^{'}_{1}, f^{'}_{2}, \ldots, f^{'}_{2820})$
SVM Classifier
KNN Classifier
LogReg Classifier
RDF Classifier
Training Features for Meta-Classifier:
$X^{''}$ = ($X^{'}$ + bonding and non-bonding probabilities from models: SVM, KNN, LogReg, RDF)
SVM Classifier
StackDPPred
Final Prediction Output

Supplementary Fig. 1S. Overview of the StackDPPred prediction framework.

StackDPPred Online Server

StackDPPred is available as an online server at http://bmll.cs.uno.edu/add. The details of using StackDPPred are as follows: StackDPPred accepts input as a single protein sequence and generates an output which contains the predicted annotation as binding/non-binding followed by non-binding probability and binding probability. A screenshot of the top page of StackDPPred online server is shown in Supplementary Fig. 2S. To use StackDPPred, users need to provide a job title, protein sequence, email address and passkey before submitting a job to StackDPPred server. The email address and passkey are optional. If the user provides an email address with the job submission, then the output of the StackDPPred is sent to the email once the job is processed. Additionally, the passkey is used to make the output of the job private or public. By default the outputs of the jobs submitted to StackDPPred are public, but the users can make the job private by entering the passkey before submitting a job to StackDPPred server. The screenshot of the output of the StackDPPred online server is shown in Supplementary Fig. 3S.



StackDPPred

Stacking based Prediction of DNA-binding Proteins from Sequences Avdesh Mishra1,§, Pujan Pokhrel1,§, Md Tamjidul Hoque1,*

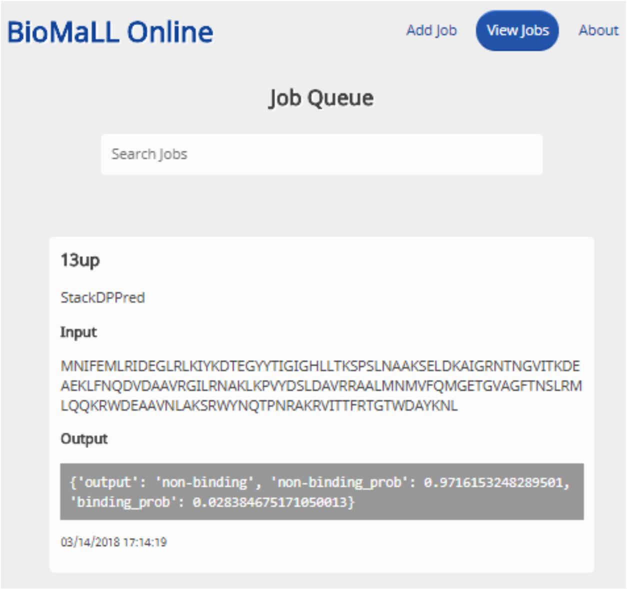
13up

MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNAAKSELDKAIGRNTNGVITKDEAEKLFNQDVD
AAVRGILRNAKLKPVYDSLDAVRRAALMNMVFQMGETGVAGFTNSLRMLQQKRWDEAAVNLAKSRWY
NQTPNRAKRVITTFRTGTWDAYKNL

□ Email me the output!

☑ Make my job private!

Supplementary Fig. 2S. Screenshot shows the top page of StackDPPred online server.



13up

StackDPPred

Input

MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNAAKSELDKAIGRNTNGVITKDE
AEKLFNQDVDAAVRGILRNAKLKPVYDSLDAVRRAALMNMVFQMGETGVAGFTNSLRM
LQQKRWDEAAVNLAKSRWYNQTPNRAKRVITTFRTGTWDAYKNL

Output

```
{'output': 'non-binding', 'non-binding_prob': 0.9716153248289501,
'binding_prob': 0.028384675171050013}
```

03/14/2018 17:14:19

Supplementary Fig. 3S. Screenshot shows the output page of StackDPPred online server.

References

Altman, N. S. (1992), 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', *The American Statistician,* 46, 175-85.

Bergstra, James and Bengio, Yoshua (2012), 'Random Search for Hyper-Parameter Optimization', *Journal of Machine Learning Research,* 13.

Breiman, Leo (1996), 'Bagging predictors', *Machine Learning,* 24 (2), 123-40.

Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis (2006), 'Extremely randomized trees', *Machine Learning,* 63 (1), 3-42.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome (2009), *The Elements of Statistical Learning* (2 edn., Springer Series in Statics: Springer-Verlag New York).

Ho, Tin Kam (1995), 'Random decision forests', *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* (Montreal, Que., Canada: IEEE), 278-82.

Szilágyi, András and Skolnick, Jeffrey (2006), 'Efficient Prediction of Nucleic Acid Binding Function from Low-resolution Protein Structures', *Journal of Molecular Biology,* 358 (3), 922-33.

Vapnik, Vladimir N. (1999), 'An Overview of Statistical Learning Theory', *IEEE TRANSACTIONS ON NEURAL NETWORKS,* 10 (5).